George Kampis
István Karsai
Eörs Szathmáry (Eds.)

LNAI 5778

Advances in Artificial Life

Darwin Meets von Neumann

10th European Conference, ECAL 2009
Budapest, Hungary, September 2009
Revised Selected Papers, Part II

2 Part II

Springer

Lecture Notes in Artificial Intelligence 5778
Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

George Kampis   István Karsai
Eörs Szathmáry (Eds.)

Advances in Artificial Life

Darwin Meets von Neumann

10th European Conference, ECAL 2009
Budapest, Hungary, September 13-16, 2009
Revised Selected Papers, Part II

Springer

Volume Editors

George Kampis
Eötvös University
Department of History and Philosophy of Science
Pázmány P.s. 1/C, 1117 Budapest, Hungary
E-mail: gk@hps.elte.hu

István Karsai
East Tennessee State Unviversity
Department of Biological Sciences
Johnson City, TN 37614-1710, USA
E-mail: karsai@etsu.edu

Eörs Szathmáry
Collegium Budapest, Szentháromság u.2
1014 Budapest, Hungary
E-mail: szathmary@colbud.hu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21313-7 e-ISBN 978-3-642-21314-4
DOI 10.1007/978-3-642-21314-4

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011927893

CR Subject Classification (1998): I.2, J.3, F.1.1-2, G.2, H.5, I.5, J.6

LNCS Sublibrary: SL 7 – Artificial Intelligence

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Artificial Life, unlike artifical intelligence, had humble beginnings. In the case of the latter, when the word itself was born, the first breathtaking results were already out, such as The Logic Theorist, a computer program developed by Allen Newell, Herbert Simon and Cliff Shaw in 1955–56. In artifical life, for a long while, amibition seems to have dominated over the results, and this was certainly true for the first, formative years. It was a bit unclear what *exactly* Artificial Life is. As not uncommon in science, the first definition was attempted in the negative form, just like when psychology (the study of the mental) was first defined as "anything not physics" (meaning, not natural science) in the nineteenth century. A tempting early definion of Artificial Life was one that distinguished it from theoretical and mathematical biology, modeling, evolution theory, and all the rest of what consituted "an old kind of science" about life. This was not without grounds, perhaps, and the parallel with artificial intelligence comes up again. Artificial intelligence was conceptually based on "machine functionalism," the philosophical idea that all operations, such as the mental operations of humans, are to be captured as "mere functions," or, in other words, as independent of their actual physical realizations. Functionalism has put computers and algorithms in the focus of interest in all dealings with human intelligence, and artificial intelligence was a computerized approach to the mind that was designed to capture human mental operations in the functional sense. Now it was simply the case that the *functionalism of life* was not yet born, and Artificial Life looked like the candidate that was needed for exactly that—to discover how computers can be used to uncover the secrets of life. There were cellular automata, that John von Neumann discovered back around 1950, that are capable of self-reproduction. Perhaps life was just a step away. This and a new fascination with functionalism in Artificial Life put computer scientists (who could technically master cellular automata and computer math) into a central position—in Artificial Life as it could be.

But Artificial Life became different. Incidentally, the slogan "life as it could be" was coined as a motto for Artificial Life, but now the same conditionals apply to Artificial Life itself. The reason is that functionalism turned out to be just one part of the story.

There is a well-known and much used (maybe over-used) phrase in biology, called "Dobzhansky's dictum," which says that "nothing in biology makes sense except in the light of evolution." Evolution is, as rightly captured in the dictum, central to the understanding of all things alive, and hence it is, and this had to be discovered, central to the studies of Artificial Life as well. And soon it also turned out that evolution cannot be readily reduced to algorithmic problems, or and least not in that pure, detached sense as it was attempted in functionalism. Evolution is *complex* in a sense recently acknowledged in the sciences of complexity: there is no single principle, no simple set of rules, and no transparency. Instead, evolution is a combination of many heterogeneous, and sometimes contingent factors, many of which have to do with the complex ways of existence of organisms: their body, their interaction, their development, their geo-spatial relations, their temporal history, and so on. This brought biology and biologists back into the equation, and Artificial Life has greatly benefited from that. Evo-devo (the interplay between evolutionary and developmental biology), evolutionary robotics, or systems biology are examples of fields where mathematical and algorithmic thinking combined with “wet” reality started to produce new and fascinating results. (Those who kept an eye on cognitive science and artificial intelligence found that over all those years a similar process has taken place there as well. *Embodiment*, or situated physical realization, has permeated and changed these fields to the same extent, or even more, as it did Artificial Life).

Today, as a result of these processes, Artificial Life is a prolific field that combines the best of computer science with the best of theoretical biology, mathematical modeling, and simulation. One way to express this is to say “Darwin meets von Neumann” at last—where “real” biology and “pure” function are no longer seen as contradicting, or even complementary. Rather, they permeate and fertilize each other in a number of fascinating ways.

ECAL 2009 was the 10th Europan Conference of Artificial Life, which means 20 years of history. It was an occasion to celebrate the 20 years of development of the field and also the new symbiosis referred to in the title. Famously, 2009 was also “Darwin year,” celebrating his 200th birthday and the 150 years of the *Origin of Species*. Thus it was highly appropriate to dedicate the meeting to Darwin—and von Neumann together. Five keynote lectures were delivered by eminent invited speakers, in the order of appearance they were: Peter Hammerstein (Humboldt University, Berlin), Hod Lipson (Cornell), Nick Barton (FRS, Edinburgh), Richard Watson (Southampton) and Eva Jablonka (Tel-Aviv University)—their choice reflected the spirit of convergence alluded to above.

The conference featured 161 submissions, out of which 54 were accepted as talks and 87 as posters (making up a total of 141 presentations). Adopting the recent practice of many science meetings, submissions could be based either on full papers or extended abstracts. The meeting was organized in a single track over three days, with parallel (whole-day) poster sections, to give best visibility to everyone’s work. We decided to publish all papers of the accepted presentations, not making any difference between posters and talks. This resulted from different factors: many excellent submissions had to be put into the poster section to keep reasonable time limits for the talks, and often this included second or third papers of some of the key speakers. Poster or talk was therefore not necessarily a quality issue. But also, we decided to publish all poster papers because we wanted to show the heterogeneity and the full spectrum of activities in the field, in order to provide an authentic overview. The result is this volume in 2 parts, containing 116 full papers.

The conference was sponsored by the Hungarian Academy of Science in different ways, one of them the special rates we enjoyed when using the wonderful historical building of the Academy in the very center of Budapest, just across the castle and the Chain Bridge. It is a building with a unique historical atmosphere and one that has seen many major scientific events. The conference talks were held in the Great Lecture Hall, which added to the impression that Artificial Life—and ECAL—are coming of age. The other sponsor was Aitia International, Inc., whose support is gratefully acknowledged. Aitia is the maker of MEME, or Model Exploration ModulE, a platform for DoE (design of experiments) and parameter sweeping, running on a cloud (https://modelexploration.aitia.ai/).

The publication process experienced several unexpected difficulties and delays. The proceedings could never have been published without a final push by Mark Jelasity, of Szeged University, a member of the Local Organizing Committee. It was his support and his offer for a hands-on contribution and equally his expertise of LaTeX and prior experience with Springer LNCS publications that made the essential difference that helped cross the line. Mark was offered but declined to be an Editor, lacking a scientific contribution to this conference and bearing a responsibility for the selection process, and this is a decision we had to respect. Nevertheless, here is a "big thank you," Mark. Several other people provided important help in the production of the volume, of whom Balazs Balint (Collegium Budapest) and Chrisantha Fernando (Sussex) need special mention. We thank the TenSi Congress Ltd. for the seamless technical organization of the meeting. In the evaluation phase, important help was given by several members of the Program Committee and also by additional reviewers, listed separately, whose contribution is highly appreciated. The conference and the proceedings have been the work of several people, and we thank all of them for making it happen. Finally, we thank Anna Kramer of Springer for her support and patience.

February 2011

George Kampis
István Karsai
Eörs Szathmáry (Editors)

Organization

Organizing Committee

George Kampis and Eörs Szathmáry (Chairs)
Chrisantha Fernando, Márk Jelasity, Ferenc Jordán, András Lőrincz, and István Scheuring

Program Committee

Wolfgang Banzhaf
Xabier Barandiaran
Zoltan Barta
Mark Bedau
Randall Beer
Seth Bullock
Philippe Capdepuy
Peter Cariani
Andy Clark
Tamás Czárán
Ezequiel Di Paolo
Dario Floreano
Inman Harvey
Takashi Ikegami
Istvan Karsai
Jozef Kelemen
Simon Kirby
Doron Lancet
Robert Lowe
John McCaskill
Bruce MacLennan
Federico Moran
Alvaro Moreno
Chrystopher Nehaniv
Stefano Nolfi
Charles Ofria
Alexandra Penn
Daniel Polani
Luis Rocha
Matthias Scheutz
Thomas Schmickl
Peter Stadler
Elio Tuci
Andreas Wegner
Franjo Weissing
Larry Yaeger
Klaus-Peter Zauner

Additional Reviewers

Rose Canino-Koning
Arthur Covert
Tomassino Ferrauto
Sherri Goings
Heather Goldsby
Laura Grabowski
David Knoester
Anuraag Pakanati
Bess Walker
Luis Zaman

Table of Contents – Part II

Group Selection

Ecosystems and Evolution

Algorithms and Evolutionary Computation

Philosophy and Arts

Optimization, Action, and Agent Connectivity

Swarm Intelligence

Table of Contents – Part I

Evolutionary Developmental Biology and Hardware

Evolutionary Robotics

Protocells and Prebiotic Chemistry

Systems Biology, Artificial Chemistry and Neuroscience

The Evolution of Cooperation

Investigations of Wilson's and Traulsen's Group Selection Models in Evolutionary Computation

Shelly X. Wu[⋆] and Wolfgang Banzhaf[⋆⋆]

Computer Science Department, Memorial University of Newfoundland,
St John's, Canada, A1B 3X5
{d72xw,banzhaf}@mun.ca

Abstract. Evolving cooperation by evolutionary algorithms is impossible without introducing extra mechanisms. Group selection theory in biology is a good candidate as it explains the evolution of cooperation in nature. Two biological models, Wilson's trait group selection model and Traulsen's group selection model are investigated and compared in evolutionary computation. Three evolutionary algorithms were designed and tested on an *n-player* prisoner's dilemma problem; two EAs implement the original Wilson and Traulsen models respectively, and one EA extends Traulsen's model. Experimental results show that the latter model introduces high between-group variance, leading to more robustness than the other two in response to parameter changes such as group size, the fraction of cooperators and selection pressure.

Keywords: the evolution of cooperation, evolutionary computation, Wilson's trait group selection model, Traulsen's group selection model.

1 Introduction

Evolutionary computation (EC) is often viewed as an optimization process, as it draws inspiration from the Darwinian principle of variation and natural selection. This implies that EC may fail to solve problems which require a set of cooperative individuals to jointly perform a computational task. When cooperating, individuals may contribute differently, and hence might lead to unequal fitnesses. Individuals with lower fitness will be gradually eliminated from the population, despite their unique contributions to overall performance of the algorithm. Hence, special mechanisms should be implemented in EC that avoid selecting against such individuals.

In nature, the success of cooperation is witnessed at all levels of biological organization. A growing number of biologists have come to believe that the theory of group selection is the explanation even though this theory has been unpopular for the past 40 years; hence new models and their applications are investigated

[⋆] S.W. is grateful to Tom Lenaerts and Arne Traulsen for helpful discussions.

[⋆⋆] W.B. would like to acknowledge support from NSERC Discovery Grants, under RGPIN 283304-07.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 1–9, 2011.

[2]. Individuals are divided into groups, and only interact with members in the same group. The emergence of cooperation is due to competition between individuals and between groups. Individual competition selects against individuals with lower fitness, but group competition favors individuals who cooperate with others, regardless to their individual fitness. The group selection model proposed by Wilson and Sober [9,12] and the model by Traulsen and Nowak [10] represent two research strands in this area; groups in Wilson's model are mixed periodically during evolution, while groups in Traulsen's model are isolated. Hence, selection between groups and within groups work differently in these models.

Extending these two models to encourage cooperation in artificial evolution is a relatively new research direction; most research [1,3,6,7] so far is based on Wilson's model or its variations, not on Traulsen's. This motivated us to investigate the role each model can play to encourage cooperation in EC, and to analyze their differences. Three evolutionary algorithms adapting the two models were designed and examined under different parameter settings; these parameters refer to group size, fraction of cooperators and selection pressure, and they directly affect the selection dynamics. Our results show that the algorithm which extends Traulsen's model is more robust towards parameter changes than the algorithms implementing the original Wilson and Traulsen models.

The reminder of this paper is organized as follows. Section 2 introduces the three evolutionary algorithms. Section 3 describes the experiments and the results obtained. Section 4 concludes.

2 Algorithms Design

Wilson's and Traulsen's models interpret the idea of group selection in a different fashion. In Wilson's model, groups reproduce proportional to group fitness. Offspring groups are periodically mixed in a migrating pool for another round of group formation. The number of individuals a group contributed to this pool is proportional to its size; so cooperative groups contribute more to the next generation. On the contrary, Traulsen's model keeps groups isolated. An individual is selected proportional to its fitness from the entire population, and the offspring is added to its parent's group. When the group reaches its maximal size, either an individual in this group is removed, or the group splits into two, so another group has to be removed. Cooperative groups grow faster, and therefore, split more often. For detailed descriptions of these models, please refer to [9,10,12].

Our study aims to investigate the performance of the two models in extending evolutionary algorithms to evolve cooperation. The investigation is conducted in the context of the *n-player* prisoner's dilemma (NPD). The NPD game offers a straightforward way of thinking about the tension between the individual and group level selection [4]; meanwhile it represents many cooperative situations in which fitness depends on both individual and group behavior. In this game, N individuals are randomly divided into m groups. Individuals in a group independently choose to be a cooperator or a defector without knowing the choice of others. The fitness function of cooperators ($f_{C_i}(x)$) and defectors ($f_{D_i}(x)$) in group i are specified by the following equations:

$$f_{C_i}(x) = base + w(\frac{b(n_i q_i - 1)}{n_i - 1} - c), \quad (0 \leq i < m) \tag{1a}$$

$$f_{D_i}(x) = base + w\frac{b n_i q_i}{n_i - 1}, \quad (0 \leq i < m) \tag{1b}$$

where $base$ is the base fitness of cooperators and defectors; q_i the fraction of cooperators in group i; n_i the size of group i; b and c are the benefit and cost caused by the altruistic act, respectively; w is a coefficient. Evidently, cooperators have a lower fitness than defectors, because they not only pay a direct cost, but also receive benefits from fewer cooperators than defectors do. The fitness of group i is defined as the average individual fitness. Although defectors dominate cooperators inside a group, groups with more cooperators have a higher group fitness. Hence, the dynamics between individual and group selection will drive the game in different directions.

A simple evolutionary algorithm implementing Wilson's model (denoted as W) is described in Algorithm 1.

Algorithm 1. An Evolutionary Algorithm Based on Wilson's Model

```
$P \leftarrow$ Initialize_Population($N, r$);
while population does not converge or max generation is not reached do
    $P' \leftarrow$ Disperse_Population($P, m$);
    Evaluate_Fitness($P'$);
    for $i \leftarrow 0$ to $N'$ do
        $gn \leftarrow$ Select_Group($P'$);
        $idv \leftarrow$ Select_Individual($P', gn$);
        $idv' \leftarrow$ Reproduce_Offspring($idv$);
        Add_Individual($idv', NP, gn$)
    end
    $P \leftarrow$ Mixing_Proportionally($NP$);
end
```

This algorithm starts with randomly initializing a population P with N individuals, r percent of which are cooperators. P is then divided into m groups, and the individual and group fitness of the dispersed population P' is evaluated. Afterwards, reproduction begins; a group with number gn is first selected, from which an individual idv is selected to produce offspring idv'. idv' is then added to group gn in the new population NP. The reproduction causes groups in NP vary in size, because the selection of groups is proportional to fitness. In total N' offspring will be produced, where N' is normally larger than population size N. This gives cooperators an opportunity to increase their frequency in the next generation. To maintain the original population size N, groups in NP are mixed and each contributes individuals proportional to its size to new population P. P will repeat the above steps until the population converges or the maximum number of generations is reached.

Algorithm 2. An Evolutionary Algorithm Based on Traulsen's Model

```
$P \leftarrow$ Initialize_Population($N, r$);
$P' \leftarrow$ Disperse_Population($P, m$);
while population does not converge or max generation is not reached do
    Evaluate_Fitness($P'$);
    for $i \leftarrow 0$ to $N''$ do
        $idv \leftarrow$ Select_Individual_from_Population($P'$);
        $idv' \leftarrow$ Reproduce_Offspring($idv$);
        Put_Back_to_Group($idv', gn$);
        if Group_Size($gn$) $> n$ then
            $rnum \leftarrow$ Generate_Random_Number($0, 1$);
            if $rnum < q$ then
                Split_Group($gn$);
                Remove_a_Group();
            else
                Remove_an_Individual_in_Group($gn$);
            end
        end
    end
end
```

Similarly, Traulsen's is embedded into an evolutionary algorithm shown in Algorithm 2. This algorithm initializes, divides, and evaluates the population the same way algorithm W does. However, there are two major differences. First, the population only disperses once at the beginning of the process; the groups are kept isolated afterwards. Second, the reproduction step is different. An individual *idv* is selected from the entire population for reproduction, rather than from a group. Offspring *idv'* is put back into its parent's group, group *gn*. If the size of group *gn* exceeds the pre-defined group size *n*, a random number *rnum* is generated. If *rnum* is less than a group splitting probability *q*, group *gn* splits and its individuals are randomly distributed into offspring groups. A group has to be removed to maintain a constant number of groups; otherwise, an individual from group *gn* is removed. In Traulsen's model, a group or an individual to be eliminated is randomly selected. As an extension, we also investigate selecting such a group or individual reversely proportional to its fitness. Therefore, two variations of Algorithm 2 are implemented, one refers to the former (denoted as T1) and the other to the latter (denoted as T2).

3 Investigations with the Algorithms

The investigations focus on the effects caused by different group size n, initial fraction of cooperators r, and coefficient w. Parameters n and r affect the assortment between cooperators and defectors in groups, and coefficient w affects the individual and group fitness; both cause changes in selection dynamics.

To focus on the selection dynamics, we assume asexual reproduction without the interference of mutation. A roulette wheel selection is adopted in the reproduction step for all algorithms. Parameters that are common to all experiments are set as follows: runs $R = 20$, generation $gen = 5,000$, population size $N = 200$, base fitness $base = 10$, benefit $b = 5$, cost $c = 1$, group splitting probability $q = 0.05$, N''=10, and N' is decided by the following equation [9].

$$N' = \sum_{i=1}^{m} n_i \times (q_i \times f_{C_i}(x) + (1 - q_i) \times f_{D_i}(x)) \tag{2}$$

For each algorithm, we measure the success ratio by the number of runs whose population converges to cooperators to the number of total runs 20. The larger the ratio, the more likely an algorithm favors cooperation. We also collect the average variance ratio [5], which indicates composition difference between groups. The higher this ratio, the more prominent the effect of group selection.

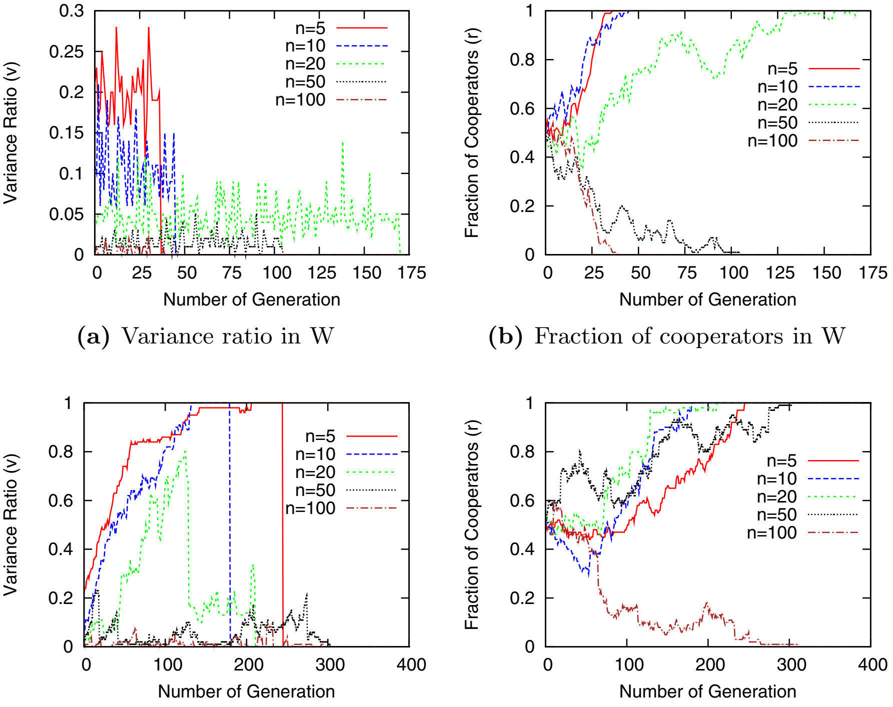
3.1 The Effects of Group Size and Initial Fraction of Cooperators

First we investigate how the three algorithms behave under different group sizes. We set $r = 0.5$ and $w = 1$. Group size n is varied from {5, 10, 20, 50, 100}. The success ratio and average variance ratio (in brackets) for each setting are listed in the first 3 columns in Table 1. As can be seen, the performance of T1 degrades as n grows. The population in W converges to cooperators when small groups are employed ($n = 5$ or 10). As n increases, evolving cooperation becomes difficult. In contrast, T2 converges to cooperators except for $n = 100$.

Table 1. The effects of group size n and initial fraction of cooperators r on the three algorithms

	r=0.5			r=0.3			r=0.1		
n	W	T2	T1	W	T2	T1	W	T2	T1
5	1(0.196)	1(0.820)	1	1(0.201)	1(0.853)	0.95	1(0.196)	1(0.893)	0.7
10	1(0.092)	1(0.655)	0.85	1(0.098)	1(0.665)	0.55	1(0.095)	1(0.767)	0.2
20	0.8(0.045)	1(0.291)	0.65	0.55(0.045)	1(0.398)	0.25	0.25(0.042)	0.65(0.465)	0.1
50	0(0.015)	1(0.112)	0.15	0(0.016)	0.8(0.105)	0.1	0(0.015)	0.55(0.049)	0.05
100	0(0.004)	0(0.011)	0	0(0.005)	0(0.014)	0	0(0.005)	0(0.015)	0

The observation can be explained by Figure 1. Figure 1(a) shows that variance ratio v in W decreases as n increases, which reduces the effect of group selection. As a result, selection on the individual level is becoming the dominate force, so the population converges quicker to defectors, see Figure 1(b). The same trend between v and n is also observed in T2. However, given that n ranges from 5 to 50, its v value is much higher than or equal to the highest v value of W (see Figure 1(c)). This implies that T2 preserves variance between groups better than W, and explains why T2 is more effective than W in evolving cooperation.



(a) Variance ratio in W

(b) Fraction of cooperators in W

(c) Variance ratio in T2

(d) Fraction of cooperators in T2

Fig. 1. The variance ratio v and fraction of cooperators r for algorithm W and T2 under different group sizes when $r = 0.5$ and $w = 1$

Unlike W, convergence to cooperators of T2 is not accelerated as n gets smaller; for example, runs with $n = 10$ converge first. When groups are too small or too large, much averaging is required to remove defectors from the population (see Figure 1(d)).

We further adjusted the value of r from 0.5 to 0.3 and 0.1. We were curious about the response of the three algorithms to this change, because when r drops, the number of cooperators assigned to groups is smaller, which increases the influence of individual selection in a group. As shown in Table 1, the performance of T1 decreases as r drops. For W and T2, when n is small (5 or 10), due to the strong group selection effects, the decrease of r does not affect the success ratio, but only slows convergence speed towards cooperation; for larger groups, as n increases (group selection is weaker) and r decreases (individual selection is stronger), group selection can hardly dominate individual selection; so it becomes difficult for both algorithms to preserve cooperation. However, T2 is less affected, because for a given group size, similar v values in W are observed despite the changes of r, while relatively high v values are produced by T2 even r drops.

3.2 Weak *vs.* Strong Selection

The composition of groups is not the only factor that drives selection dynamics; a difference in fitness values of cooperators and defectors is another one. It affects the pressure put on groups and individuals. In the next experiment, we use coefficient w to adjust the selection pressure. If w is small, the selection is called weak selection; otherwise it is called strong selection.

We tested the three algorithms with r=0.5 and w set to {0.1, 0.5, 1, 2, 5, 10}, respectively on all group sizes. Results are shown in Table 2. One first notices that the performance of the three algorithms increases and then deceases as selection pressure goes from weak to strong. If selection is too weak, the fitness between two roles and between groups are very close. Hence, group and individual selection become neutral, especially if large groups are adopted, so defectors can more easily take over the population. If the selection is too strong, though group selection still favors cooperative groups, because the larger fitness difference between both roles, cooperators are more difficult to be selected. To be more specific, for small groups (n = 5 or 10) only T2 can successfully preserve cooperation under both weak and strong selection. The increase of selection pressure raises the influence of individual selection. In response to this increase, the variance ratio in W for a given group size does not change at all, while T2 still keeps noticeable high variance ratios. This also explains why T2 outperforms W with larger groups.

Table 2. The performance of the algorithms under weak and strong selection

	w=0.1			w=0.5			w=1		
n	W	T2	T1	W	T2	T1	W	T2	T1
5	1(0.197)	1(0.949)	0.6	1(0.201)	1(0.884)	0.9	1(0.196)	1(0.820)	1
10	1(0.095)	1(0.766)	0.6	1(0.096)	1(0.515)	0.8	1(0.092)	1(0.655)	0.85
20	0.85(0.044)	0.95(0.601)	0.55	1(0.046)	1(0.370)	0.65	0.8(0.045)	1(0.291)	0.65
50	0.4(0.015)	0.45(0.174)	0	0(0.015)	1(0.115)	0.3	0(0.015)	1(0.112)	0.15
	w=2			w=5			w=10		
5	1(0.196)	1(0.806)	1	0.9(0.196)	1(0.820)	1	0(0.190)	1(0.875)	1
10	1(0.096)	1(0.543)	0.9	0.1(0.096)	1(0.596)	0.8	0(0.096)	1(0.638)	0.85
20	0.1(0.042)	1(0.309)	0.8	0(0.050)	0.8(0.334)	0.5	0(0.046)	0.8(0.347)	0.15
50	0(0.014)	0.8(0.079)	0.15	0(0.014)	0.45(0.021)	0	0(0.016)	0.1(0.053)	0

3.3 Discussion

The above experiments demonstrate that maintaining variance between groups has great impact on group selection models. For W, if groups are randomly formed, small group sizes are desired because small groups increase group variance. This confirms previous investigations (see [5,8,11] for examples). We further show that such a requirement only works if the selection is weak. T2, because it is able to introduce high group variance, is more robust towards parameter changes. The reason lies in how the two models manage groups. Mixing and reforming of groups in Wilson's model constantly averages the variance between groups, so in Figure 1(a) we observe the variance between groups fluctuating. In contrast, because groups in Traulsen's model are kept isolated, and the selection step in reproduction is proportional to individual fitness, the fraction of cooperators in a cooperative group grows faster than in a less cooperative group, hence gradually increases the variance between groups. T2 performs better than T1 under all settings, because removing an individual or a group according to reversed fitness value at death selection is very likely wiping out defectors, thus it certainly helps cooperators.

4 Conclusion

Wilson's and Traulsen's models are possible extensions of EC to evolve cooperation. Here, we investigated evolutionary algorithms that adapt the two models, and analyzed their differences. Our experimental results show that an algorithm which extends Traulsen's model is less sensitive to parameter changes than the algorithms based on the original Wilson and Traulsen models, because it is able to maintain high between-group variance, which is able to override individual selection arisen by the parameter changes. Future work will consider to theoretically investigate the extended algorithm; its extensions to multilevel selection; and its role in the theory of evolutionary transition.

References

1. Beckmann, B.E., McKinley, P.K., Ofria, C.: Selection for group-level efficiency leads to self-regulation of population size. In: Ryan, C., Keijzer, M. (eds.) Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO), pp. 185–192. ACM, New York (2008)
2. Borrello, M.E.: The rise, fall and resurrection of group selection. Endeavour 29(1), 43–47 (2005)
3. Chang, M., Ohkura, K., Ueda, K., Sugiyama, M.: Group selection and its application to constrained evolutionary optimization. In: Proceedings of the Congress on Evolutionary Computation (CEC), pp. 684–691. IEEE Computer Society, Los Alamitos (2003)
4. Fletcher, J.A., Zwick, M.: N-player prisoner's dilemma in multiple groups: A model of multilevel selection. In: Boudreau, E., Maley, C. (eds.) Proceedings of the Artificial Life VII Workshops (2000)
5. Fletcher, J.A., Zwick, M.: The evolution of altruism: Game theory in multilevel selection and inclusive fitness. Journal of Theoretical Biology 245, 26–36 (2007)
6. Hales, D.: Emergent group level selection in a peer-to-peer network. Complexus 3(1-3), 108–118 (2006)
7. Knoester, D.B., McKinley, P.K., Ofria, C.A.: Using group selection to evolve leadership in populations of self-replicating digital organisms. In: Proceedings of the Conference of Genetic and Evolutionary Computation (GECCO), London, England, pp. 293–300 (2007)

8. Lenaerts, T.: Different Levels of Selection in Artificial Evolutionary Systems. PhD thesis, Computational modeling lab, Vrije Universiteit Brussel (2003)
9. Sober, E., Wilson, D.S.: Unto Others: The Evolution and Psychology of Unselfish Behavior. Harvard University Press, Cambridge (1999)
10. Traulsen, A., Nowak, M.A.: Evolution of cooperation by multilevel selection. Proceedings of the National Academy of Sciences of the USA (PNAS) 103, 10952–10955 (2006)
11. Wade: A critical review of the models of group selection. The Quarterly Review of Biology 53, 101–104 (1978)
12. Wilson, D.S.: A theory of group selection. Proceedings of the National Academy of Sciences of the USA (PNAS) 72, 143–146 (1975)

The Evolution of Division of Labor

Heather J. Goldsby, David B. Knoester, Jeff Clune,
Philip K. McKinley, and Charles Ofria

Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan 48824
{hjg,dk,jclune,mckinley,ofria}@cse.msu.edu

Abstract. We use digital evolution to study the division of labor among heterogeneous organisms under multiple levels of selection. Although division of labor is practiced by many social organisms, the labor roles are typically associated with different individual fitness effects. This fitness variation raises the question of why an individual organism would select a less desirable role. For this study, we provide organisms with varying rewards for labor roles and impose a group-level pressure for division of labor. We demonstrate that a group selection pressure acting on a heterogeneous population is sufficient to ensure role diversity regardless of individual selection pressures, be they beneficial or detrimental.

Keywords: digital evolution, cooperative behavior, specialization, altruism.

1 Introduction

Within nature, many organisms live in groups where individuals assume different roles and cooperate to survive [1–4]. For example, in honeybee colonies, among other roles, drones care for the brood, workers forage for pollen, and the queen focuses on reproduction [1]. A notable aspect of these roles is that they do not all accrue the same fitness benefits. For example, leadership is a common role found within multiple species, where the benefits of leadership are significantly greater than that of a follower. In human societies, leaders of organizations commonly earn many times more than the average worker [4]. An open question is why individuals would not all attempt to perform the role associated with the highest fitness benefit, or in other words, why individuals would perform roles that put their genes at an evolutionary disadvantage for survival. Group selection pressures among human tribes have been proposed as one possible explanation for the evolution of leaders and followers [4]. In this paper, we explore whether group selection is sufficient to produce division of labor, where individual selection rewards different roles unequally.

Numerous evolutionary computation approaches have been used to study the behavior of cooperative groups comprising heterogeneous members [5–9]. Two key differentiating characteristics for these approaches are the *level of selection* used (i.e., individual or group) and whether or not *division of labor*

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 10–18, 2011.

occurs. Ecological approaches [6] use individual-level selection in concert with limited resources to promote the evolution of specialists. Some coevolutionary approaches [5, 9] evolve cooperative groups using individual selection, where different species are isolated in distinct subpopulations. Cooperation among these species occurs only at the time of fitness evaluation, when individuals from one species are evaluated with representatives from each of the other species. Perez-Uribe *et al.* [7] and Waibel *et al.* [8] overview work performed in this area, and also describe the effects of varying the levels of selection and population composition (i.e., heterogeneous or homogeneous populations) [8]. However, these prior studies do not address multi-level selection, where organisms experience individual-level competition to survive and also group-level pressure to cooperate. To explore these conditions, which are pertinent to biological studies of the division of labor in cooperative groups, we apply multi-level selection to a heterogeneous population.

For this study, we use Avida [10], a digital-evolution platform previously used to study topics including the origin of complex features [11] and the evolution of cooperation among homogeneous individuals [12]. Within an Avida experiment, a population of self-replicating computer programs exists in a user-defined computational environment and is subject to mutations and natural selection. These digital organisms execute their genome to perform tasks that metabolize resources in the environment, interact with neighboring organisms, and self-replicate.

In this paper, we describe how we used Avida and multi-level selection to evolve groups of heterogeneous organisms that perform roles with different fitness benefits. First, we enabled organisms to self-select roles associated with different costs and/or benefits. We applied a group-level pressure for division of labor that successfully counteracted the individual pressure to perform only the highest rewarded role. Second, rather than having an organism select a role, we conducted experiments in which a role was associated with a labor task that the organism had to perform. Again, we observed the evolution of division of labor. Third, we analyzed one of the successful groups and determined that it used a combination of genotypic diversity, phenotypic plasticity, and cooperation to perform all roles. The model developed for this approach can be used to inform biological studies of cooperation, such as those performed by Dornhaus *et al.* for honeybees [1]. Additionally, this technique can serve as a means to achieve division of labor within artificial life in order to solve engineering problems, such as developing multiple software components that must interact to achieve an overall objective [12], as well as cooperation among heterogeneous robots [13].

2 Methods

For each experiment, 30 trials were conducted to account for the stochastic nature of evolution. Figure 1 depicts an Avida organism and population. An Avida organism consists of a circular list of instructions (its *genome*) and a virtual CPU that executes those instructions. The virtual CPU architecture comprises three general-purpose registers $\{AX, BX, CX\}$ and two stacks $\{GS, LS\}$. The standard AVIDA instruction set used in this study is Turing complete and is designed so that random mutations will always yield a syntactically correct program, albeit one that may not perform any meaningful computation [14]. This AVIDA instruction set performs basic computational tasks (addition, multiplication, and bit-shifts), controls execution flow, enables communication, and allows for replication. In this study, the instruction set also included several instructions developed for the evolution of distributed problem solving [12]; these instructions are summarized in Table 1.

AVIDA organisms can perform *tasks* that enable them to metabolize resources from their environment. It is typical for these tasks to be logical operations performed on 32-bit integers. Performing tasks increases an organism's *merit*, which determines the rate at which its virtual CPU will execute instructions relative to the other organisms in the population. For example, an organism with a merit of 2 will, on average, execute twice as many instructions as an organism with a merit of 1. Since organisms self-replicate, an organism with greater merit will generally out-compete other organisms, eventually dominating the population. For these experiments, the amount of merit that an organism gains for completing a task depends on a user-defined constant called the task's *bonus value*. When an organism performs a task, the organism's merit is multiplied by the task's bonus value. For example, if an organism performs a task with a bonus value of 2, its merit is doubled.

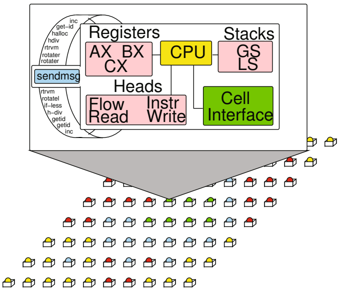


Fig. 1. An Avida organism and population

An AVIDA population comprises a number of *cells* in which at most one organism can live. Thus, the size of an AVIDA population is bounded by the number of cells in the environment. The cells are divided into a set of distinct subpopulations, called *demes*. In this study, demes compete every 100 *updates* in a tournament based on their fitness function, where a deme's fitness is determined by the behavior of its constituent organisms. An update is the unit of experimental time in AVIDA corresponding to approximately 30 instructions per organism. Each tournament contains a set of demes selected at random, and the deme with greatest fitness replaces the other demes (ties are broken randomly). When a deme is replaced, all organisms from the source deme are copied into the target deme, overwriting its previous inhabitants. Within each deme, organisms are still able to self-replicate. Thus, an individual's survival is dependent not only on its ability to out-compete its neighbors for the limited space available in its deme, but also on the collective behavior of the group. This process is similar to competition among human tribes [3].

For the experiments described in this paper, we created mutually-exclusive tasks for each possible role. An organism fulfills a role by performing its associated

Table 1. Communication and coordination instructions for this study

Instruction	Description
send-msg	Sends a message to the neighbor currently faced by the caller.
retrieve-msg	Loads the caller's virtual CPU from a previously received message.
rotate-left-one	Rotate this organism counter-clockwise one step.
rotate-right-one	Rotate this organism clockwise one step.
get-role-id	Sets register BX to the value of the caller's role-id register.
set-role-id	Sets the caller's role-id register to the value in register BX.
bcast1	Sends a message to all neighboring organisms.
get-cell-xy	Sets register BX and CX to the $x-y$ coordinates of the caller.
collect-cell-data	Sets register BX to the value of the cell data where the caller lives.

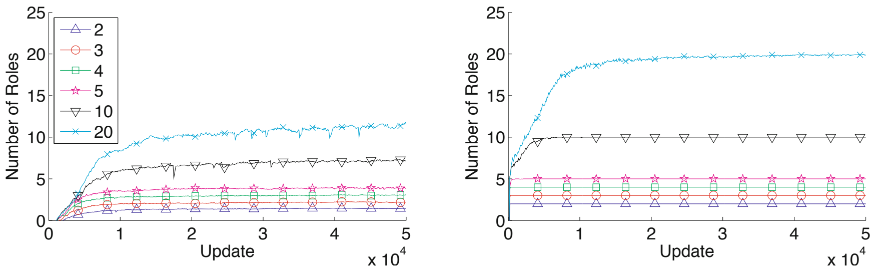
task; each organism may only have one role. For some of our experiments, we used *role-ids*, a mechanism whereby an organism sets a special-purpose virtual CPU register to an integer value, to indicate the role that an organism performs. For others, we required the organisms to implement logical operations. We tried these two mechanisms to see if the complexity of labor tasks affected the division of labor. We varied the benefits of performing a task (and thus performing a role) by changing the task's bonus value. In all experiments presented here, we used 400 demes, each containing 25 organisms on a 5 × 5 toroidal grid. Deme fitness was based on the diversity of tasks performed by the organisms. Thus, our experiments contain both the individual-level pressure to perform the task with the highest reward and the group-level pressure to diversify the tasks performed.

3 Experimental Results

Varying Roles. Our first experiment was designed to ascertain whether group selection is a strong enough pressure to produce division of labor when all roles have the same fitness benefit. For this experiment, we considered an organism to be performing a given role if it sets its role-id register to a desired value using the set-role-id instruction. We varied the desired number of roles from 2 to 20 and associated each role-id with a task that has a bonus value of 2. For example, when two roles are desired, the rewarded role-ids are 1 and 2. If an organism replicates after setting its role-id to 1, then it has completed task 1 and as a result, its merit is doubled. Similarly, if five roles are desired, then the rewarded role-ids are $\{1, 2, 3, 4, 5\}$. If an organism sets its role-id to a value outside this range, then no reward is granted. Additionally, we impose a group-level pressure for both the number of organisms that have set a role-id and the diversity of the role-ids. Specifically, the deme fitness function used here is:

$$F = \begin{cases} 1+n & \text{if } n < 25 \\ 1+n+r & \text{if } n = 25 \end{cases} \tag{1}$$

where F is the fitness of a deme, n is the number of organisms that have set a role-id, and r is the number of unique rewarded role-ids set by organisms in the



(a)

(b)

Fig. 2. (a) grand mean and (b) grand maximum number of unique roles over all demes when the number of desired roles was varied from 2 to 20

deme. Experiments described in this paper were repeated with tournaments of size 2, 5, and 10; results were not qualitatively different. Due to space limitations, we present results for a tournament of size 2, except where noted.

Figure 2 depicts the grand mean and maximum performance of all demes across 30 trials for each of $\{2, 3, 4, 5, 10, 20\}$ roles. The different curves represent the varying number of desired roles. In general, the best performing deme achieves the desired number of roles within 5,000 to 15,000 updates. Our analysis of the behavior of the demes indicates that they exploit location awareness (using instruction get-cell-xy) to determine which role to perform. Specifically, their x (or y) coordinate was used to calculate their role. Thus, we conclude that group selection is strong enough to produce division of labor among equally rewarded roles.

Varying Rewards. In the next set of experiments, we explored whether the group selection pressure for division of labor was strong enough to counteract rewarding roles unequally. The different rewards associated with roles provides an individual pressure to specialize on the most rewarded role, even if this behavior is detrimental to the performance of the group. This setup is designed to reflect a leader/follower situation, where it is desirable for the group to have one leader, and yet the rewards for the leader may be significantly different than those of a follower. To test this, we set the desired number of roles to be 2, and conducted trials for different multiplicative benefits of role-id 1 (the leader role). All other role-ids were neutral (i.e., they did not affect merit). We then specified a group-level pressure to limit the leader role to only a single organism. The deme fitness function used here was:

$$F = \begin{cases} 1+n & \text{if } n < 25 \\ (1+n-(o_1-d_{o_1}))^2 & \text{if } n = 25 \end{cases} \tag{2}$$

where F is the fitness of a deme, n is the number of organisms that have set a role-id, d_{o_1} is the desired number of organisms that perform the leader role, and o_1 is the actual number of organisms that perform the leader role.

Figure 3 depicts the results of varying the multiplicative benefit of the leader role from 0.5 (a penalty) to 64 (a significant reward) across 30 trials. Each treatment has two lines: a dashed line to indicate the number of followers, and a solid line to indicate the number of leaders; different symbols are used to indicate different treatments. In general, by 50,000 updates, the average deme has reached an equilibrium between leaders and followers, where the number of leaders is less than five in all treatments. The larger the benefit of leadership, the slower the population was to reach equilibrium and the larger the number of leaders. These results indicate that group selection is able to effectively counteract individual selection pressures.

To assess the generality of these results, we ran a control experiment without group selection, where the fitness of all demes were always equivalent. As expected, the majority of organisms chose to be leaders when there was a reward and followers when there was a penalty. Additionally, we ran two-role experiments where we set the desired number of leaders to be 13 (approximately half of the population). Similar to the results depicted in Figure 3, the population reached equilibrium by 50,000 updates with the population nearly evenly divided between leaders and followers. Lastly, we conducted treatments where we set the desired number of role-ids to be five and varied the distribution of the benefits among roles. Specifically, we conducted experiments where the benefits: (1) increased linearly; (2) where one role was rewarded significantly greater than the others; and (3), where the majority of task bonus values were penalties. For these experiments, the deme fitness function was the number of unique tasks performed. In all cases, the best performing deme rapidly evolved to perform all five roles, with the average deme performing three or more roles.

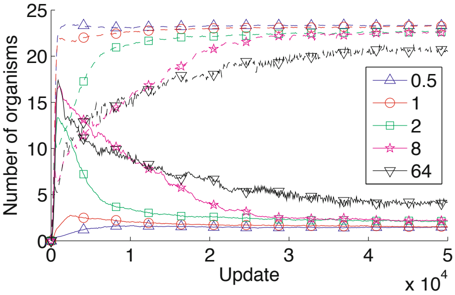
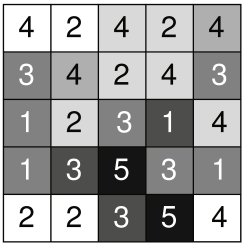


Fig. 3. Group selection is strong enough to overcome individual pressures for leadership, which we test by varying the multiplicative bonus value of leadership from 0.5 to 64

Increasing Complexity. For the last set of experiments, we studied whether this multi-level selection technique was sufficient to evolve division of labor when the complexity of the corresponding tasks varied. In the first two sets of experiments, an organism performed a role by setting its role-id to a specific value. For these experiments, we required the organism to perform a bit-wise logic operation on 32-bit integers. In this case, we used five mutually-exclusive logic operations: not, nand, and, orn, or. For the first experiment in this set, we rewarded all five logic operations equally with a task bonus value of 2. The deme fitness function was set to the number of unique tasks performed plus 1. The best demes varied between 4 and 5 tasks, whereas the average deme consistently performed between 3 and 4 tasks. Thus, this technique was successful in producing division of labor.

Next we examined whether division of labor occurred when the tasks were complex and their rewards were unequal. We rewarded the tasks based on complexity. Task not was assigned a bonus value of 2, tasks nand, and, orn were assigned a bonus value of 3, and or was assigned a bonus value of 4. This treatment increases the difficulty of the problem because organisms must evolve to perform logic operations and coordinate roles with different benefits. The best performing demes continued to vary between 4 and 5 tasks, and the average deme continued to perform between 3 and 4 tasks. This result indicates that the group selection pressure is strong enough to counteract the varying rewards among complex individual tasks.

Behavioral Analysis. To better understand how group selection maintained diversity of all five logic tasks in the previous treatment, we analyzed one deme from the end of a trial where all tasks had equivalent rewards. This deme had 18 unique genotypes. After we put the deme through a period of 100 updates without mutations, an *ecological period*, the deme maintained all 5 tasks and had 6 different genotypes. Figure 4 provides a graphical depiction of the genotype (shading) and phenotype (task-id of the task performed) of the organisms in the deme. Of the 6 genotypes, five exhibited *phenotypic plasticity.* Specifically, depending on their environmental context, genotypes in these families could perform one of two different tasks, nand/orn and not/and, respectively. For example, the first two blocks are both shaded white, indicating they have the same genotype, and yet one performs task 4 (orn) and the other performs task 2 (and). The remaining genotype (depicted in black) exclusively performed task 5 (or). It is often the case that explanations from a group selection perspective can also be interpreted from a kin selection perspective, and both are equally valid yet provide different intuitions [15]. For this paper, interpreting the results from a group selection perspective provides the most intuitive explanations. Additionally, our analyses revealed that the different genotypes belonged to three distantly-related lineages, with those performing tasks 1-4 more closely related to each other than to the genotype performing task 5.



4	2	4	2	4
3	4	2	4	3
1	2	3	1	4
1	3	5	3	1
2	2	3	5	4

Fig. 4. A deme after an ecological period. Genotypes are denoted by different color shading. Phenotypes are indicated by task-id of the role performed.

Lastly, we conducted "knockout" experiments for each communication (send-msg, retrieve-msg) and environment-sensing instruction (get-cell-xy, collect-cell-data). Specifically, all instances of the target instruction were replaced with a placeholder instruction that performs no useful function. We conducted 30 trials for each instruction knockout, and the results indicated that the ability to communicate with neighboring organisms and to sense their environment were critical for the success of the group. Without these instructions, at the end of a 100 update period (that did include mutations), the deme performed an average of 2.2 tasks. In summary, the most effective deme strategy relied on a combination of genotypic diversity, phenotypic plasticity, and cooperation to achieve all five tasks.

4 Conclusion

In this paper, we demonstrated that group selection is a sufficient pressure to produce division of labor among heterogeneous organisms. Specifically, we found that group selection can produce demes whose constituent organisms perform five different, mutually-exclusive complex roles, regardless of the underlying reward structure. In future work, we seek to use this technique to better understand behavior of social organisms and to harness this approach to apply evolutionary computation to complicated problems, such as automatically generating software systems [12].

Acknowledgements. The authors thank Ben Beckmann, Jeff Barrick, Rob Pennock, and Carlos Navarrete for their contributions. This work was supported by the U.S. National Science Foundation grants CCF-0820220, CCF-0643952, and CCF-0523449.

References

1. Dornhaus, A., Klügl, F., Puppe, F., Tautz, J.: Task selection in honeybees - experiments using multi-agent simulation. In: Proc. of the Third German Workshop on Artificial Life (1998)
2. Robinson, G.E.: Regulation of division of labor in insect societies. Annual Review of Entomology 37, 637–665 (1992)
3. Bowles, S., Hopfensitz, A.: The co-evolution of individual behaviors and social institutions. Journal of Theoretical Biology 223(2), 135–147 (2003)
4. VanVugt, M., Hogan, R., Kaiser, R.: Leadership, followership, and evolution: Some lessons from the past. American Psychologist 63, 182–196 (2008)
5. Potter, M.A., DeJong, K.A.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evol. Comput. 8(1), 1–29 (2000)
6. Goings, S., Ofria, C.: Ecological Approaches to Diversity Maintenance in Evolutionary Algorithms. In: IEEE Symposium on Artificial Life, Nashville, TN (2009)
7. Perez-Uribe, A., Floreano, D., Keller, L.: Effects of group composition and levels of selection in the evolution of cooperation in artificial ants. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) ECAL 2003. LNCS (LNAI), vol. 2801, pp. 128–137. Springer, Heidelberg (2003)
8. Waibel, M., Keller, L., Floreano, D.: Genetic team composition and level of selection in the evolution of cooperation. IEEE Transactions on Evolutionary Compuation (2009) (to appear)
9. Yong, C.H., Miikkulainen, R.: Cooperative coevolution of multi-agent systems. Technical report, University of Texas at Austin (2001)
10. Ofria, C., Wilke, C.O.: Avida: A software platform for research in computational evolutionary biology. Journal of Artificial Life 10, 191–229 (2004)

11. Lenski, R.E., Ofria, C., Pennock, R.T., Adami, C.: The evolutionary origin of complex features. Nature 423, 139–144 (2003)
12. McKinley, P.K., Cheng, B.H.C., Ofria, C., Knoester, D.B., Beckmann, B., Goldsby, H.J.: Harnessing digital evolution. IEEE Computer 41(1) (2008)
13. Labella, T.H., Dorigo, M., Deneubourg, J.L.: Division of labor in a group of robots inspired by ants foraging behavior. ACM Transactions on Autonomous and Adaptive Systems (TAAS) (2006)
14. Ofria, C., Adami, C., Collier, T.C.: Design of evolvable computer languages. IEEE Transactions in Evolutionary Computation 6, 420–424 (2002)
15. Foster, K., Wenseleers, T., Ratnieks, F., Queller, D.: There is nothing wrong with inclusive fitness. Trends in Ecology and Evolution 21, 599–600 (2006)

A Sequence-to-Function Map for Ribozyme-Catalyzed Metabolisms

Alexander Ullrich[1] and Christoph Flamm[2]

[1] University of Leipzig
Chair for Bioinformatics
Härtelstr. 16, 04275 Leipzig, Germany
[2] University of Vienna
Institute for Theoretical Chemistry
Währingerstr. 17, 1090 Vienna, Austria

Abstract. We introduce a novel genotype-phenotype mapping based on the relation between RNA sequence and its secondary structure for the use in evolutionary studies. Various extensive studies concerning RNA folding in the context of neutral theory yielded insights about properties of the structure space and the mapping itself. We intend to get a better understanding of some of these properties and especially of the evolution of RNA-molecules as well as their effect on the evolution of the entire molecular system. We investigate the constitution of the neutral network and compare our mapping with other artificial approaches using cellular automatons, random boolean networks and others also based on RNA folding. We yield the highest extent, connectivity and evolvability of the underlying neutral network. Further, we successfully apply the mapping in an existing model for the evolution of a ribozyme-catalyzed metabolism.

Keywords: genotype-phenotype map, RNA secondary structure, neutral networks, evolution, robustness, evolvability.

1 Introduction

For many years it is known that neutral mutations have a considerable influence on the evolution in molecular systems [14]. The RNA sequence-to-structure map with its many-to-one property represents a system entailing considerable neutrality. It has been used successfully to shed light on the mechanisms of evolution [12,13] and explain basic properties of biological systems, such as resolving the interplay between robustness and evolvability [19].

Several alternative scenarios have been proposed to describe the emergence and evolution of metabolic pathways (for a review see [4]) including spontanious evolution of enzymes with novel functions, the "backwards evolution scenario" suggesting that pathways evolve backward from key metabolites and the "enzyme recruitment scenario" or "patchwork model" which assumes that highly specialized an efficient enzymes evolved from a few inefficient ancestral enzymes

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 19–26, 2011.

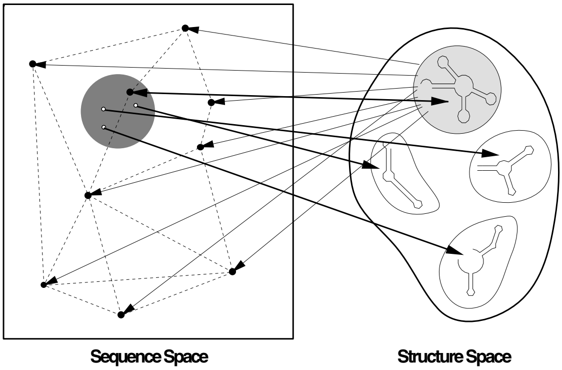


Fig. 1. The RNA sequence-to-structure map: There are many more sequences than structures which brings redundancy into the map. Sequences which fold into the same secondary structure form extended neutral networks in sequence space. The strong interweavement of the neutral networks implies that the sequences in a small volume around an arbitrary sequence realize all possible secondary structures.

with broad specificities. While these scenarios explain certain aspects of the early history of life and the evolution of metabolic pathways, the general mechanisms of the transition from an uncatalysed to a catalysed reaction network and the evolution of novel catalytic functions remains poorly understood.

We have proposed a multi-level computational model to study this transition in particular and the evolution of catalyzed reaction-networks in general [18]. The model combines a graph-based algebraic chemistry [1] with a RNA based structure-to-function-map within a protocellular entity. The structure-to-function-map projects sequence and structure features of the ribozyme into a hierarchical classification scheme for chemical reactions based on Fujita's "imaginary transition structure concept"[8]. The structure-to-function-map itself is evolvable and allows the emergence of completely new catalytic functions. Furthermore it forms the link between the genotype (i.e. the RNA sequence of the ribozymes) and the phenotype (i.e. the (partially) catalyzed reaction network).

2 The RNA Sequence-to-Structure Map

The folding of RNA sequences into their respective secondary structures [11] is a relatively well understood biophysically grounded process. Efficient algorithms [9] have been developed which allow the computation of nearly any desired thermodynamic or kinetic property of RNA sequences at the level of secondary structure. The relation between RNA sequences and their secondary structures can be viewed as a minimal model of a genotype-phenotype map (see Figure 1), where the RNA sequence carries the "heritable" information for the formation of the phenotype, which in turn is the RNA secondary structure upon which selection acts in evolutionary experiments [2,3].

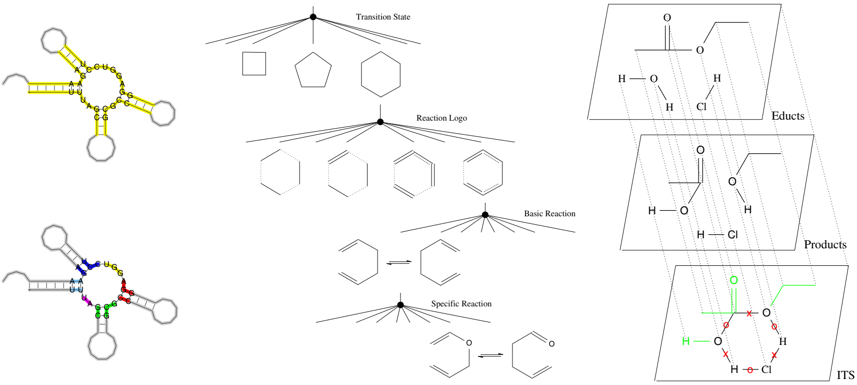


Fig. 2. The structure-to-function map: (lhs) The colored regions of the ribozyme fold determine the catalytic function i.e. which leaf in ITS-tree is picked; (middle) Along the levels of the ITS-tree the amount of chemical detail increases; (rhs) Acidic hydrolysis of ethyl acetat. The ITS of the reaction is gained from the superposition of educts and products by removing those atoms and bonds which do not directly participate in the reaction (shown in green).

The statistical architecture of the RNA sequence-to-structure map and it's implications for the evolutionary dynamics [6,7] has been extensively studied over the past decade. In particular, the map possesses a high degree of neutrality, i.e. sequences which fold into the same secondary structure are organized into extended mutationally connected networks reaching through sequence space. A travel along such a "neutral network" leaves the structure unchanged while the sequence randomizes. The existence of neutral networks in sequence space has been demonstrated in a recent experiment [16]. Due to the fact, that the neutral networks are strongly interwoven, the sequence-to-structure map shows another interesting property called "shape space covering" [17]. Meaning that within a relatively small volume of sequence space around an arbitrary sequence any possible secondary structure is realized. Both features of the RNA sequence-to-structure map account for directionality and the partially punctuated nature of evolutionary change.

3 The Mapping

In the following, we elaborate the aspects of our novel genotype-phenotype mapping. First, the mapping is realized in two steps. While both steps have a biological motivation, they are also designed regarding statistical considerations. We will refer to the two steps as the sequence-to-structure and the structure-to-function map, respectively.

The sequence-to-structure map is equivalent to the folding of the RNA-sequence to its minimum free energy structure. We use the RNA sequence-to-structure map on the one hand to model ribozymes for our artificial metabolism

(Section 4) and on the other hand to add the properties described in Section 2 to the mapping. The second and more interesting part, the structure-to-function map, can vary in some aspects for different applications, since the resulting phenotype spaces may differ in size and structure as well as the constitution of the phenotype itself. Here we focus on the mapping for our artificial metabolism model (Section 4). Therefore, we map from the RNA secondary structure to a chemical reaction. The reactions are represented by their transition state structure. The classification of those structures is hierarchical (see Figure 2). Accordingly, we extract features for every level, to determine the resulting reaction. Thereby, we regard structural and sequential information only of the longest loop in the structure and adjacent stems. The length of the loop specifies the size of the transition state, i.e. the number of involved atoms. Further, length and starting base-pairs of the adjacent stems define the constitution of the basic reaction, i.e. the abundance and position of single, double or triple bonds. Finally, the sequence inside the loop region decides over the specific reaction, i.e. the position and type of the involved atoms. The decision to focus on the loop region is based on the observation that most catalytic RNA molecules have their active center in a hairpin loop and also generally enzymes have an active site that covers only a small part of the entire structure. Furthermore, some studies suggest that transition state stabilizing is an important catalytic strategy for ribozymes [15], matching our representation of chemical reactions as transition state structures.

4 Ribozyme-Catalyzed Artificial Metabolism

In order to investigate the evolution of biological networks and the emergence of their properties, we have developed a sophisticated model of a ribozyme-catalyzed metabolism. The graph-based model is supported by an artificial chemistry, to allow for realistic kinetic behavior of the system. The basic elements of the model are represented as graphs. In case of the metabolites this complies with the intuitive presentation of molecules in chemistry. Similarly, metabolic reactions can be represented as a set of graphs or by one superimposition graph, combining the relevant parts of all involved molecules. The process of the reaction itself is performed by a graph-rewriting mechanism, through which new molecules can be generated. The conjunction of metabolites and their participation in metabolic reactions is captured in the metabolic network. We use several techniques to analyze the properties of the produced networks and compare them with analyzes of real-world networks [18]. Besides the metabolism, our model also comprises a genome that is divided into several genes by TATA-boxes. The genes are RNA-sequences of fixed length, intended to code for the catalytic elements, ribozymes, of the model system. The idea for the genotype-phenotype mapping that we introduce in this paper originates from the task to find a realistic and efficient mapping from these RNA-sequences (genotype) to the ribozymes (phenotype), or better, to the chemical reactions they implement.

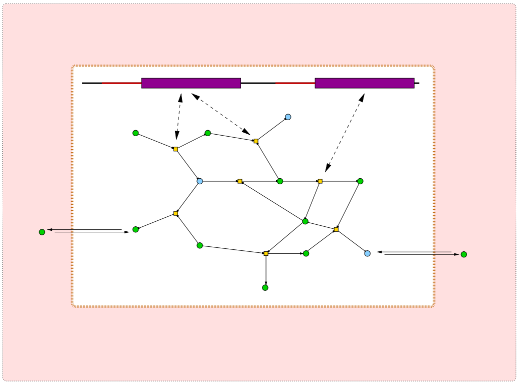


Fig. 3. Overview of the artificial metabolism model. Each gene of the genome expresses for a ribozyme. While these catalysts are trapped within the protocellular entities, metabolites under a certain size can pass through the membrane, allowing exchange with the environment.

5 Results

We will now compare our genotype-phenotype mapping with other existing approaches. Mappings based on cellular automaton (CA) and random boolean network (RBN) have been proven to exhibit desirable properties [5] and, thus, will serve here as reference. In previous evolutionary models and for the statistical analysis in this paper, we developed other mappings based on the RNA sequence-to-structure map but varying structure-to-function maps. Here we will mention only two of them. For the first mapping, we assign one target structure for each point in the phenotype space, the structure then maps to the phenotype of the closest target structure. The second mapping extracts structural features, similar to our mapping, but for the entire structure.

5.1 Random Neutral Walk

One way of gaining statistical results for the comparison of different genotype-phenotype mappings is through a random neutral walk on the genotype space. It starts from an arbitrary point of the mapping's genotype space. The first step is to choose randomly one neutral mutation out of the set of all possible one-point mutations. Applying the mutation, a new genotype is created that differs from its predecessor in only one position and maps to the same point in phenotype space. This procedure is repeated until the length of the walk reaches the pre-defined limit or no neutral mutation can be applied.

In each step, we keep track of several statistics that give insight into the structure of the neutral network. For this aim, we observe the one-point mutation neighborhoods of all genotypes of the random neutral walk. The fraction of neutral mutations within these neighborhoods will be referred to as the neutrality

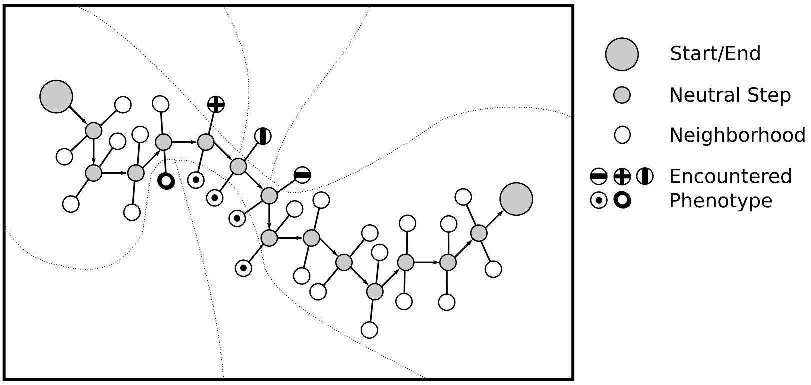


Fig. 4. Random neutral walk from an arbitrary point in genotype space to random neutral neighbors until a certain length is reached. In each step, encountered phenotypes in the one-point mutation neighborhood are protocolled.

of the underlying neutral network. In terms of biological systems this can also be regarded as the robustness that is gained through the mapping, since neutrality protects from harmful mutations. Further, we search for new phenotypes that are encountered during the course of the walk, i.e. the number of different phenotypes that were found in the one-point neighborhoods. This measurement indicates the rate with which the mapping discovers new phenotypes and other neutral components. The faster a system can access different points in the phenotype space, the more evolvable is it. Thus, the discovery rate can be equated with the notion of evolvability. Other measures that we regard as important here, are the extent and the connectivity of the neutral networks. Former, is given by the maximal distance between genotypes of the neutral walk. Latter, is defined by the fraction of possible phenotypes that ca be reached from an arbitrary starting configuration.

5.2 Comparison

For the following study, we performed 1000 random neutral walks of length 100 for each mapping. The size of the genotype spaces vary for the different mappings. For the RNA-based mappings the size is 4^{100}, for the random boolean network mapping it is 2^{144} and for the cellular automaton mapping 2^{76}. The phenotype space is $2^8 = 256$ for all mappings.

Figure 5 shows the results of the performed random walks. It can be seen that our mapping (RNA loop) reaches the most phenotypes (≈ 200 of 256), following the other RNA-based mappings (≈ 175 and 150), the random boolean network (145) and the cellular automaton (100). We can also see that the difference in the first steps is even more drastically, indicating a faster discovery rate of our mapping compared to all others. This can be explained by the connectivity, whereas CA and RBN have about 14 and 21 neighboring phenotypes,

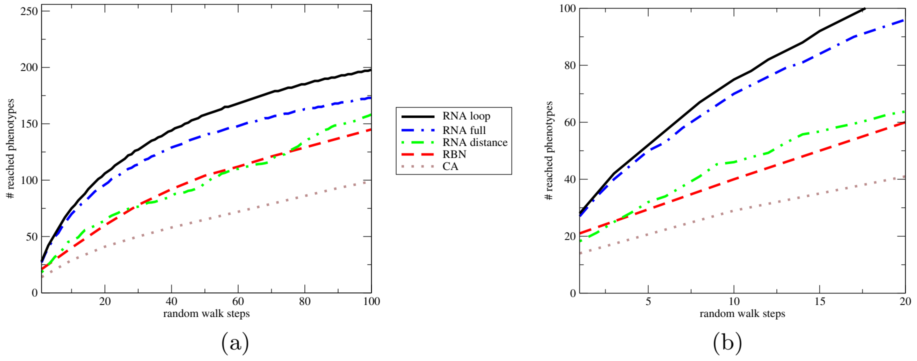


Fig. 5. Encountered phenotypes in a random neutral walk of length 100 (left) and for the first 20 steps (right). RNA loop = our mapping; RNA full = RNA-based mapping considering the entire structure; RNA distance = RNA-based mapping using target structures; RBN = random boolean network mapping; CA = cellular automaton mapping.

respectively, our mappings have about 27 distinct phenotypes in their one-point neighborhood. Furthermore, our mapping travels further in the genotype space, allowing a steady discovery of new phenotypes. However, in terms of neutrality the random boolean network mapping performs best, ≈ 58% of its mutations are neutral, for CA it is ≈44% and the RNA-based mappings are in between with around 50%.

5.3 Discussion

We have proposed here a genotype-phenotype mapping that is suitable for evolutionary studies, as shown for the given example and the comparative study with other mappings. Due to its advantageous properties, it is conceivable to use our mapping in studies where so far rather simple artificial genomes have been used and the RNA sequence-to-structure map can be integrated in a meaningful way, such as the evolution of regulatory networks [10]. First of all, our mapping allows open-ended evolution of the simulation model, since it continuously produces new phenotypes and does not get trapped in local optima. Further, it provides sufficient robustness to evolve a stable system and at the same time achieves high evolvabilty, allowing the system to react fast to perturbations or changes in the environment. We have also shown here, that a good genotype-phenotype mapping needs more than neutrality, the connectivity and extent of the underlying neutral network is at least as important.

Observing the mapping within the proposed evolutionary study of ribozyme-catalyzed metabolism, we hope not only to get insights about the emergence of pathways but also about the properties of the mapping itself, thus, the RNA sequence-to-structure map.

Acknowledgments

We thank the VolkswagenStiftung, the Vienna Science and Technology Fund (WWTF) Projectnumber MA07-30 and COST-Action CM0703 "Systems Chemistry" for financial support.

References

1. Benkö, G., Flamm, C., Stadler, P.F.: A Graph-Based Toy Model of Chemistry. J. Chem. Inf. Comput. Sci. 43, 1085–1093 (2003)
2. Bieberich, C.K.: Darwinian Selection of Self-Replicating RNA Molecules. Evolutionary Biology 16, 1–52 (1983)
3. Bieberich, C.K., Gardiner, W.C.: Molecular Evolution of RNA *in vitro*. Biophys. Chem. 66, 179–192 (1997)
4. Caetano-Anollés, G., Yafremava, L.S., Gee, H., Caetano-Anollés, D., Kim, H.S., Mittenthal, J.E.: The origin and evolution of modern metabolism. Int. J. Biochem. & Cell Biol. 41, 285–297 (2009)
5. Ebner, M., Shackleton, M., Shipman, R.: How neutral networks influence evolvability. Complex 7, 19–33 (2001)
6. Fontana, W., Schuster, P.: Continuity in evolution: on the nature of transitions. Science 280, 1451–1455 (1998)
7. Fontana, W., Schuster, P.: Shaping Space: the Possible and the Attainable in RNA Genotype-phenotype Mapping. J. Theor. Biol. 194, 491–515 (1998)
8. Fujita, S.: Description of Organic Reactions Based on Imaginary Transition Structures. J. Chem. Inf. Comput. Sci. 26, 205–212 (1986)
9. Gruber, A.R., Lorenz, R., Bernhart, S.H., Neuböck, R., Hofacker, I.L.: The Vienna RNA websuit. Nucl. Acids Res. 36, W70–W74 (2008)
10. Hallinan, J., Wiles, J.: Evolving genetic regulatory networks using an artificial genome. In: Proc. 2nd Conf. on Asia-Pacific Bioinformatics, vol. 29, pp. 291–296 (2004)
11. Hofacker, I.L., Stadler, P.F.: RNA secondary structures. In: Lengauer, T. (ed.) Bioinformatics: From Genomes to Therapies, vol. 1, pp. 439–489 (2007)
12. Huynen, M.A.: Exploring phenotype space through neutral evolution. J. Mol. Evol. 43, 165–169 (1996)
13. Huynen, M.A., Stadler, P.F., Fontana, W.: Smoothness within ruggedness: the role of neutrality in adaptation. Proc. Natl. Acad. Sci. USA 93, 397–401 (1996)
14. Kimura, M.: Evolutionary rate at the molecular level. Nature 217, 624–626 (1968)
15. Rupert, P.B., Massey, A.P., Sigurdsson, S.T., Ferre-D'Amare, A.R.: Transition State Stabilization by a Catalytic RNA. Science 298, 1421–1424 (2002)
16. Schultes, E.A., Bartel, D.P.: One sequence two ribozymes: Implications for the emergence of new ribozyme folds. Science 289, 448–452 (2000)
17. Schuster, P., Fontana, W., Stadler, P.F., Hofacker, I.L.: From sequence to shape and back. Proc. Roy. Soc. (London) B 255, 279–284 (1994)
18. Ullrich, A., Flamm, C.: Functional Evolution of Ribozyme-Catalyzed Metabolisms in a Graph-Based Toy-Universe. In: Heiner, M., Uhrmacher, A.M. (eds.) CMSB 2008. LNCS (LNBI), vol. 5307, pp. 28–43. Springer, Heidelberg (2008)
19. Wagner, A.: Robustness and evolvability: a paradox resolved Proceedings. Biological Sciences / The Royal Society 2758, 91–100 (2008)

Can Selfish Symbioses Effect Higher-Level Selection?

Richard A. Watson, Niclas Palmius, Rob Mills,
Simon T. Powers, and Alexandra Penn

Natural Systems group, University of Southampton, U.K
raw@ecs.soton.ac.uk

Abstract. The role of symbiosis in macro-evolution is poorly understood. On the one hand, symbiosis seems to be a perfectly normal manifestation of individual selection, on the other hand, in some of the major transitions in evolution it seems to be implicated in the creation of new higher-level units of selection. Here we present a model of individual selection for symbiotic relationships where individuals can genetically specify traits which partially control which other species they associate with – i.e. they can evolve species-specific grouping. We find that when the genetic evolution of symbiotic relationships occurs slowly compared to ecological population dynamics, symbioses form which canalise the combinations of species that commonly occur at local ESSs into new units of selection. Thus even though symbioses will only evolve if they are beneficial to the individual, we find that the symbiotic groups that form are selectively significant and result in combinations of species that are more cooperative than would be possible under individual selection. These findings thus provide a systematic mechanism for creating significant higher-level selective units from individual selection, and support the notion of a significant and systematic role of symbiosis in macro-evolution.

1 Introduction: Can Individual Selection Create Higher-Level Selection?

Symbiotic relationships in general are ubiquitous and uncontroversial, but the role of symbiosis in macro-evolutionary processes such as the major evolutionary transitions (1) and symbiogenesis (the creation of new species through symbiosis) (2), is poorly understood. Clearly, the evolution of symbiotic relationships may change the effective selection pressures on individuals in complex ways – but can they enable higher-level selection? When the fitness of individuals is context sensitive (i.e. under frequency dependent selection) grouping individuals together in small groups can change the average selection pressure on cooperative traits by altering the variance in contexts (3, 4). This effect is stronger when group membership is assortative on behavioural traits (5). In most models, however, the existence of groups is presupposed and accordingly any group selection effect observed is unsurprising in the sense that it is fully explained by changes in *individual* selection *given the context of these groups*. In contrast, we are interested in scenarios where individually selected traits affect the strength of group selection or create group selection de novo (6). For example, related

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 27–36, 2011.

work addresses the evolution of individually specified traits that affect group size (7, 8), or the evolution of markers that influence behavioural grouping (9). Here we address a multi-species scenario where species can evolve symbiotic relationships that allow explicit control over whether they group and who they group with.

Symbiosis, the living together of different species, implies that one species ‘seeks out’ another, actively controlling (to a limited extent) the species composition of its environmental context. When organisms create their own environments a complex dynamic is created between the traits they evolve that affect their symbiotic relationships, and the ‘ordinary traits’ (traits that do not affect symbioses) they evolve given the context they have created for themselves. Our research question concerns whether it is possible for an individual to evolve symbiotic relationships that cause it to create a significant higher-level unit of selection. This might seem to be a logical impossibility because for a higher-level unit of selection to be significant one would ordinarily assert that it must oppose individual selection. And, if a group opposes individual selection then a defector or selfish individual that exploits the group will be fit and take over. Of course, group selection that acts in alignment with individual selection is possible – e.g. individual selection may cause a mixed population to reach some evolutionarily stable strategy (ESS) (10) and group selection that acts in alignment with individual selection might cause a population to reach this ESS more quickly, but it cannot cause it to go somewhere other than the local ESS. But we show this conclusion is too hasty. We show that in cases where group selection acts in alignment with individual selection it *can* alter evolutionary outcomes. This requires that we consider a different type of evolutionary game, however.

The literature on group selection is largely preoccupied with the prisoners’ dilemma (11) – a game that has only one ESS (10) – ‘Defect’. Although a group of cooperative individuals is collectively fitter than a group of defectors, the cooperative group can never be stable given that the payoff for Defect is higher than the payoff for Cooperate when playing against other cooperators. Thus if groups are imposed Cooperate:Cooperate will beat Defect:Defect but it is not possible that a Cooperate:Cooperate group can be maintained by individual selection. In contrast, a game that has more than one ESS is a different matter. A coordination game of two strategies, for example, has two ESSs, let’s call them A-A and B-B, and these ESSs may have different overall utility, let’s say that an A-A group beats a B-B group. But the difference is that in a game that has multiple ESSs, each ESS can be supported by individual selection (there is no ‘cheat’ strategy that can invade either ESS) and this means that the two groups need not be externally imposed in order to be stable. Nonetheless, the evolutionary outcome can be significantly different from the outcome of individual selection without grouping. For example, with no grouping, if the utility of A-A is only slightly higher than B-B, then the population will reach the ESS that is closest to the initial conditions – for example, if B has a significant majority this will be the B-B ESS. But with grouping, the A-A ESS can be reached even if B has a significant majority because when A’s interact disproportionately with other A’s they are fitter than B’s. In the models that follow we will show that individual selection causes groups to form that represent combinations of species from different ESSs and thus allows the highest utility ESS to be found.

We intend our model to represent the evolution of symbiotic relationships between species, not just assortativity of behaviours within a single species. Thus we permit competition between heterogeneous groups (e.g. AB vs CD, where A and B are behaviours provided by unrelated species) rather than homogeneous groups (e.g. AA vs BB) as would be more conventional in a single-species assortative grouping model (where relatedness and inclusive fitness concepts straightforwardly apply) (12). By using a poly-species model we can show that the process we model significantly increases the likelihood of reaching a higher-utility ESS even in cases where the basin of attraction for high-utility ESSs is initially very small (13). Note that we do not change the interaction coefficients between species but only change the co-location or interaction probability of species. A species might thus change its fitness by increasing the probability of interacting with another (which is what we mean be a *symbiosis*) but it cannot change its intrinsic fitness dependency on that species (as might be part of a more general model of *coevolution* - 4,14,15). There are clearly many ways in which organisms can change interaction probabilities with other organisms either subtly or radically (16).

2 An Ecosystem Model with Evolved Symbioses

Our abstract model of an ecosystem contains 2*N* species, each of which contains *P* individuals. The fitness of each individual in each species will depend on the other species present in its local *environmental context.* A separation of timescales is crucial in this model (15): On the (fast) ecological dynamics timescale species densities within an environmental context change and quickly reach equilibrium, but on this timescale genetic changes are assumed to be negligible. At a much slower genetic evolution timescale, genetic changes that alter symbiotic relationships are significant. The genotype of an individual specifies partnerships with the other 2*N*-1 species that can partially (or completely) determine the combination of species it appears with in the environmental context. We assume that the initial composition of the local environment contains a random combination of species, but for the scenarios we investigate the ecological dynamics have only stable attractors, so the composition of the ecology quickly equilibrates to a subset of species that are stable. Although the frequency of a species may go to zero in a particular ecological context, in other contexts it will persist (i.e. no species are lost). Different individuals are evaluated in the environmental context for some time, and at the end of each period we turn attention to a new randomly initialised ecological context. Ours is therefore not an explicitly spatial model since we have no need to model different environmental contexts simultaneously.

We choose a very simple representation of the local environmental context – a binary vector representing which species are present in non-zero frequency. We suppose that each position in the vector is a ‘niche’ that may be occupied by one of two possible species that are mutually exclusive, such that some species cannot coexist in the same ecological context. For example, in a forest where deciduous and coniferous trees are competing, patches of the forest may, in simplistic terms, contain either one or the other but not both simultaneously, and simultaneously a patch may contain one species of ant or another but not both, and moreover, the type of tree present may influence which type of ant is fittest., and vice versa. An *N*-bit vector thus indicates which N species, of the possible $2N$, are present in the environmental context. A species, '------0---', indicates which type it is (e.g. '0') and which environmental niche in the environmental context it occupies (e.g. 6^{th}). This choice of representation has some properties that are required and some that are merely convenient. It is necessary for our purposes that not all species are present in all environmental contexts – otherwise, genetically specifying a symbiotic partnership would be redundant. It is also necessary that there are many different possibilities for the species composition in an environmental context – so the number of species present in any one environment should be large and many combinations of species should be allowed. The fact that species are arranged in mutually exclusive pairs is a contrivance for convenience: having all environmental states contain exactly N species allows us to define the environmental state and as N-dimensional space and to define fitness interactions between species using an energy function discussed below. And the fact that environmental states are defined using a binary 'present or not' representation rather than a continuous species density model is again a convenience – a continuous model would be interesting to investigate in future.

Each individual in the ecosystem has a fitness that is a function of the other species present in the current environmental context. In principle, this requires an environmentally sensitive fitness function for each species and the resultant ecological dynamics could be arbitrarily complex in general. In the experiments that follow we restrict our attention to ecosystems with simple monotone dynamics and point attractors. Such dynamics can be modelled using an 'energy function' (17,18) over environmental states, $e(E)$, such that the fitness of an individual of species, s, given an environmental context, E, is determined by the change in energy, $\Delta e(E, s)$ produced by adding s to E. That is, the fitness of an individual of species s in context E is, $fitness(s, E) = \Delta e(E, s) = e(E+s) - e(E)$, where '$E+s$' is the environmental state E modified by adding species s. (Dynamical systems theory would normally *minimise* energy, but for familiarity we let positive Δe correspond to positive fitness such that selection tends to *increase* e).

Each individual has a genotype that defines which other species it forms groups with (see Figure 1). This genotype is simply a binary vector length $2N$ defining which of N possible '0' species it groups with followed by which of N possible '1' species it groups with. Binary relationships of this form are somewhat crude perhaps, but although the partnerships of any one individual are binary, the evolved associations of the species as a whole, as represented by the frequencies of partnerships in the population of individuals for that species, is a continuous variable (to the resolution of 1/*population-size*). We use the term 'association' to refer to this population-level inter-species average and reserve the word 'partnership' for the binary relationships specified by the genotype of an individual. The meaning of the binary partnership vector for an individual is simply that its fitness, already a function of the environmental context, is modified by the inclusion of its symbiotic partners into that context. Specifically, the fitness of an individual genotype, g, belonging to species, s, given a context, E, is defined as $fitness(g, E) = \Delta e(E, s+S) = e(E+s+S) - e(E)$, where S is the set of species that g specifies as partners.

Using the components introduced above, illustrated in Figure 1, our model operates as defined in Figure 2.

A species, *s*: `-------0--`

May contain an individual genotype: `<0001100100,0100000010>`[#]

This example individual specifies partnerships with the following 5 species:

`----0-----`, `-----0----`, `-------0--`[†], `-1--------`, `--------1-`

So, if this individual is placed into an environmental context, it and these partner species will be present: i.e. s+S = `-1--00-01-`.

For example, if this individual is placed into E= `1000100000`, with e(E)=α.

It will create E+S+s=1**1**00**00**0**01**0, with e(E+S+s)=β.

And it will receive a fitness of Δe(E, S+s) = e(E+S+s) - e(E) = β-α.

Fig. 1. An individual, its partners and its fitness in an environmental context. ([†]For implementational convenience, each individual specifies a partnership with itself. [#]A comma indicates separation of 0-partnerships from 1-partnerships.)

```
Initialise Ecosystem containing 2N species, $s_1 \ldots s_{2N}$.
    For each species, $s_n$, initialise P individuals, $g_{n1} \ldots g_{nP}$.
        For each individual, $g_n$, initialise 2N associations: $a_{i=n}=1$, $a_{i \neq n}=0$.   (notes 1 & 2)

Until (stopping-criterion) evolve species:
    For i from 1 to N:  $E_i$=rand({0,1}). //create random context E.
    t=1. // counter to decide when to reinitialise the context.
    //Evaluate g in context E.
        For all, g, from Ecosystem in random order:
        E'=add(E,g).  (note 2)
        Fit(g) = e(E') - e(E).
    //Update environmental context
        If (Fit(g) > 0) then {E=E'. t=1.} else t++.
        If (t>T) {For i from 1 to N:  $E_i$=rand({0,1}). t=1.}
    For each species s: s=reproduce(s).

add(E,S) → E':  // add individual, with partners, to ecosystem state to create E'.
    For n from 1 to N:
        If (($a_n$==1) and ($a_{n+N}$==0)) $E'_n$=0. (note 4)
        If (($a_n$==0) and ($a_{n+N}$==1)) $E'_n$=1.
        If (($a_n$==1) and ($a_{n+N}$==1)) $E'_n$=rand({0,1}).  (note 5)

reproduce(s) → s': // reproduce all the individuals in a species s
    For p from 1 to P:
        Select g1 and g2 from s with uniform probability (note 6).
        If (Fit(g1)>Fit(g2)) $s'_n$= mutate(g1) else $s'_n$= mutate(g2).
```

Fig. 2. Model details. 1) Initially, each species associates with itself only. 2) Associating with self makes implementation of E+S+s identical to E+S. 3) This insertion of a species into the ecological state is cumbersome because there are 2*N* species that fit into *N* niches. 5) If an individual associates with '1'-species and '0'-species that are mutually exclusive (i.e. occupy the same niche) – then either species is added to E with equal probability. 6) Individuals specifying deleterious partnerships (negative fitness) have probability 0 of reproducing, but it is important that individuals specifying neutral partnerships have non-zero probability to reproduce. For the following experiments, *N*=50, *P*=100, and *T*=5*N* was sufficient to ensure an environmental context found a local ESS before being reinitialised. Mutation is single bit-flip.

2.1 A Poly-ESS Ecological Dynamics

We define the energy of an environmental state, *E*, as a sum over *B* copies of the sub-function, *f*, applied to disjoint subsets of species as follows:

$$e(E)=\sum_{b=0}^{B-1} f(\mathrm{s}_{\mathrm{bk+1}}, \mathrm{s}_{\mathrm{bk+2}}, \ldots \mathrm{s}_{\mathrm{(b+1)k\text{-}1}}) \quad \text{where} \quad f(G)=\begin{cases} 2, & \textit{if } G=N, \\ \max\left(\dfrac{1}{1+G}, \dfrac{1}{1+(N-G)}\right) & \textit{otherwise.} \end{cases}$$

where *B* is the number of sub-functions, and *k*=*N*/*B* is the number of species in each sub-function. For convenience *f* is defined as a function of *G*, the number of 1-species in the subset of species. *f*, defines a simple 'U-shaped' energy function with local optima at all-0s and all-1s, but all-1s has higher energy than all-0s. Concatenating *B* of these U-shaped sub-functions creates a poly-attractor system. Five subsystems with two attractors each, as used in the following experiments, creates a system with 2^5=32 point attractors (local optima in the energy function, corresponding to ESSs with *N* species each). For *N*=25, *B*=5, *k*=10, a local attractor under individual selection is: 11111111111111111111000000000011111111110000000000. The attractor with the globally-maximal *e*-value is simply the concatenation of the superior solution to each sub-system, i.e. 11111111111111111111111111111111111111111111111111. However, which of the two-possible local 'sub-attractors' for each sub-system (e.g. ...1111111111... or ...0000000000...) will be found depends (under individual selection) on whether the initial environmental conditions have type-0s or type-1s in the majority. The all-type-1 attractor for each sub-system is thus found with probability 0.5 from a random initial condition and the probability of finding the global-maximal energy attractor is 0.5^B=1/32. (This poly-attractor system is identical to a building-block function used in (19) to show sexual recombination permits selection on subfunctions if genetic linkage is 'tight' – but here we evolve useful linkages.)

3 Results and Discussion

Figure 3 (left) shows that under individual selection (before associations are evolved) different attractors are found (categorised by *G*). The globally-maximal energy attractor is not found in any of the 16 samples depicted. Figure 3 (right) shows that after associations have evolved the globally optimal attractor is being reached in every instance of the local ecological dynamics, regardless of the random initial conditions of the environmental context. Thus the basin of attraction of the globally optimal species configuration now absorbs the entire space of possible initial species configurations (Figure 4).

To examine the evolved partnerships that have enabled these changes in the ecological dynamics, we can display an association matrix, Figure 5. Figure 5 (left) clearly shows not only that the majority of evolved associations are correct (between species of the same type) but also that they are correctly restricted to partnerships between species in the same sub-systems not across sub-systems. The evolution of partnerships is therefore successful at identifying sub-systems correctly, and identifying correct (single-type) partnerships

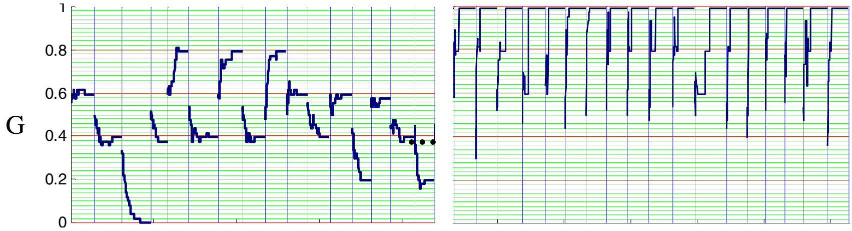
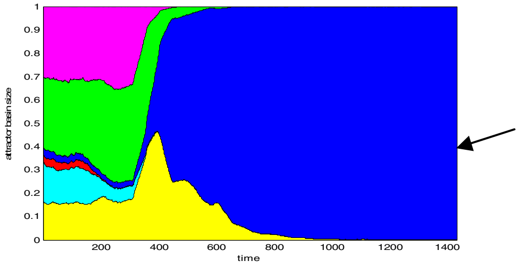


Fig. 3. Ecosystem dynamics: Left) before associations evolve (initial 10,000 time steps), Right) after associations evolve (around $2.8 \cdot 10^7$ time steps). G=1 → globally optimal attractor, G=0.2,0.4,0.6,0.8, 0.0 → other local attractors. Vertical lines indicate points at which a new random initial ecological condition is created.



The proportion of initial conditions that reach the globally optimal attractor is intially 1/32 and eventually becomes 1.

Fig. 4. Change in size of basins of attraction for different attractor classes over evolutionary time: Shades indicate attractor classes grouped by energy values.

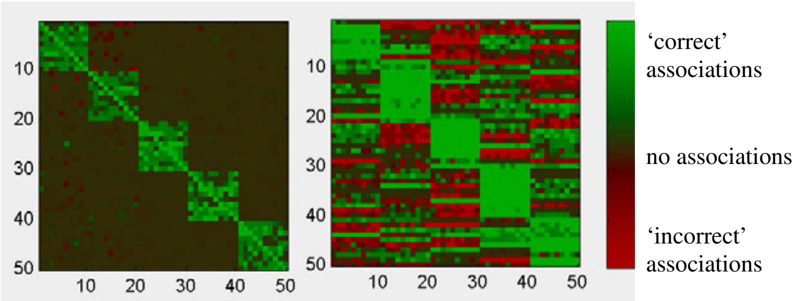


Fig. 5. Evolved associations. Pixel (*i,j*) depicts the strength and correctness (i.e. 1s with 1s, and 0s with 0s) of the associations between the species i and j (and species i+N and j+N). Left) associations in the main experiment reveal the modularity of the fitness dependencies defined in the energy function. Right) a control experiment fails to separate modules, see text.

within those subsystems. Figure 5, right shows that the ability to evolve these correct associations is dependent on the separation of timescales. Specifically, if ecological dynamics are not allowed to settle to a local attractor (by setting T=1 in Figure 2), i.e. partnerships evolve in arbitrary ecological contexts, then although they find useful associations within sub-systems, they find incorrect associations between sub-systems.

These results show that individual selection for symbiotic relationships is capable of creating groups that are adaptively significant. After the relationships have evolved, the only attractor of the ecological dynamics is the attractor with the maximal energy. This is surprisingly 'cooperative' since ecological energy corresponds to collective fitness whereas individual selection should just go to the local ESS. The selective pressures that cause individuals to form these groups has two possible components: a) When the ecological context is not yet at an ESS, an individual that brings with it a partner that accelerates approach to the ESS is fitter than one that does not. Thus *directional* selection on two species promotes symbiosis between them (see (20) for an analogous argument regarding "relational QTLs"). b) When the ecological context is already at an ESS, an individual that brings with it a partner that is also part of the ESS has the same fitness as one that does not (because the species is already present). But an individual that brings a partner that is not part of the ESS will have negative Δe – the partnership is deleterious because it attempts to introduce a species that is selected against in that context. Thus *stabilising* selection on two species also promotes symbiosis between them albeit in a rather subtle manner.

We suggest that the former direct effect is less significant than the latter subtle effect given that the ecosystem spends most of its time at ESSs. This implies that the fommon form of evolved symbioses is to create associations between species that co-occur most often, and suggests that relationship formation in ecosystems will be basically Hebbian (15,18,21,22) – 'species that fire together wire together'. This has the effect of reinforcing the future co-occurrence of species that already co-occur, and enlarges the basin of attraction for those species combinations in the same manner as Hebbian learning forms an associative memory (18,22). Note that the groups that form do not represent an entire N-species ESS but only contain 10 species each (Figure 5) as per the interactions in the energy function. These small groups are both sufficient and selectively efficient in the sense that they create B independent competitions between the two sub-ESSs in each sub-function rather than a single competition between all 2^B complete ESSs (19,24). These small groups form because the co-occurrence of species within each sub-function is more reliable than the co-occurrence of species in different sub-functions. In (13) we provide a model where we assume that relationships form in a manner that reflects species co-occurrence at ESSs and show that this is sufficient to produce the same effects on attractors as those shown here. Using this abstraction we are also able to assess the scalability of the effect and show that it can evolve rare, high-fitness complexes that are unevolvable via non-associative evolution. This suggests a scalable optimisation method for automatic problem decomposition (24), creating algorithmic leverage similar to that demonstrated by (25).

How does individual selection create higher-level selection? Well, from one point of view it doesn't. If we take into account the combined genetic space of characters, both those addressed directly in the energy function and the genetic loci that control partnerships, then all that happens in our model is that natural selection finds a local attractor in this space. It is only when we pretend not to know about the evolved partnerships, and examine the attractors in the energy function alone, that we see group selection. However, this separation is biologically meaningful and relates to the separation of timescales. That is, the most obvious characteristics of species are those that are under direct selection – the ones whose frequencies are affected by selection on short timescales – the ecological population dynamics. But less obvious characteristics are simultaneously under indirect selection – characters that affect co-location of species for example. These change more slowly, over genetic evolution timescales rather than population dynamic timescales (15). When both systems are taken into account, individual selection explains all the observations (if it did not, we would not be satisfied that an evolutionary explanation had been provided). Specifically, partnerships form when group selection is in alignment with individual selection (at ESSs), but in multi-ESS games, these same groupings can cause selection that acts in opposition to (non-associative) individual selection and alter future selective trajectories when individuals are far from that ESS.

Because the indirectly selected characters only have fitness consequences via the directly selected characters their evolution is characterisable by statistics such as co-occurrence of the directly selected characters. This produces systematic consequences on the attractors of directly selected characters – i.e. they enlarge attractors for species combinations reliably found at ESSs. This is equivalent to effecting higher-level selection on these combinations of species. Thus in our opinion, the two types of language - ‘higher levels of selection are created’ and ‘it is all explained by individual selection’ – are entirely reconcilable.

Acknowledgements. Chris Buckley, Seth Bullock, Jason Noble, Ivor Simpson, Nicholas Hayes, Mike Streatfield, David Iclanzan, Chrisantha Fernando.

References

1. Maynard Smith, J., Szathmary, E.: The major transitions in evolution, Oxford (1995)
2. Margulis, L.: Origins of species: acquired genomes and individuality. BioSystems 31(2-3), 121–125 (1993)
3. Wilson, D.S.: A theory of group selection. PNAS 72(1), 143–146 (1975)
4. Wilson, D.S.: The Natural Selection of Populations and Communities. Benjamin/ Cummings, New York (1980)
5. Wilson, D.S., Dugatkin, L.A.: Group selection and assortative interactions. The American Naturalist 149(2), 336–351 (1997)
6. Powers, S.T., Mills, R., Penn, A.S., Watson, R.A.: Social niche construction provides an adaptive explanation for new levels of individuality (ABSTRACT). Submitted to Levels of Selection and Individuality in Evolution Workshop, ECAL 2009 (2009)
7. Powers, S.T., Watson, R.A.: Evolution of Individual Group Size Preference can Increase Group-level Selection and Cooperation. In: ECAL 2009 (2009) (to appear)
8. Powers, S.T., Penn, A.S., Watson, R.A.: Individual Selection for Cooperative Group Formation. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 585–594. Springer, Heidelberg (2007)
9. Snowdon, J., Powers, S.T., Watson, R.A.: Moderate contact between sub-populations promotes evolved assortativity enabling group selection. In: ECAL 2009 (2009) (to appear)
10. Maynard Smith, J.: Evolution and the Theory of Games, Cambridge (1982)
11. Axelrod, R.: The Complexity of Cooperation. Princeton (1997)
12. Hamilton, W.D.: The Genetical Evolution of Social Behaviour I and II. J. Theor. Biol. 7, 1–52 (1964)

13. Mills, R., Watson, R.A.: Symbiosis Enables the Evolution of Rare Complexes in Structured Environments. In: ECAL 2009 (2009) (to appear)
14. Noonburg, V.W.: A Neural Network Modeled by an Adaptive Lotka-Volterra System. SIAM Journal on Applied Mathematics 49(6), 1779–1792 (1989)
15. Poderoso, F.C., Fontanari, J.F.: Model ecosystem with variable interspecies interactions. J. Phys. A: Math. Theor. 40, 8723–8738 (2007)
16. Moran, N.A.: Symbiosis as an adaptive process and source of phenotypic complexity. PNAS USA 10, 1073 (2007)
17. Strogatz: Non-Linear Dynamics and Chaos, Westview (1994)
18. Watson, R.A., Buckley, C.L., Mills, R.: The Effect of Hebbian Learning on Optimisation in Hopfield Networks. Technical Report, ECS, University of Southampton (2009)
19. Watson, R.A., Jansen, T.: A Building-Block Royal Road Where Crossover is Provably Essential. In: GECCO 2007, pp. 1452–1459 (2007)
20. Wagner, G.P., Pavlicev, M., Cheverud, J.M.: The road to modularity. Nature Reviews Genetics 8, 921–931 (2007)
21. Watson, R.A., Mills, R., Buckley, C.L., Palmius, N., Powers, S.T., Penn, A.S.: Hebbian Learning in Ecosystems: Species that fire together wire together (ABSTRACT). In: ECAL 2009 (2009b) (to appear)
22. Watson, R.A., Buckley, C.L., Mills, R.: Implicit Hebbian Learning and Distributed Optimisation in Complex Adaptive Systems (in prep)
23. Watson, R.A., Pollack, J.B.: Modular Interdependency in Complex Dynamical Systems. Artificial Life 11(4), 445–457 (2005)
24. Mills, R., Watson, R.A.: A symbiosis Algorithm for automatic problem decomposition (2009) (in prep)
25. Iclanzan, D., Dumitrescu, D.: Overcoming hierarchical difficulty by hill-climbing the building block structure. In: GECCO 2007, pp. 1256–1263 (2007)

The Effect of Group Size and Frequency-of-Encounter on the Evolution of Cooperation

Steve Phelps[1], Gabriel Nevarez[2], and Andrew Howes[2]

[1] Centre for Computational Finance and Economic Agents (CCFEA),
University of Essex, UK
sphelps@essex.ac.uk
[2] Manchester Business School, University of Manchester, UK

Abstract. We introduce a model of the evolution of cooperation in groups which incorporates both conditional direct-reciprocity ("tit-for-tat"), and indirect-reciprocity based on public reputation ("conspicuous altruism"). We use ALife methods to quantitatively assess the effect of changing the group size and the frequency with which other group members are encountered. We find that for moderately sized groups, although conspicuous altruism plays an important role in enabling cooperation, it fails to prevent an exponential increase in the level of the defectors as the group size is increased, suggesting that economic factors may limit group size for cooperative ecological tasks such as foraging.

1 The Model

Pairs of agents $(a_i, a_j) : i \neq j$ are drawn at random from $A = \{a_1, a_2, \ldots a_n\}$, and engage in bouts of grooming at different time periods $t \in \{0, 1, \ldots N\}$. We refer to n as the *group size* and N as the *frequency-of-encounter*.

At each time period t the groomer a_i may choose to invest a certain amount of effort $u_{(i,j,t)} \in [0, U] \subset \mathbb{R}$ in grooming their partner a_j, where $U \in \mathbb{R}$ is a parameter determining the maximum rate of grooming. This results in a negative fitness payoff $-u$ to the groomer, and a positive fitness payoff ku to the partner a_j:

$$\phi_{(j,t+1)} = \phi_{(j,t)} + k \cdot u_{(i,j,t)}$$
$$\phi_{(i,t+1)} = \phi_{(i,t)} - u_{(i,j,t)}$$

where $\phi_{(i,t)} \in \mathbb{R}$ denotes the fitness of agent a_i at time t, and $k \in \mathbb{R}$ is a constant parameter.

In an ecological context, the positive fitness payoff ku might represent, for example, the fitness gains from parasite elimination, whereas the fitness penalty $-u$ would represent the opportunity cost of foregoing other activities, such as foraging, during the time u allocated for grooming.

Since we are interested in the *evolution* of cooperation, we analyse outcomes in which agents choose values of u that maximise their own fitness ϕ_i. Provided

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 37–44, 2011.

that $k > 1$, over many bouts of interaction it is possible for agents to enter into reciprocal relationships that are mutually-beneficial, since the groomer's initial cost u may be reciprocated with ku yielding a net benefit $ku - u = u(k-1)$. Provided that agents reciprocate, they can increase their net benefit by investing larger values of u. However, by increasing their investment they put themselves more at risk from exploitation, since just as in the alternating prisoner's dilemma [5], defection is the dominant strategy if the total number of bouts N is known: the optimal behavior is to accept the benefits of being groomed without investing in grooming in return. In the case where N is *unknown*, and the number of agents is $n = 2$, it is well known that conditional reciprocation is one of several evolutionary-stable solutions in the form of the so-called *tit-for-tat* strategy which copies the action that the opposing agent chose in the preceding bout at $t - 1$ [4]. However, this result does not generalise to larger groups $n > 2$.

Nowak and Sigmund [7] demonstrate that reciprocity can emerge *indirectly* in large groups, provided that information about each agent's history of actions is summarised and made publicly available in the form of a reputation or "image-score" $r_{(i,t)} \in [r_{\min}, r_{\max}] \subset \mathbb{Z}$. The image-score r_i summarises the propensity-to-cooperate of agent a_i. As in the Nowak and Sigmund model, image scores in our model are initialised $\forall_i \; r_{(i,0)} = 0$ and are bound at $r_{\min} = -5$ and $r_{\max} = 5$. An agent's image score is incremented at $t + 1$ if the agent invests a non-zero amount at time t, otherwise it is decremented:

$$r_{(i,t+1)} = \begin{cases} \min(r_{(i,t)} + 1, \; r_{\max}) & : \quad u_{(i,x,t)} > 0 \\ \max(r_{(i,t)} - 1, \; r_{\min}) & : \quad u_{(i,x,t)} = 0 \end{cases}$$

and agents invest conditionally on their partner's image score:

$$u_{(i,j,t)} = \begin{cases} \gamma & : \quad r_{(j,t)} \geq \sigma_i \\ 0 & : \quad r_{(j,t)} < \sigma_i \end{cases}$$

where σ_i is a parameter determining the threshold image score above which agent a_i will cooperate, and $\gamma \in \mathbb{R}$ is a global parameter (as in [7] we use $\gamma = 10^{-1}$ and $k = 10$).

Nowak and Sigmund [6] demonstrate that widespread defection is avoided if, and only if, the initial proportion of agents using a discriminatory[1] strategy is above a critical value, implying that strategies based on indirect reciprocity via reputation are an essential prerequisite for the evolution of cooperation in large groups.

We are interested in the effect of group size n and interaction frequency N on the evolution of cooperation. The analytical model of Nowak and Sigmund [6] assumes: a) that the group size n is large enough relative to N that strategies based on private history, such as *tit-for-tat*, are irrelevant (since the probability of encountering previous partners is very small); and b) that the we do not need to take into account the fact that an agent cannot cooperate with itself when

[1] Discriminatory strategies cooperate only if their partner's image score is non-negative, that is: $\sigma_i = 0$.

calculating the probability with which any given agent is likely to encounter a particular strategy. However, in order to model changes in group size, and hence interaction in smaller groups, it is necessary to drop both of these assumptions. The resulting model is more complicated, and it is difficult to derive closed-form solutions for the equilibrium behaviour. Therefore we use ALife simulation to estimate payoffs, and numerical methods to compute asymptotic outcomes, as described in the next section.

2 Methodology

In order to study the evolution of populations of agents using the above strategies, we use both ALife methods and mathematical modeling based on evolutionary game-theory. However, rather than considering pairs of agents, our analysis concerns interactions amongst *groups* of size $n > 2$ assembled from a larger population of individuals. The resulting game-theoretic analysis is complicated by the fact that this results in a many-player game, which presents issues of tractability for the standard methods for computing the equilibria of normal-form games. A popular ALife approach to this issue is to use *Co*-evolutionary algorithms [3,4]. In a co-evolutionary optimisation, the fitness of individuals in the population is evaluated relative to one another in joint interactions (similarly to payoffs in a strategic game), and it is suggested that in certain circumstances the converged population is an approximate Nash solution to the underlying game; that is, the stable states, or equilibria, of the co-evolutionary process are related to the evolutionary stable strategies (ESS) of the corresponding game. However, there are many caveats to interpreting the equilibrium states of standard co-evolutionary algorithms as approximations of game-theoretic equilibria, as discussed in detail by Sevan Ficici [1,2]. In order to address this issue, we adopt a methodology called *empirical game-theory* [10,12], which uses a combination of simulation and rigorous game-theoretic analysis. The empirical game-theory method uses a *heuristic* payoff matrix which is calibrated by running many simulations, as detailed below.

We can make one important simplification by assuming that the game is symmetric, and therefore that the payoff to a given strategy depends only on the *number* of agents within the group adopting each strategy. Thus for a game with j strategies, we represent entries in the payoff matrix as vectors of the form $\boldsymbol{p} = (p_1, \ldots, p_j)$ where p_i specifies the number of agents who are playing the i^{th} strategy. Each entry $\boldsymbol{p} \in P$ is mapped onto an outcome vector $\boldsymbol{q} \in Q$ of the form $\boldsymbol{q} = (q_1, \ldots, q_j)$ where q_i specifies the expected payoff to the i^{th} strategy.

For each entry in the payoff matrix we estimate the expected payoff to each strategy by running 10^5 ALife simulations and taking the mean[2] fitness.

With estimates of the payoffs to each strategy in hand, we are in a position to model the evolution of populations of agents using these strategies. In our evolutionary model, we do not restrict reproduction to within-group mating;

[2] We take the average fitness of every agent adopting the strategy for which we are calculating the payoff, and then also average across simulations.

rather, we consider a larger population which temporarily forms groups of size n in order to perform some ecological task. Thus we use the standard replicator dynamics equation [11] to model how the frequency of each strategy in the larger population changes over time in response to the within-group payoffs:

$$\dot{m}_i = [u(e_i, \boldsymbol{m}) - u(\boldsymbol{m}, \boldsymbol{m})]\, m_i \tag{1}$$

where $\boldsymbol{m}$ is a mixed-strategy vector, $u(\boldsymbol{m}, \boldsymbol{m})$ is the mean payoff when all players play $\boldsymbol{m}$, and $u(e_i, \boldsymbol{m})$ is the average payoff to pure strategy i when all players play $\boldsymbol{m}$, and $\dot{m}_i$ is the first derivative of m_i with respect to time. Strategies that gain above-average payoff become more likely to be played, and this equation models a simple co-evolutionary process of adaptation. Since mixed strategies represent population frequencies, the components of $\boldsymbol{m}$ sum to one. The geometric corollary of this is that the vectors $\boldsymbol{m}$ lie in the *unit-simplex* $\triangle^{j-1} = \{\boldsymbol{x} \in \mathbb{R}^j : \sum_{i=1}^{j} x_i = 1\}$. In the case of $j = 3$ strategies the unit-simplex $\triangle^2$ is a *two*-dimensional triangle embedded in three-dimensional space which passes through the coordinates corresponding to pure strategy mixes: $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. We shall use a two dimensional projection of this triangle to visualise the population dynamics in the next section[3].

In our experiments we solve this system numerically: we choose 10^3 randomly sampled initial values which are chosen uniformly from the unit simplex [9], and for each of these initial mixed-strategies we solve Equation 1 as an initial value problem using MATLAB's `ode15s` solver [8]. This results in 10^3 trajectories which either terminate at stationary points, or enter cycles.

We consider $j = 5$ strategies:

1. C which cooperates unconditionally ($\sigma_i = r_{min}$);
2. D which defects unconditionally ($\sigma_i = r_{max} + 1$);
3. S which cooperates conditionally with agents who have a good reputation ($\sigma_i = 0$) but cooperates unconditionally in the first round (when reputations have not yet been established);
4. Sd which cooperates conditionally ($\sigma_i = 0$) but defects unconditionally in the first round of play;
5. $T4T$ which cooperates conditionally with agents who have cooperated in previous rounds, and cooperates unconditionally against unseen opponents.

3 Results

Initially we restrict attention to $j = 3$ strategies and 10^2 initial values, allowing us to more easily visualise the population dynamics and to compare our results with that of Nowak and Sigmund [6] (who assume a large group size n relative to N).

[3] See [11, pp. 3–7] for a more detailed exposition of the geometry of mixed-strategy spaces.

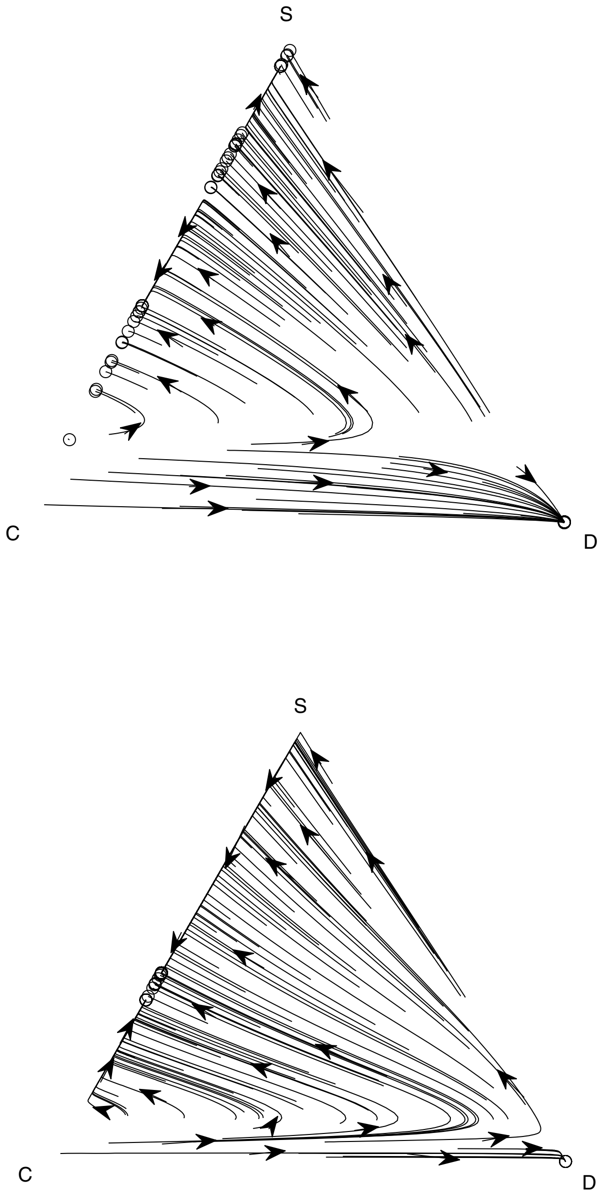


Fig. 1. Direction field for $n = 10$ agents and $N = 13$ pairwise interactions per generation (above) compared with $N = 100$ (below). C denotes unconditional altruists, D unconditional defectors and S discriminators who cooperate in the first round. Each line represents a trajectory whose termination is represented by an open circle. The arrows show the direction of change.

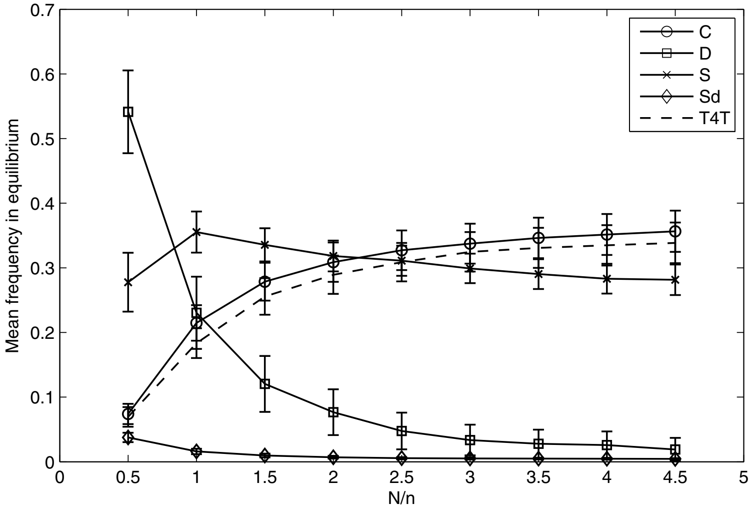


Fig. 2. Mean frequency of each strategy in equilibrium as the number of pairwise interactions per generation N is increased relative to n. The error bars show the confidence interval for $p = 0.05$. C denotes the proportion of unconditional altruists; D unconditional defectors; S discriminators who cooperate in the first round; Sd discriminators who defect in the first round; $T4T$ the tit-for-tat strategy.

When we exclude discriminators and consider only cooperator (C), defectors (D) and tit-for-tat ($T4T$) we find that defection is the dominant strategy regardless of N, implying that conditional reciprocity cannot sustain group cooperation in the absence of reputation.

We obtain more subtle results when we introduce reputation-based strategies. Figure 1 shows the phase diagram for the population frequencies when we analyse the interaction between cooperators (C), defectors (D) and discriminators (S) when we have a small group of $n = 10$ agents. As in [6] we find that a minimum initial frequency of discriminators (y axis) is necessary to prevent widespread convergence to the defection strategy (in the bottom right of the simplex). However, the results of our model differ in two important respects.

Firstly, when the critical threshold of discriminators is reached, our model results in various stationary mixes of discriminators and cooperators with a total absence of defection, and no limit cycles. This is in contrast to [6] where the population cycles endlessly between all three strategies if the critical threshold is exceeded.

Secondly, the behaviour of our model is sensitive to the number of pairwise interactions per generation N: as N is increased from $N = 13$ to $N = 100$ we see that the basin of attraction of the pure defection equilibrium is significantly decreased, and correspondingly the critical threshold of initial discriminators necessary to avoid widespread defection. Our intuitive interpretation of these results is that defection is less likely[4] as we increase the frequency of interaction relative to the group size.

As discussed in Section 1, if we increase N relative to n we need to consider the effect of strategies that take into account private interaction history as well as strategies that are based on public reputation. Figure 2 shows the mean frequency in equilibrium of each strategy when we analyse all five strategies and systematically vary N while holding n fixed. We plot the equilibrium population frequency against $\frac{N}{n}$, and obtain the same graph for both $n = 10$ and $n = 20$ agents. This suggests that the ratio $\frac{N}{n}$ determines the asymptotic behaviour.

4 Discussion

The frequency of both unconditional cooperation and discrimination increases with N, and these strategies become more prevalent than discriminators for $\frac{N}{n} > 2$. As we would intuitively expect, for $\frac{N}{n} > 1$ discriminators become more prevalent as we increase group size or decrease frequency of interaction. However, this is not sufficient to prevent free-riding. Most striking is that the likelihood of defection decreases exponentially as we increase the number of interactions per generation N. Correspondingly, as we increase the group size n we observe an *exponential increase* in the level of defection.

If we consider the possibility of inter-group competition and hence group selection, then since the expected frequency of defectors in equilibrium determines the per-capita fitness of the agents in the overall population[5], we can interpret our results as showing how group fitness changes as a function of group size (n) and frequency-of-encounter (N). That is, for any given N we can determine the optimum group size n.

Our results indicate that the stylised cooperation task described by our model introduces a very strong selection pressure for smaller group sizes. Of course, this task is not the only ecological task which influences per-capita fitness for any given species. For example, our model could be used in conjunction with optimal foraging models to derive a comprehensive model of optimum group size for a particular species in a particular niche. Our main contribution is to highlight that economic factors play a significant role in determining optimal group size, when other ecological tasks such as foraging favour group sizes that are relatively small compared with the frequency-of-encounter.

5 Conclusion

Our model predicts that neither reputation nor conditional punishment are sufficient to prevent free-riding as the group size increases. Although both types of strategy play an important role, as the group size increases the level of

[4] Assuming that all points in the simplex are equally likely as initial values.

[5] In the absence of defection all other strategies are able to gain the maximum available surplus.

conspicuous altruism based on reputation rises, but defection rises faster. Thus, when other tasks already favour smaller groups, we predict that economic factors will limit the maximum group size independently of the group size favoured by other niche-specific tasks.

Acknowledgements

We are grateful for the comments of the anonymous reviewers, and for financial support received from the UK EPSRC through the *Theory for Evolving Sociocognitive Systems* project (EP/D05088X/1).

References

1. Ficici, S.G., Pollack, J.B.: Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In: Proceedings of of ALIFE-6 (1998)
2. Ficici, S.G., Pollack, J.B.: A game-theoretic approach to the simple coevolutionary algorithm. In: Schwefel, H.-P., Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J. (eds.) PPSN VI 2000. LNCS, vol. 1917, pp. 16–20. Springer, Heidelberg (2000)
3. Hillis, W.D.: Co-evolving parasites improve simulated evolution as an optimization procedure. In: Langton, et al. (eds.) Proceedings of ALIFE-2, pp. 313–324. Addison Wesley, Reading (1992)
4. Miller, J.H.: The coevolution of automata in the repeated Prisoner's Dilemma. Journal of Economic Behavior and Organization 29(1), 87–112 (1996)
5. Nowak, M.A., Sigmund, K.: The alternating prisoner's dilemma. Journal of Theoretical Biology 168, 219–226 (1994)
6. Nowak, M.A., Sigmund, K.: The dynamics of indirect reciprocity. Journal of Theoretical Biology 194(4), 561–574 (1998)
7. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity by image scoring. Nature 383, 537–577 (1998)
8. Shampine, L.F., Reichelt, M.W.: The MATLAB ODE suite (2009), `http://www.mathworks.com/access/helpdesk/help/pdf_doc/otherdocs/ode_suite.pdf`
9. Stafford, R.: Random vectors with fixed sum (January 2006), `http://www.mathworks.com/matlabcentral/fileexchange/9700`
10. Walsh, W.E., Das, R., Tesauro, G., Kephart, J.O.: Analyzing complex strategic interactions in multi-agent games. In: AAAI 2002 Workshop on Game Theoretic and Decision Theoretic Agents (2002), `http://wewalsh.com/papers/MultiAgentGame.pdf`
11. Weibull, J.W.: Evolutionary Game Theory. MIT Press, Cambridge (1997) (First MIT Press edition)
12. Wellman, M.P.: Methods for Empirical Game-Theoretic Analysis. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence, pp. 1152–1155 (2006)

Moderate Contact between Sub-populations Promotes Evolved Assortativity Enabling Group Selection

James R. Snowdon, Simon T. Powers, and Richard A. Watson

School of Electronics and Computer Science,
University of Southampton, U.K.
raw@ecs.soton.ac.uk

Abstract. Group selection is easily observed when spatial group structure is imposed on a population. In fact, spatial structure is just a means of providing assortative interactions such that the benefits of cooperating are delivered to other cooperators more than to selfish individuals. In principle, assortative interactions could be supported by individually adapted traits without physical grouping. But this possibility seems to be ruled-out because any 'marker' that cooperators used for this purpose could be adopted by selfish individuals also. However, here we show that stable assortative marking can evolve when sub-populations at different evolutionarily stable strategies (ESSs) are brought into contact. Interestingly, if they are brought into contact too quickly, individual selection causes loss of behavioural diversity before assortative markers have a chance to evolve. But if they are brought into contact slowly, moderate initial mixing between sub-populations produces a pressure to evolve traits that facilitate assortative interactions. Once assortative interactions have become established, group competition between the two ESSs is facilitated without any spatial group structure. This process thus illustrates conditions where individual selection canalises groups that are initially spatially defined into stable groups that compete without the need for continued spatial separation.

1 Introduction

The perspective of group selection is often used to explain altruistic behaviour. Sober and Wilson provide the definition that 'a behaviour is altruistic when it increases the fitness of others and decreases the fitness of the actor' [1], which can appear unsupportable by traditional theories of natural selection as echoed by Dawkins [2], who claims that for any gene to survive they must promote themselves at the expense of others. A gene which supports altruistic behaviour would quickly be exploited to extinction by selfish cheaters.

As such selfishness can be described as an Evolutionarily Stable Strategy (ESS). A behaviour is considered to be an ESS if when all individuals within a population adopt a particular behavioural strategy no other type can successfully invade [3]. Altruism is therefore not an ESS since it can be invaded by cheats

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 45–52, 2011.

who will exploit the benefit awarded to them and not reciprocate. For natural selection to occur at any level, there must exist a fitness variance between entities [4], where an entity can be an individual or a group composed of individuals forming a meta-population. In this one ESS system all groups will move towards a situation where every individual holds the same behaviour, resulting in no variance for natural selection to act upon.

Wilson [5] uses groups comprised of altruists and cheaters to provide a setting in which altruism can prevail. Each group reproducing individually would ultimately be drawn towards the one stable attractor which is the all selfish ESS, but groups are dispersed prior to this occurring and the progeny are mixed. New groups are composed of random samples and the aggregation and dispersal process is repeated indefinitely. Since groups which contain more altruists grow at a faster rate than groups composed of majority cheats, the effect is that the net proportion of altruists rises. However, when practically assessing the model it is found that between group variance in the frequencies of cheats and selfish types is required to be extremely high, and as such a stronger effect is observed through taking an extremely small sample of the population when creating each new group in the aggregation process. This limit of small group sizes coupled with the aggregation and dispersal movement limits the applicability to real world situations. Although the conditions for such altruism to evolve are restrictive, Powers et al. [6] have investigated how these conditions can in fact arise by evolution of individual traits that modify aspects of population structure, such as group size.

Wilson [7] studies group level selection in complex meta-communities. His model shows how individuals detrimental to local community productivity can be purged through selection at the community level. This is an example of a multi-ESS system, for the internal community dynamics have many attractors (ESSs) and proves to be more effective than previous altruist/ cheat models, since between-group variance could be preserved even when the groups reached an internal equilibrium. Such a form of group selection is also explored by Boyd and Richerson who consider selection acting among multiple ESSs [8], where each group reaches an ESS holding an inherent fitness thus providing a between group variance. Through competition with other groups based upon a migratory process, the ESSs compete and some are forced into extinction as groups reach the point of highest individual fitness. Their model does not consider the forces which give rise to these groupings, even though they are able to illustrate the selection processes which act at the group level.

We are interested in situations where a form of group selection can occur which is not restricted by the need for constant spatial segregation. Consider a scenario where two spatially separated sub-populations develop different behaviours, A and B, which are *both* ESSs. The sub-populations are then slowly brought into contact, perhaps through natural expansion of each of the sub-populations. One of these behaviours, say A, even though it may have been intrinsically fitter than B, may be lost when the two populations are brought into contact because it fares poorly in interaction with B, because B may be more numerous. In principle, if individuals within the two sub-populations evolved distinguishable markers, such as the secretion of a unique, identifiable chemical which correlated with behaviour and promoted assortative interactions then competition between the groups would be enabled when the two are brought into contact. That is, although A loses to B, A-A wins in interaction with B-B. This would mean that the two sub-populations had formed higher-level units of selection.

A model provided by McElreath, Boyd and Richerson [9] in the domain of cultural evolution shows that individual selection can evolve markers that are correlated with behaviour and promote assortative interactions. Their work does not address the notion that such marking facilitates higher-level selection - they are interested in the promotion of stably coexisting (ethnically marked) groups. But we show that the conditions they illustrate for evolving behaviourally-correlated markers are also suitable to thereby facilitate inter-group competition.

2 A Model of the Evolution of Assortative Markers under Some Degree of Spatial Segregation

Our model is founded upon the work of McElreath et al [9]. We describe individuals as exhibiting a behaviour, labelled A and B, and a 'marker' trait, labelled 1 and 2. This can be interpreted as an externally apparent phenotype used for the purpose of facilitating assortative interactions (as per Boyd et al's work [10]) or any other trait which has the effect of producing assortative interactions, such as a habitat preference [11].

Upon initialization individuals are distributed between two sub-populations, and we start from the scenario where each is already at a stable A or B ESS. The algorithmic operation of the model for each subsequent time step is as follows:

1. **Interactions** - All individuals interact with each other within the same sub-population in a coordination game with a payoff matrix as shown in table 1.

Table 1. Payoff matrix for interactions between individuals harbouring either behaviour A or behaviour B. This system exhibits both A and B as ESSs.

	A	B
A	$1 + \delta + \alpha$	1
B	1	$1 + \delta$

An individual interacting with another holding the same behavioural trait will enjoy an advantage, δ but only the marker can influence which other individual is likely to be interacted with. A parameter e describes assortativity, or marker strength - the probability that an individual will interact with another possessing the same marker. When e=1 individuals will only interact with others of the same marker as themselves, and when e=0 individuals interact totally randomly with no reliance upon marker.

2. **Reproduction** - The proportions, N, of each marker, i, and behaviour, j, within the next generation of individuals changes according to equation 1, where $\sum W_{ij}$ is the total payoff awarded to each behaviour/marker type (A1, A2, B1 and B2) and $\bar{W_{ij}}$ is the average payoff awarded to individuals of that type.
$$N_{ij} = \frac{\sum W_{ij}}{\bar{W_{ij}}} \tag{1}$$
3. **Migration** - Prior to being brought together, there is a degree of migration between the two groups, represented by the parameter m. This relocates a random proportion of each group to within the other, thus representing a metric of spatial segregation. Groups at a value of $m = 0$ implies that no migration exists as groups are completely segregated, and a larger value of m (maximum 0.5) represents a freely mixed population.

In order to generate a between group fitness variance, A-A interactions are given an additional payoff bonus α, and groups are brought into a competitive state by bringing the sub-populations together after a fixed period by setting the inter-group migration to a maximum, $m = 0.5$.

3 Exploring the Behaviour of Two Sub-populations

It would of course be possible to set up the initial conditions of the two groups such that markers correlated with behaviours will be reached regardless of interaction. However, we set the initial conditions of the sub-populations such that they have *different* behaviours in the majority and the *same* marker in the majority. This means that without some selective pressure to cause markers to diversify, both groups will have the same marker and assortative interactions will not be possible. We also set the size of one sub-population to be slightly larger than the other - the sub-population with the inferior behaviour - for similar reasons. That is, we want to identify conditions where the superior behaviour prevails despite being initially disadvantaged. We will refer to this correlation of marker/ behaviour pairs as the evolution of assortative markers.

The simulation was run with the initial parameters shown in table 2, and as will be seen the values of e and m will come under further investigation so these are not fixed.

Crucially groups are initialised such that if the initial migration rate is too high then the B behaviour will overwhelm as before markers have evolved in the

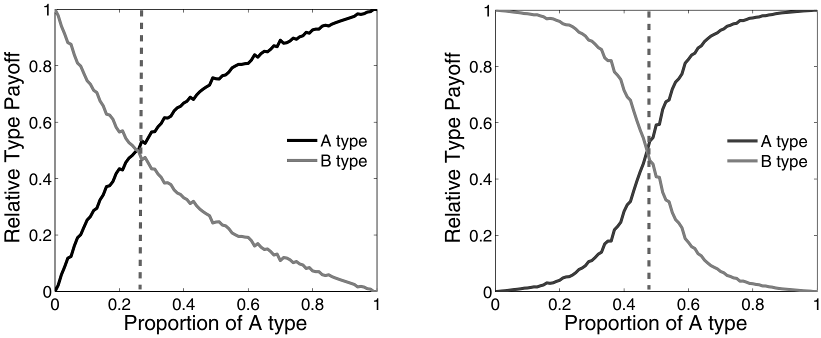
Table 2. Parameters used for the simulation process

Parameter	Value	Parameter	Value
Sub-population 1 Prop. 1 Marker	0.9	Sub-population 2 Prop. 1 Marker	0.6
Sub-population 1 Prop. 2 Marker	0.1	Sub-population 2 Prop. 2 Marker	0.4
Sub-population 1 Size	2000	Sub-population 2 Size	3800
Like-for-like payoff, δ	0.5	A-A payoff bonus, α	0.1

system due to the larger sub-population 2 holding a higher majority of B type. If the migration rate is too low then assortative markers will not evolve despite sub-populations being behaviourally marked. Under these conditions should assortative marking occur, if one of the sub-populations was going to go to the marker 1, we would expect it to be sub-population 1.

When groups are freely mixed after an initial partial mixing phase, we would expect the ultimate results to be predictable through observing the basins of attraction shown in figure 1. If markers have not been able to evolve (for example from a zero migration rate) then we would expect the distribution of each marker type in a sub-population to be random as frequencies would drift. The ultimately winning strategy would be B as when the sub-populations are mixed, the majority - 65% - of the newly formed group would be from the all-B group 2, and this would overwhelm any advantage which A-A receives which would be unable to successfully interact assortatively.

If the migration rate is too high then all individuals in both sub-populations will become type B due to the initial majority of behaviour Bs in the entire population, and so the mixed group would then already be at the all-B attractor. However, if assortative markers have evolved then at $e = 0.5$ despite only 35% of the mixed group being A1 type, the resultant attractor is expected to be A1 as figure 1(b) shows.



(a) Full correlation between behaviour and marker ($e = 1$)

(b) No correlation between behaviour and marker ($e = 0$)

Fig. 1. Relative fitness of A1 and B2 type when in competition in a single population. The line indicates the lowest initial proportion of A1 type required for A1 to reach fixation.

Figure 2 illustrates system behaviour for a fixed $e = 0.5$ and varying migration rates. Figure 2(a) introduces a metric of the polarisation of marking present within the sub-populations as the linkage disequilibrium of marker and behaviour after an appropriate period of mixing. A higher value shows assortative markers have evolved, and 0 shows that such marking has not occurred. It is clear that a region exists where a degree of spatial segregation enables the evolution of such assortative marking.

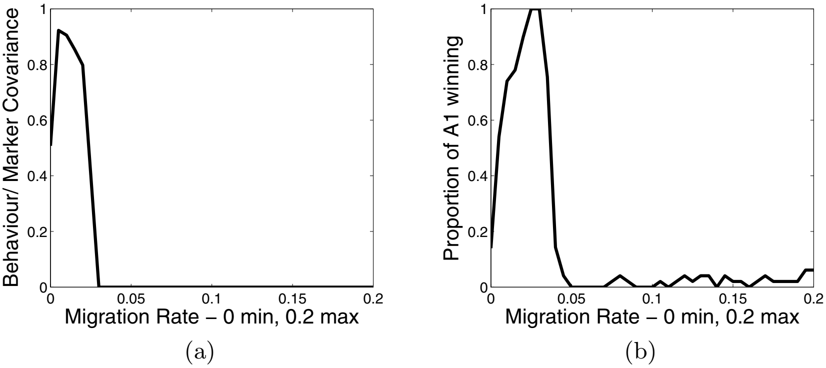


Fig. 2. a) Analysis over a range of spatial segregation values, when $e = 0.5$, displaying marker/ behaviour covariance evolved with spatial segregation after 600 time steps, averaged over 20 runs b) Resulting proportion of A1 types which went on to reach fixation determined from an average of the same 20 runs

Figure 2(b) shows the proportion of runs when the groups were then brought together and the A1 behaviour type went on to reach fixation and drive the B2 behaviour extinct. Again we see that there is a region where A1 is able to outcompete B2 - and this corresponds to the region of polarised markers.

Taking single points from the graph in figure 3, we can observe the internal behaviour between the two sub-populations which leads to the results shown:

- **No initial contact**, $m = 0.0$, shows that because the migration rate is zero there is no pressure for markers to evolve, so when the groups are brought together the effect is that B behaviour types overwhelm A behaviour.
- **Moderate contact**, $m = 0.01$, shows that although sub-population 2 starts with marker 1 in the majority, there is a pressure not to interact with migrants of majority A1 type from sub-population 1. This causes marker 2 to cross over and become marked with behaviour B. When sub-populations are freely mixed this results in A1 winning.
- **High contact**, $m = 0.2$, shows that B behaviour fixates within the entire population due to the higher migration rate, and again there is no pressure for markers to evolve. Accordingly the resultant ‘winner’ is non-polarised behaviour B.

4 Discussion

We have shown that, as per McElreath et al’s work, a small amount of initial mixing between sub-populations that exhibit different behaviours can produce a selective pressure to favour the evolution of markers that are correlated to those behaviours. No mixing and the markers have no function, too much mixing and one of the behaviours is lost before markers evolve; but a period with a small

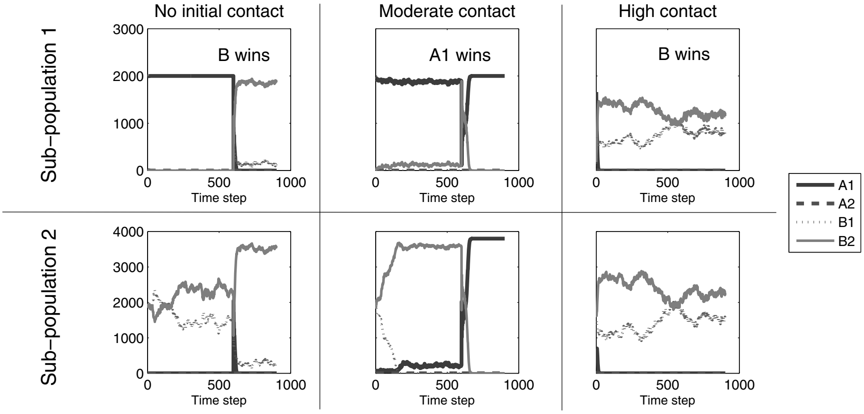


Fig. 3. Illustrating interactions with varying initial contact between sub-populations. Sub-population 1 (above) is fixed at size 2000, and sub-population 2 (below) is fixed at size 3800. At t=600 the sub-populations are mixed through setting inter-group migration to maximum, resulting in the domination of one behaviour.

amount of mixing produces this effect. In essence, this occurs because it enables a period where weak indirect selective pressures on markers can be felt, whilst precluding the strong selective pressures on behaviours that would lose diversity before the markers have evolved. This effect means that partial spatial segregation between sub-populations (but not complete segregation) enables individuals to evolve behaviours that reinforce within-group interactions and we show that the assortative interactions that result facilitate competition between groups when increased mixing occurs. The evolution of markers in this way can be seen as construction of an individual's social environment [12]. Group selection can be facilitated by such a process, as shown here and in other work [13,14].

In conventional altruist/ cheat dynamics involving the evolution of assortative markers there exists a possibility of cheats evolving the phenotypic altruistic marker, such as a green beard [2], signalling altruistic intent without actually posessing a cooperative gene. Because this model holds *multiple* stable ESSs the issue of cheating does not occur, as an individual falsely advertising its' behaviour would receive a lower payoff than if it had been honest.

This model illustrates very simple conditions where individual selection can favour the evolution of traits that support between-group competition. Accordingly, from one point of view, the expectation that individual selection cannot create significant group selection is shown to be false. But, from another point of view, one which takes into account both individual selection on markers and behaviours under these spatial conditions - individual selection explains the outcomes we observe. Indeed, if there were not the case, we would not have provided an evolutionarily explanation at all. Nonetheless, because the markers are different from behaviours in that they only have fitness consequences via their indirect effects on the assortativity of behaviours - we argue that a two scale selection theory is conceptually useful.

References

1. Sober, E., Wilson, D.S.: Unto others: the evolution of psychology and unselfish behavior. Harvard University Press, Cambridge (1998)
2. Dawkins, R.: The Selfish Gene. Oxford University Press, Oxford (1976)
3. Maynard Smith, J.: Evolution and the Theory of Games. Cambridge University Press, Cambridge (1982)
4. Wilson, D.S., Sober, E.: Reintroducing group selection to the human behavioral sciences. Behavioral and Brain Sciences 17(4), 585–654 (1994)
5. Wilson, D.S.: A theory of group selection. PNAS 72(1), 143–146 (1975)
6. Powers, S.T., Watson, R.A.: Evolution of individual group size preferences can increase group-level selection and cooperation. In: Proceedings of the 10th European Conference on Artificial Life (2009) (to appear)
7. Wilson, D.S.: Complex interactions in metacommunities, with implications for biodiversity and higher levels of selection. Ecology 73, 1984–2000 (1992)
8. Boyd, R., Richerson, P.J.: Group Selection among Alternative evolutionarily Stable Strategies. J. Theor. Biol. 145, 331–342 (1990)
9. McElreath, R., Boyd, R., Richerson, P.J.: Shared norms and the evolution of ethnic markers. Current Anthropology 44(1), 122–129 (2003)
10. Boyd, R., Richerson, P.J.: Culture and the Evolutionary Process. University of Chicago Press, Chicago (1985)
11. Wilson, D.S., Dugatkin, L.A.: Group selection and assortative interactions. Am. Nat. 149, 336–351 (1997)
12. Powers, S.T., Mills, R., Penn, A.S., Watson, R.A.: Social Environment Construction Provides an Adaptive Explanation for New Levels of Individuality. In: Proceedings of ECAL 2009 Workshop on Levels of Selection and Individuality in Evolution: Conceptual Issues and the Role of Artificial Life Models (2009)
13. Mills, R., Watson, R.A.: Symbiosis enables the evolution of rare complexes in structured environments. In: Proceedings of the 10th European Conference on Artificial Life (2009) (to appear)
14. Watson, R.A., Palmius, N., Mills, R., Powers, S.T., Penn, A.S.: Can selfish symbioses effect higher-level selection? In: Proceedings of the 10th European Conference on Artificial Life (2009) (to appear)

Evolution of Individual Group Size Preference Can Increase Group-Level Selection and Cooperation

Simon T. Powers and Richard A. Watson

Natural Systems group, University of Southampton, U.K.
stp05r@ecs.soton.ac.uk

Abstract. The question of how cooperative groups can evolve and be maintained is fundamental to understanding the evolution of social behaviour in general, and the major transitions in particular. Here, we show how selection on an individual trait for group size preference can increase variance in fitness at the group-level, thereby leading to an increase in cooperation through stronger group selection. We are thus able to show conditions under which a population can evolve from an initial state with low cooperation and only weak group selection, to one where group selection is a highly effective force.

1 Introduction

Understanding how cooperative social groups can evolve and be maintained is a major challenge in both artificial life [1,2] and evolutionary biology [3,4]. In particular, work on cooperative group formation has gained an impetus in recent years through a recognition of the major transitions in evolution [5], and the corresponding realisation that cooperation can be a driving force in evolution and not a mere curio [6]. The major transitions include the evolution of multi-cellular organisms from single cells, and the evolution of societies from solitary organisms. In these transitions, lower-level entities (single cells, solitary organisms) formed groups and donated part of their personal fitness to contribute to shared group success, to the point where cooperation amongst the lower-level entities was so great that the group ultimately became an individual in its own right [5,6] (e.g., somatic cells give up independent reproductive rights in multi-cellular organisms, likewise for sterile workers in eusocial insect colonies). Such transitions therefore represent premier examples of cooperative group formation. From the viewpoint of artificial life, understanding the mechanisms behind these transitions, and cooperative group formation more generally, may help us to understand the evolution of increased complexity [1,7], and to create applied systems where the evolution of a high degree of cooperation is supported [2].

The key difficulty that any explanation for the evolution of cooperative groups must overcome is this: if performing a group-beneficial cooperative act entails some cost, then why should an individual donate a component of its own fitness in order to contribute to the success of its group? Surely selfish cheats who do not themselves cooperate but instead simply reap the benefits of other group

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 53–60, 2011.

members' cooperation should be expected to be fitter, for they receive all of the group benefits whilst paying none of the individual costs. However cooperation can nevertheless evolve, even in the face of selfish cheats, if there is competition and hence selection between groups [8]. This is because groups with more cooperators will, all other things being equal, outcompete other groups that are dominated by selfish individuals. In this way, selection acting between groups provides an evolutionary force favouring cooperation. However, because selfish cheats still enjoy a relative fitness advantage within each group through free-riding, any between-group selection for cooperation is opposed by within-group selection for selfishness. In such a scenario selection is therefore multi-level, operating both within- and between- groups. The degree to which group cooperation evolves then depends on the extent to which selection *between* groups is stronger than selection *within* groups [8].

However, it is commonly held that the conditions under which between-group selection can overpower within-group selection are rather limited [9]. In particular, there must be a high variance in the proportion of cooperators between different groups, such that some groups contain many more cooperators, and are hence much more productive, than others. One way that this variance can be generated is if groups are formed assortatively, such that cooperators tend to form groups with other cooperators. This could happen, for example, if groups were founded by kin [10], for the genealogical relatives of a cooperator will themselves tend to be cooperators. Another way a high between-group variance can arise is if groups are founded by individuals sampled at random from the global population, but the number of founding individuals is small [11]. In this case the groups will start out as small, unrepresentative, samples of the population and so will tend to have a high variance. Selection at the group-level will be more effective as a result of this increased variance, and the increased effect of group selection would lead to greater cooperation. However, the initial group size must typically be very small if significant group-level variance and hence selection is to be generated by this mechanism.

We consider here how such groups with high variance, and that are consequently much affected by group selection, can evolve from an initial population where between-group variance is low and group selection weak. In order to do so we relax an assumption that is present in nearly all other multi-level selection models, namely, that initial group size must remain fixed over evolutionary time. Rather, we consider here the possibility that many organisms across all taxa may be able to influence the size of their group to some extent, through genetically coded (individual) traits. As an example, bacteria living in biofilms have a trait that controls the amount of extracellular polymeric substance that they secrete [12]. Since this substance allows bacterial cells to bind together, the amount produced could influence microcolony (group) size. In our model we then consider the evolution of both a group size preference trait such as this, and a behavioural trait that determines whether an individual cooperates or not, such that both traits evolve concurrently. If smaller groups lead, through increased group selection, to increased cooperation then individuals with a preference for founding smaller groups could potentially be selectively favoured, for they would experience a greater frequency of the benefits of cooperation in their groups and hence be on average fitter.

In our previous work [13] we showed that, in principle, individuals with a preference for a group size very much smaller than the current size could invade if they arose in a sufficiently high frequency. Such a sudden, large, decrease in group size is easier to evolve, since the increase in cooperation is greater if there is more of a decrease in group size. However, if group size decreased gradually, which seems a more plausible evolutionary mechanism, then it is not clear whether there could be sufficient immediate individual benefit to be selectively favoured. Here, we examine conditions under which groups of a smaller initial size can in fact evolve from larger ones, simply through gradual unbiased mutations on individual size preference, even when there are some other advantages to being in a larger group (e.g., better predator defence, access to resources that a smaller number of individuals cannot obtain [14]). We find that large jumps in size preference, as modelled by our previous work, are not required under (negative) frequency-dependant within-group selection. Consequently, this process can provide a gradualist explanation for the origin of cooperative groups.

2 Modelling the Concurrent Evolution of Initial Group Size and Cooperative Behaviour

The simulation model presented here considers a population structured as follows. Individuals reproduce within discrete groups for a number of generations, before individuals in all groups disperse and form a global migrant pool (dispersal stage). New groups are then formed by a uniform random sampling of individuals from this migrant pool, and the process of group growth, dispersal, and reformation repeats. Selection *within* groups occurs during the group growth stage, where individuals reproduce and have fitness-affecting interactions with other group members (i.e., selfish individuals will have more offspring than cooperators within that same group). On the other hand, selection *between* groups occurs at the dispersal stage, since those groups that have grown to a larger size (i.e., those with more cooperators) will contribute more individuals to the migrant pool. This kind of population structure is very similar to that considered in some classical group selection models [15,11,10], and especially fits organisms that live on temporary resource patches which become depleted, thereby periodically forcing dispersal and the formation of new groups.

Unlike previous models, we give individuals a genetically coded initial group size preference. An individual genotype then contains two loci: the first codes for cooperative or selfish behaviour, the second for an initial group size preference. Mutation on these loci happens after the dispersal stage; double mutations within a single organism are not allowed. Mutation at the size locus is done by adding or subtracting 1 to the current size preference (with 50% probability each); mutation at the behaviour locus is done by swapping to the other behaviour. Groups are formed randomly with respect to behaviour (cooperative or selfish), but assortatively on size, such that individuals with a preference for smaller groups tend to, on average, find themselves in such groups, likewise for individuals with a preference for larger groups. This is implemented by creating a list of individuals in the migrant pool sorted ascendingly by their size preference. Individuals from this list are then added in order to a group, until the size of the group exceeds the mean preference of the group members, at which point a new group is created for the next individual and is populated in the same manner. An algorithmic overview of the model is as follows:

1. **Initialisation:** Let the migrant pool be the initial population.
2. **Group formation:** Assign individuals in the migrant pool to groups.
3. **Reproduction:** Perform reproduction and selection within groups for a fixed number of generations, using the fitness functions defined below.[1]
4. **Dispersal:** Place all individuals into the migrant pool.
5. **Mutation:** Mutate individuals in the migrant pool.
6. **Iteration:** Repeat from step 2 onwards until and equilibrium is reached.

2.1 Within-Group Selection: Snowdrift and Prisoner's Dilemma Games

To model within-group selection and reproduction, we assume that group members have fitness-affecting interactions with each other as defined by the payoff matrix in Table 1, where b is the benefit of cooperating, and c the cost. If $b > c$ then this represents the Snowdrift game [16], where a coexistence of cooperative and selfish individuals is supported at equilibrium in a freely-mixed population, since both types enjoy a fitness advantage when rare. Such an equilibrium coexistence of behaviours can occur if cooperators are able to internalise some of the benefits of cooperating. For example, if cooperation involves the production of a public good, then cooperators may be able to keep a fraction of the good they produce for themselves [17]. Where this occurs cooperators will receive, on average, a greater *per capita* share of the benefits of cooperation, giving them a fitness advantage. However, if the benefit of cooperation becomes discounted with additional cooperators, but the cost remains the same, then selfish individuals will become fitter as cooperators increase above a threshold frequency [17]. The Snowdrift game is thus a model of negative frequency-dependent selection leading to a mixed equilibrium; there is a growing realisation that this type of dynamic occurs in many biological systems [16,17]. It should be stressed that despite an equilibrium coexistence of behaviours, it is still the case that mean fitness would be higher if all individuals cooperated. This then means that if the game is played in a group structured population, groups with more cooperators will still be fitter than those with less. Group selection can, therefore, still further increase the global frequency of cooperation.

For $c > b > c/2$, the payoff matrix in Table 1 yields the classical Prisoner's Dilemma [16], where selfish individuals are always fitter and cooperators are

[1] We rescale the groups after each generation to maintain a constant population size.

Table 1. Payoff matrix for within-group interactions

	Cooperate	**Selfish**
Payoff to Cooperate	$b - c/2$	$b - c$
Payoff to Selfish	b	0

driven extinct at equilibrium in a freely-mixed population. Again, however, mean fitness would be higher if all individuals cooperated. We generalise the structure of this 2-player payoff matrix to a group of n players by multiplying the payoff matrix by the proportion of behaviours within the group, as is standard when forming a replicator equation in evolutionary game theory; in our model this corresponds to treating each group as a separate freely-mixed population. Doing so yields the following fitness equations, where w_c is the fitness of cooperators, w_s the fitness of selfish individuals, p the proportion of cooperators within the group, w_0 a baseline fitness in the absence of social interactions, and σ_n a sigmoidal function that provides a benefit depending on group size n, as described below:

$$\begin{aligned}
w_c &= p\left(b - \frac{c}{2}\right) + (1-p)\,(b-c) + w_0 + \sigma_n \\
w_s &= pb + w_0 + \sigma_n \\
\sigma_n &= \frac{\beta}{1 + e^{-\mu n}} - \frac{\beta}{2}
\end{aligned}$$

σ_n is a sigmoidal function which takes in as input the current group size, and has as parameters a gradient μ (which determines how quickly the benefit tails off as the group gets larger), and a maximum fitness benefit β. This provides what is known in ecology as an Allee effect, whereby there is an advantage in number when groups are small, but as the group grows this is overwhelmed by the negative effects of crowding, i.e, the effect saturates with increasing size [14].

3 Results

We investigate here the conditions under which an individual preference for groups of a smaller initial size can evolve, and lead to greater group selection and cooperation. In all cases, we start all individuals out with the same size preference (20), and then consider the evolutionary dynamics that occur through mutation and selection. The initial frequency of the cooperation allele is taken to be the global equilibrium frequency in the model that occurs if group size was fixed at the starting size. We then record the changes in mean initial group size and global proportion of cooperators over time. A particular focus of this study is to contrast the effects of within-group selection modelled on the Snowdrift versus the Prisoner's Dilemma game. We set $b/c = 1.1$ to yield the Snowdrift game, and $b/c = 0.9$ to produce the Prisoner's Dilemma . The following parameter settings were used throughout: 3 generations within groups prior to dispersal, $w_0 = 1$, mutation rate 1%, 90% of mutations on the size locus, and population size 1000.

We initially considered a case where there is no intrinsic advantage to larger groups (i.e., set $\sigma_n = 0$). We found that given time for a sufficient number of mutations to accumulate, a population could evolve down to an initial group size of 1, and 100% cooperation, from a large range of initial conditions. The reason initial group size tended towards size 1 is that when there is no intrinsic benefit to larger groups, such an initial size maximises cooperation and hence absolute individual fitness. A representative example of the population dynamics is shown in Fig. 1. Interestingly, our results show that a selective gradient towards smaller groups does not always exist at the start; although it was present in the Snowdrift game, in the Prisoner's Dilemma the process relied on genetic drift until such a time as very small groups were created. This drift can be seen by individual size preferences spreading out in both directions at the start, whereas in the Snowdrift game the mass of the population moved in one direction, thus showing the presence of an individual selective gradient favouring smaller groups from the outset.

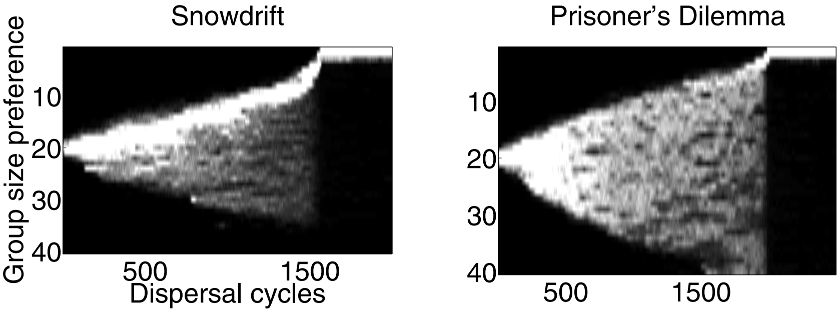


Fig. 1. Change in group size preference under Snowdrift and Prisoner's Dilemma games (lighter shades = greater frequency in population)

Next, we considered the case where there is an intrinsic advantage to larger groups (σ_n parameters: $\beta = 1$ and $\mu = 0.4$) . In such cases, the optimum initial group size (in terms of absolute individual fitness) is a trade-off between the benefit of cooperation and the intrinsic advantage to larger groups. It will thus be greater than 1, but not so large as to prevent any significant between-group selection and hence cooperation: using the parameters here, the optimum is 4. The question is then whether this optimum size can be reached by mutation. Figure 2 shows a representative case in which mean group size decreases to the optimum in the Snowdrift, but not Prisoner's Dilemma, game. The reason size preference cannot decrease in the Prisoner's Dilemma is that drift can no longer be effective if there is an intrinsic advantage to larger groups, for the advantage of larger groups provides a selective pressure away from small. By contrast, in the Snowdrift game a counter selective gradient towards smaller groups and increased cooperation exists from a much larger range of initial conditions. This counter gradient is provided by the benefits of increased cooperation that can be realised in smaller groups due to increased group selection. In the Prisoner's

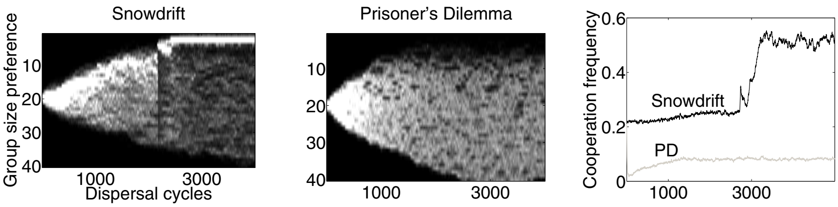


Fig. 2. Allowing for some benefit to larger groups. Left and centre: change in group size preference (lighter shades = greater frequency). Right: proportion of cooperators.

Dilemma there is no such gradient because group selection is completely ineffective over much of the parameter space, and so a small mutation on size would not lead to any increase in cooperation.

The existence of a selective gradient over a larger range of parameters follows from the fact that the Snowdrift game maintains a low frequency of cooperators at (group) equilibrium. Consequently, because one type cannot be driven extinct there is always the possibility that group-level variance can be generated when the groups are reformed, even when groups are large, as shown in some of our previous work [18]. As a result, moving to a slightly smaller group size can further increase this variance, and allow increased group selection and a subsequent increase in cooperation. By contrast, in the Prisoner's Dilemma game moving to a slightly smaller group size would not increase the effect of group selection over much of the parameter space, for cooperators are driven extinct over a large range of group sizes, thereby destroying the possibility of any group variance. Where this occurs, there cannot be selection for a smaller group size, and so the process must rely on drift, which cannot overcome an intrinsic advantage to larger groups. Our results therefore demonstrated that the evolution of smaller initial group size and greater cooperation is much more plausible if within-group selection takes the form of a Snowdrift game.

4 Concluding Remarks

Many of the major transitions involved the evolution of mechanisms that increased variance and hence selection at the group level, thereby allowing a high degree of cooperation between group members to evolve [6]. One such mechanism could be a reduction in initial group size. For example, multi-cellular organisms are themselves cooperative groups of individual cells. Most multi-cellular organisms develop from a single cell, and it has been argued that this evolved at least partly because it increased between-organism (cell group) variance, and hence increased cooperation between individual cells within the organism [5,6]. We have shown here how selection on an individual trait can lead to the evolution of increased variance in fitness at the group level, and hence a rise in cooperation between group members through increased group selection. Our results demonstrated that such a process is much more plausible if negative frequency-dependent selection, as modelled here by the Snowdrift game, operates within groups.

Acknowledgements. Thanks to Alex Penn and Seth Bullock for many discussions, and Rob Mills for feedback on the manuscript.

References

1. Stewart, J.: Evolutionary transitions and artificial life. Artificial Life 3(2), 101–120 (1997)
2. Nitschke, G.: Emergence of cooperation: State of the art. Artificial Life 11, 367–396 (2005)
3. Hammerstein, P.: Genetic and Cultural Evolution of Cooperation. MIT Press, Cambridge (2003)
4. West, S.A., Griffin, A.S., Gardner, A.: Evolutionary explanations for cooperation. Current Biology 17(16), R661–R672 (2007)
5. Maynard Smith, J., Szathmáry, E.: Major Transitions in Evolution. Spektrum (1995)
6. Michod, R.E.: Darwinian Dynamics: Evolutionary Transitions in Fitness and Individuality. Princeton University Press, Princeton (1999)
7. Bedau, M.A.: Artificial life: organization, adaptation, and complexity from the bottom up. Trends in Cognitive Science 7, 505–512 (2003)
8. Wilson, D.S., Sober, E.: Reintroducing group selection to the human behavioral sciences. Behavioral and Brain Sciences 17(4), 585–654 (1994)
9. Maynard Smith, J.: Group selection. Quarterly Review of Biology 51, 277–283 (1976)
10. Wilson, D.S.: Altruism in mendelian populations derived from sibling groups: The Haystack model revisited. Evolution 41(5), 1059–1070 (1987)
11. Wilson, D.S., Colwell, R.K.: Evolution of sex ratio in structured demes. Evolution 35(5), 882–897 (1981)
12. Flemming, H.C., Neu, T.R., Wozniak, D.J.: The EPS matrix: The "House of Biofilm Cells". Journal of Bacteriology 189(22), 7945–7947 (2007)
13. Powers, S.T., Penn, A.S., Watson, R.A.: Individual selection for cooperative group formation. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 585–594. Springer, Heidelberg (2007)
14. Avilés, L.: Cooperation and non-linear dynamics: An ecological perspective on the evolution of sociality. Evolutionary Ecology Research 1, 459–477 (1999)
15. Maynard Smith, J.: Group selection and kin selection. Nature 201, 1145–1147 (1964)
16. Doebeli, M., Hauert, C.: Models of cooperation based on the prisoner's dilemma and the snowdrift game. Ecology Letters 8(7), 748–766 (2005)
17. Gore, J., Youk, H., van Oudenaarden, A.: Snowdrift game dynamics and facultative cheating in yeast. Nature 459, 253–256 (2009)
18. Powers, S.T., Penn, A.S., Watson, R.A.: The efficacy of group selection is increased by coexistence dynamics within groups. In: Proceedings of ALife XI, pp. 498–505. MIT Press, Cambridge (2008)

Implications of the Social Brain Hypothesis for Evolving Human-Like Cognition in Digital Organisms

Suzanne Sadedin and Greg Paperin

Clayton School of Information Technology,
Monash University, Vic. 3800 Australia
Suzanne.Sadedin@infotech.monash.edu.au

Abstract. Data show that human-like cognitive traits do not evolve in animals through natural selection. Rather, human-like cognition evolves through runaway selection for social skills. Here, we discuss why social selection may be uniquely effective for promoting human-like cognition, and the conditions that facilitate it. These observations suggest future directions for artificial life research aimed at generating human-like cognition in digital organisms.

Keywords: artificial intelligence, sociality, evolution, social selection, Machiavellian intelligence.

1 Introduction

One of the most ambitious goals of Artificial Life research is the development of a digital organism that exhibits human-like cognitive behaviour [1, 2]. Such human-like behaviours include creative innovation, tool manufacture and use, self-awareness, abstract reasoning, and social learning [3-6]. Early researchers in artificial intelligence (AI) hoped that human-like cognition would rapidly emerge from AIs endowed with inference systems and explicit symbolic representations of their universe [2]. They were too optimistic. These inference systems, composed of search strategies and formal logic, provided good solutions in domains with semantics of limited complexity, such as games, mathematical problems and other formal systems, as well as simple model worlds [7] [8]. However, they failed to exhibit any useful intelligence in real-world settings [1].

Biomimicry is the development of engineered designs using biological principles. In the 1980s Brooks [9] and others argued that the true challenge of AI was to engineer systems that exhibit not human-like, but insect-like intelligence: a robust capability to perceive and respond in complex and dynamic environments. This view sparked many advances in areas such as machine learning, evolutionary computation and artificial life. Yet despite these advances, an engineered design for a human-like cognition remains elusive [1]. While many new systems perform well in specialised applications in complex environments, the integration of low-level behavioural intelligence and cognitive reasoning appears to hit a complexity wall [1].

We propose that evolutionary in-silico techniques may provide a way around the complexity wall by recreating conditions that promote human-like cognition in

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 61–68, 2011.

animals. Empirical and theoretical studies are approaching consensus on both the features associated with human-like cognition in animals, and the evolutionary conditions required [3-6]. These studies imply that our cognitive abilities are not a collection of independent traits, but a predictable product of selection in specific social environments. This view is termed the social brain or Machiavellian intelligence hypothesis [10, 11]. We discuss how social brain evolution may be embodied in a model system to develop human-like cognition in a digital context.

2 Patterns in the Evolution of Human-Like Cognition

Numerous studies show that large brains are strongly associated both with human-like cognition and with specific social systems and behaviours, but only weakly related to ecological factors. Thus, human-like cognition evolves as a social, not ecological, adaptation. Today, this view is widely accepted for primate cognition; recent reviews also support extending it to other taxa [3, 6, 12-15].

Brains and ecology. Relative brain size (brain-body mass ratio) differs dramatically among species. Among mammals, exceptionally large brains occur in primates, cetaceans (dolphins and whales) [6], and carnivores [12, 16]. Most bird brains are larger than those of comparable mammals, with the largest non-primate brains occurring in parrots and corvids (crows, jays and magpies) [13, 17, 18]. Ecological selection does not account for these patterns [17, 19-21]: why should birds need bigger brains than mammals? Why should seed-eating parrots need extremely large brains? Physical capacity for complex, flexible behaviour also seems unrelated to brain size: brain size and forelimb dexterity are uncorrelated in terrestrial carnivores [16], and handless dolphins are second only to apes in relative brain size [6].

Brains and social system. The idea that sociality might drive brain evolution was first proposed in the mid-twentieth century [19, 22, 23], but only recently attained wide acceptance [4]. In contrast to the diverse ecologies of large-brained animals, their social behaviour is strikingly similar. Large-brained vertebrates interact with unrelated adults, forming lasting and powerful social relationships with some [13, 14, 24]. Most primates, cetaceans and carnivores share fission-fusion group dynamics, where coalitions of varying size and stability interact, cooperate and compete [21]. Large-brained birds more often form relatively stable socially or genetically monogamous pairs [24]. In all of these groups, long-term cooperative relationships among unrelated adults powerfully influence reproductive success[24].

Dunbar [11] noted that group size and relative brain size are strongly related in primates, providing the first evidence that selection for social skills may drive the evolution of primate cognition. However, studies of other taxa suggested the result might be specific to primates: group size in parrots and hoofed mammals does not correlate with brain size[3, 25]. Closer attention reveals that the relation holds for some non-primates: brain size correlates with social group size in carnivorous and insectivorous mammals [26] and cetaceans [6]. However, Emery et al.[24] showed that brain size in birds was related to relationship duration, with social monogamists having smaller brains than genetic monogamists. Similarly, hoofed mammals with unusually large brains do not aggregate in large, anonymous herds, but form small, close-knit social groups with persistent relationships [14].

Brains and human-like cognition. Many aspects of cognition once thought to be confined to humans have now been documented in a range of species. For example, elephants [27], dolphins [28], corvids [29] and apes can recognize themselves in mirrors, a classical test of self-awareness (although one that favours visually-oriented animals). Social transmission of learned behaviour is also seen in these animals, allowing simple forms of cultural evolution [30]. Creative tool-making and tool-use have also been observed in many of these species [15]. Thus, while none of these species exhibits human-like technology or complex language, their traits suggest evolutionary antecedents to human cognition in diverse groups.

Surveys of brain structure and size show that the behaviours described above – self-awareness, flexible problem-solving, tool manufacture and use, innovation and social learning – are consistently associated both with unusually large brains, and with specific social behaviours [3, 6, 20]. Large-brained animals show extended parental care, mature slowly and are playful even in adulthood [25, 31, 32]. They use sound both creatively and playfully, and their songs are wholly or partially learned [33]. Positive relationships between relative brain size, tool use, innovation and social learning are documented across a broad range of species [5, 6, 13, 20, 34]. In addition, early birth/hatching, slow maturation, long lifespan and extended parental care correlate with these behaviours and with relative brain size [32].

Reader and Laland [20] noted that innovation rate and social learning are correlated independent of brain size. This result implies that there is no trade-off between social and asocial learning; rather, social cognition provides general cognitive benefits. The same study showed that that social group size does not correlate with social learning when other factors are controlled. This observation supports the idea that cognitive evolution is not determined by the volume of social interactions per se, but rather reflects the importance of personal relationships.

3 Social Selection on Social Brains

The dominant explanation for the above observations is a theoretical framework termed the social brain hypothesis [4, 10, 22, 23, 35]. A key factor in this framework is social selection, which occurs when the fitness of individuals depends on interactions with others. Social selection can drive the evolution of dramatic displays due to positive feedback which progressively exaggerates signal traits that reliably indicate social benefits [36-38]. For most species, the traits relevant to social success are primarily physical, so displays reflect individual physical ability. Complex strategies may be invoked by social displays without complex cognition, so social selection alone need not lead to brain evolution. However, when power is distributed in a network of cooperative individual relationships, the ability to manipulate this social network may become the target of social selection, resulting in social selection for improved cognition (Figure 1). Formulations of the social brain hypothesis vary in their emphasis on social complexity, display, innovation, culture, cooperation and cheating. All, however, incorporate these factors in a positive feedback loop that drives increases in cognitive ability until resource limits are met.

The social network is a unique selective environment in that fitness depends on the ability to model the minds and relationships of others; as minds and relationships become more complex, so do these models [22]. Through social selection for such cognitive ability, extreme displays of social skill may evolve, such as humour, artistry and song [36]. Increased behavioural flexibility resulting from improved cognition may then reduce the influence of ecological selection further [34, 39].

Cooperation emerges and persists in restrictive evolutionary conditions [40]; in particular, models suggest that it is likely to evolve in viscous social networks where repeated interactions allow punishment of non-cooperators [41]. For animals with limited cognition, group sizes that are too large to recall individual interactions undermine cooperation. Competition, however, is more intense when there are more competitors. Consequently, it seems plausible that social selection and selection for cooperative relationships combine most forcefully at group sizes and interaction rates that are as large as possible for cooperation to evolve given existing cognitive limits. This observation may explain why species that live in medium-sized groups often have the largest brains.

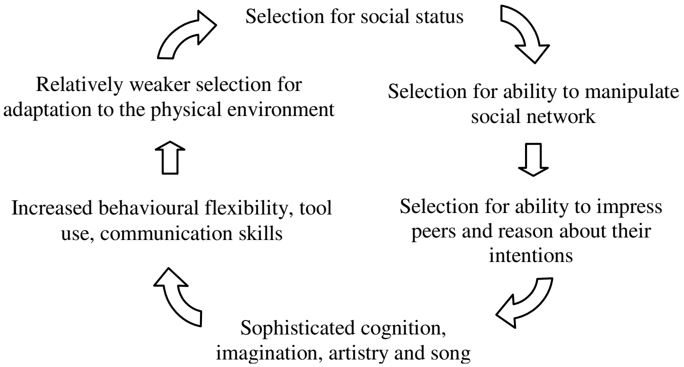

Selection for social status
Selection for ability to manipulate social network
Selection for ability to impress peers and reason about their intentions
Sophisticated cognition, imagination, artistry and song
Increased behavioural flexibility, tool use, communication skills
Relatively weaker selection for adaptation to the physical environment

Fig. 1. Conceptual model for the evolution of human-like cognition through social selection. When fitness is determined by social interactions, selection for social skills drives the evolution of displays of cognitive ability in a positive feedback loop which is ultimately inhibited by external resource limitations.

Long-term relationships also appear crucial to the evolution of human-like cognition [6, 14, 24]. Social choices made without relationship commitment have immediate consequences but can be easily modified. In contrast, relationships that require large investments impose strong selection for appropriate partner choice; this may be why birds that are genetically rather than socially monogamous have larger brains [24]. A ubiquitous feature of long-term relationships is courtship. Courtship involves playful and coordinated display activities such as dancing, duets, and mutual grooming; it seems likely that these activities allow animals to assess the value of social investment [31].

4 Evolving Artificial Human-Like Cognition

The results described in Sections 2 and 3 imply that features of human-like cognition are likely to evolve only when populations are faced with:

- A complex network of social interactions
- Selection for social status
- Ability to communicate to other member of the social network
- Ability to observe other members of the social network and their actions

Possible ways to instantiate these features in artificial life are suggested below.

Individuals. In most animals, cognition is performed by natural neural networks. Thus, one option is to represent individuals using an artificial neural network. However, as discussed in section 3, in most animals these networks evolved, not for abstract reasoning, but to channel responses to specific stimuli. Clearly, brains can be used for human-like cognition, but some functions may be prerequisite. This may include ability to represent objects, associative memory, communication, basic empathy, and others. Natural evolution of such functions took a very long time, and may be extremely complex. Including pre-wired components that can perform such basic functions may facilitate evolution of more interesting properties.

Artificial neural networks are not the only possible choice to perform cognition. It has been shown that artificial life (although not intelligent) can arise in a universe of co-evolving computer programs (e.g. [42, 43]). Logical and probabilistic systems can also exhibit limited representational thinking (see review by [44]). Combining such techniques with evolutionary systems may yield an appropriate representation.

The genotype representation in the model must support phenotypes of varying size and complexity. Lenton and Oijen [45] define a hierarchy for intrinsic control dynamics in adaptive systems. In their terms, the representational system for individuals must exhibit at least the third order of control. Second order systems with fixed representational structure may show learning capabilities in specific domains, but are not expected to develop open-ended cognition.

Fitness. Reproductive success (fitness) must be based on the social standing of an individual relative to its peers. One way to do this is to supply every newborn individual with a unit amount of an abstract resource *R*. This represents the amount of social status it can confer on other individuals. Throughout its lifetime, an individual contributes to the social status of its peers by supplying them with an arbitrary proportion of *R*. An individual must share the entire unit amount of *R* during its life time, but cannot share *R* that it received from other individuals (it cannot pass on respect received from others). The social status (relative fitness) of an individual is determined by the total *R* it receives throughout its lifetime.

Environment. We have argued that human-like cognition evolves only when selection targets social network skills. In social selection, individuals must perceive the actions of others (such as donating *R* or communicating a statement). In the real world, perception depends on individual location, but allowing perception to be determined by social network proximity seems sufficient for the current model.

Communication. Without a physical environment there is no explicit communication channel in the model universe. A statement can be encoded as a tuple (S, D, M), where S and D are source and destination individuals respectively, and M is a message. The role of noise and the circumstances under which a message between S and D may be observed by others require further consideration. The encoding of M may be critical for the evolution of meaningful communication and of cognitive abilities: a particular choice must agree with the representation chosen for individuals. Most generally, M may be represented as a bit string with a fixed maximum length. Individuals communicate randomly at the beginning of the model evolution, but once some individuals begin to be influenced in their actions by perceived communication, others will gain a selective advantage from discovering this and from sending specific messages. This may lead to evolution of primitive communication - a prerequisite for complex social interactions.

Model lifecycle. The system can be initialised with a set of populations, each consisting of 10 to 30 individuals created randomly. The lifecycle is based on non-overlapping generations, each consisting of two phases – competition and reproduction. During the competition phase, each individual receives a unit amount of resource R. The individuals then communicate by passing messages as described earlier. At any time during this phase, an individual may choose to pass on some of its R to any other individual. This process continues until all individuals have passed on all of their initial R. The reproduction phase then ensues, with individual reproductive success proportional to the amount of R received during the competition phase. Mutation supplies variation. Existing adults are replaced by juveniles and the competition phase ensues for the new generation.

5 Conclusions

We have argued that human-like cognitive traits evolve naturally in a restrictive set of evolutionary conditions when the social network of an organism is its primary selective environment. In particular, organisms that can increase their fitness by adaptively and flexibly forming long-term, high-commitment relationships seem most likely to evolve large brains, creative innovation, tool use, social learning and abstract reasoning. These principles could be embodied in an artificial life system to develop human-like cognition in a digital context.

References

[1] Bedau, M., McCaskill, J., Packard, N., Rasmussen, S., Adami, C., Green, D., Ikegami, T., Kaneko, K., Ray, T.: Open problems in artificial life. Artificial Life 6, 363–376 (2000)

[2] Dreyfus, H.L.: Why Heideggerian AI failed and how fixing it would require making it more Heideggerian. Artificial Intelligence 171, 1137–1160 (2007)

[3] Emery, N., Clayton, N.: The mentality of crows: convergent evolution of intelligence in corvids and apes. Science 306, 1903–1907 (2004)

[4] Flinn, M., Geary, D., Ward, C.: Ecological dominance, social competition, and coalitionary arms races Why humans evolved extraordinary intelligence. Evolution and Human Behavior 26, 10–46 (2005)

[5] Lefebvre, L., Reader, S., Sol, D.: Brains, innovations and evolution in birds and primates. Brain, Behav. Evol. 63, 233–246 (2004)

[6] Marino, L.: Convergence of complex cognitive abilities in cetaceans and primates. Brain Behav. Evol. 59, 21–32 (2002)

[7] Giunchiglia, F., Villafiorita, A., Walsh, T.: A Theory of Abstraction, Universita di Genova facolta di ingegneria, dipartimento informatica sistemistica telematica, Genova, Technical Report MRG/DIST #97-0051 (November 1997)

[8] Nilsson, N.J.: Shakey the robot, SRI International Menlo Park, Menlo Park, CA, Technical note no. 323 A819854 (April 1984)

[9] Brooks, R.A.: Achieving artificial intelligence through building robots, Massachusetts Institute of Technology, Boston, MA, Technical report A.I. Memo 899 (May 1986)

[10] Byrne, R.: Machiavellian intelligence: Social expertise and the evolution of intellect in monkeys, apes, and humans. Oxford University Press, Oxford (1989)

[11] Dunbar, R.: Grooming, gossip, and the evolution of language. Harvard Univ. Pr., Cambridge (1998)

[12] Holekamp, K., Sakai, S., Lundrigan, B.: Social intelligence in the spotted hyena (Crocuta crocuta). Philosophical Transactions of the Royal Society B: Biological Sciences 362, 523 (2007)

[13] Marler, P.: Social cognition: Are primates smarter than birds. Current Ornithology 13, 1–32 (1996)

[14] Schultz, S., Dunbar, R.: Both social and ecological factors predict brain size in ungulates, pp. 207–215 (2006)

[15] Seed, A., Clayton, N., Emery, N.: Cooperative problem solving in rooks (Corvus frugilegus). Proceedings of the Royal Society B: Biological Sciences 275, 1421 (2008)

[16] Iwaniuk, A., Pellis, S., Whishaw, I.: Brain size is not correlated with forelimb dexterity in fissiped carnivores (Carnivora): a comparative test of the principle of proper mass. Brain Behav. Evol. 54, 167–180 (1999)

[17] Bennett, P., Harvey, P.: Relative brain size and ecology in birds. Journal of zoology. Series A 207, 151–169 (1985)

[18] Iwaniuk, A., Dean, K., Nelson, J.: Interspecific allometry of the brain and brain regions in parrots (Psittaciformes): comparisons with other birds and primates. Brain, Behav. Evol. 65, 40–59 (2005)

[19] Chance, M., Mead, A.: Social behaviour and primate evolution. In: Machiavellian Intelligence: Social Expertise and the Evolution of Intellect in Monkeys, Apes, and Humans (1988)

[20] Reader, S., Laland, K.: Social intelligence, innovation, and enhanced brain size in primates. Proceedings of the National Academy of Sciences 99, 4436 (2002)

[21] Perez-Barberia, F., Shultz, S., Dunbar, R., Janis, C.: Evidence for coevolution of sociality and relative brain size in three orders of mammals. Evolution 61, 2811–2821 (2007)

[22] Humphrey, N.: The social function of intellect. In: Machiavellian Intelligence: Social Expertise and the Evolution of Intellect in Monkeys, Apes, and Humans, pp. 13–26 (1988)

[23] Jolly, A.: Lemur social behavior and primate intelligence. Science 153, 501–506 (1966)

[24] Emery, N., Seed, A., von Bayern, A., Clayton, N.: Cognitive adaptations of social bonding in birds. Philosophical Transactions of the Royal Society B: Biological Sciences 362, 489 (2007)

[25] Pearce, J.: Animal learning and cognition: An introduction. Psychology Pr., San Diego (1997)

[26] Dunbar, R., Bever, J.: Neocortex size predicts group size in carnivores and some insectivores. Ethology 104, 695–708 (1998)

[27] Plotnik, J., de Waal, F., Reiss, D.: Self-recognition in an Asian elephant. Proceedings of the National Academy of Sciences 103, 17053 (2006)

[28] Reiss, D., Marino, L.: Mirror self-recognition in the bottlenose dolphin: A case of cognitive convergence. Proceedings of the National Academy of Sciences 98, 5937 (2001)

[29] Prior, H., Schwarz, A., Güntürkün, O.: Mirror-induced behavior in the magpie (Pica pica): Evidence of self-recognition. PLoS Biol. 6, e202 (2008)

[30] Laland, K., Hoppitt, W.: Do animals have culture? Evolutionary Anthropology: Issues, News, and Reviews 12 (2003)

[31] Chick, G.: What is play for? Sexual selection and the evolution of play. Theory in Context and Out, 1 (2001)

[32] Iwaniuk, A., Nelson, J.: Developmental differences are correlated with relative brain size in birds: a comparative analysis. Canadian Journal of Zoology 81, 1913–1928 (2003)

[33] Fitch, W.: The biology and evolution of music: A comparative perspective. Cognition 100, 173–215 (2006)

[34] Sol, D., Duncan, R., Blackburn, T., Cassey, P., Lefebvre, L.: Big brains, enhanced cognition, and response of birds to novel environments. Proceedings of the National Academy of Sciences 102, 5460–5465 (2005)

[35] Dunbar, R.: The social brain hypothesis. Brain 9, 10

[36] Nesse, R.: Runaway Social Selection for Displays of Partner Value and Altruism. Biological Theory 2, 143–155 (2007)

[37] Tanaka, Y.: Social selection and the evolution of animal signals. Evolution, 512–523 (1996)

[38] Wolf, J., Brodie III, E., Moore, A.: Interacting phenotypes and the evolutionary process. II. Selection resulting from social interactions. The American Naturalist 153, 254–266 (1999)

[39] Sol, D., Timmermans, S., Lefebvre, L.: Behavioural flexibility and invasion success in birds. Animal Behaviour 63, 495–502 (2002)

[40] Axelrod, R., Hamilton, W.: The evolution of cooperation. Science 211, 1390–1396 (1981)

[41] Santos, F., Pacheco, J., Lenaerts, T.: Cooperation prevails when individuals adjust their social ties. PLoS Comput. Biol. 2, e140 (2006)

[42] Adami, C., Brown, C.T.: Evolutionary learning in the 2D artificial life system "Avida". In: Fourth International Conference on the Synthesis and Simulation of Living Systems (ALife IV), pp. 377–381. MIT Press, Cambridge (1994)

[43] Ray, T.S.: An approach to the synthesis of life. In: Second International Conference on the Synthesis and Simulation of Living Systems (ALife II), vol. 10, pp. 371–408 (1991)

[44] Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River (2003)

[45] Lenton, T.M., Van Oijen, M.: Gaia as a Complex Adaptive System. Philosophical Transactions of the Royal Society: Biological Sciences 357, 683–695 (2002)

Embodiment of Honeybee's Thermotaxis in a Mobile Robot Swarm

Daniela Kengyel[1], Thomas Schmickl[2], Heiko Hamann[2],
Ronald Thenius[2], and Karl Crailsheim[2]

[1] University of Applied Sciences St. Poelten, Computersimulation, Austria
[2] Artificial Life Laboratory of the Department of Zoology, Karl-Franzens University
Graz, Universitätsplatz 2, A-8010 Graz, Austria
daniela.kengyel@gmx.at,
{heiko.hamann,thomas.schmickl}@uni-graz.at,
{ronald.thenius,karl.crailsheim}@uni-graz.at

Abstract. Searching an area of interest based on environmental cues is a challenging benchmark task for an autonomous robot. It gets even harder to achieve if the goal is to aggregate a whole swarm of robots at such a target site after exhaustive exploration of the whole environment. When searching gas leakages or heat sources, swarm robotic approaches have been evaluated in recent years, which were, in part, inspired by biologically motivated control algorithms. Here we present a bio-inspired control program for swarm robots, which collectively explore the environment for a heat source to aggregate. Behaviours of young honeybees were embodied on a robot by adding thermosensors in 'virtual antennae'. This enables the robot to perform thermotaxis, which was evaluated in a comparative study of an egoistic versus a collective swarm approach.

1 Introduction

1.1 Motivation and Problem Formulation

Eusocial insects are fascinating and inspiring group consisting of termites, bees, and ants, which live in well organized colonies. Colonies of honeybees are precisely regulated. For example the temperature in the brood nest is kept constant at approx. 36°C. This is very important for the brood because otherwise this could lead to disorders in the development or even the loss of brood [1]. One challenge of a colony of honeybees in such a hive is, that the bees have to organize and orient themselves in a temperature gradient without the help of light. Such a temperature gradient is inhomogeneous and dynamic. This is a complicated challenge, because the antennae of honeybees are close to each other, but it is solved collectively by the bees [2]. To find a point of a special property, such as locating a field of heat or a leakage in a gas pipe, is a common challenge in robotics, especially if the environment is changing over time and no global gradient points to the target spot. One approach to tackle such challenges is using a bio-inspired approach, e.g. using a swarm to perform a parallelized and

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 69–76, 2011.

coordinated search. Following this idea, swarm robotics [3] emerged from combining autonomous robots with techniques originating from the field of swarm intelligence [4]. In swarm intelligent systems, the collective of agents performs decision making without a central unit that decides. As is summarized by [5], swarm intelligent solutions are robust, flexible, scalable and (computationally) cheap to implement on the individual. Camazine et. al. [6] showed that these properties apply also to biological swarms and social insect colonies, which exploit self-organizing processes to achieve swarm intelligence.

As an example, natural selection sharpened individual behaviour of bees in a way that colony fitness is maximized. Self-organisation and swarm intelligence can enhance this colony fitness significantly: In the following, we focus on groups of young honeybees which are an example of such a swarm-intelligent system. Young honeybees need a certain environmental temperature. The temperature in the hive varies usually between 32°C and 36°C while the optimal temperature for young honeybees is approximately 35°C [7]. This temperature is found at the brood nest of the hive, which is, preferred by the young honeybees [8]. Former experiments also showed that a single bee is mostly not able to detect the area of optimal temperature when the temperature gradient is rather flat [9].

1.2 Derivation of the Bio-inspired Solution

A model was reported in [10], that reproduces this behaviour of young bees without assuming any explicit communication between the bees. Inspired by the observed behaviour of young honeybees and based on this model, an algorithm for a swarm of robots was developed, which enables the robots to find an area with a special property collectively [2,10]. This typical (swarm) robotic task was frequently realized based on areas marked by light and using light sensors because this experiment configuration is supposed to be the technically simplest. We adopted the algorithm reported in [2] to consider the specific aspects of a temperature field in contrast to a light gradient field. The algorithm is based on robot-to-robot approaches. For simplicity we call it the BEECLUST algorithm further on. By using heat instead of light the robots' environment gets closer to the original bio-inspired situation young honeybees are faced with. Due to the fact that heat differs in its physical characteristics compared to light (e.g., warming up period, thermal diffusion, turbulences in the airflow) we show here how physical properties of stimuli are reflected in the swarm's control algorithm.

The typical engineering approach is very different to this swarm-intelligent approach: In his book Braitenberg describes algorithms, such as a simple gradient ascent, of robots which are sensitive to the environment [11]. There are several works describing complex approaches with cooperating sensor networks and robot groups (e.g., [12]). The sensor network measures the temperature gradient and communicates within the robot swarm. The robots are able to localize themselves globally and perform a simple gradient ascent. A single robot that produces heat trails and follows them is reported in [13].

2 Material and Method

2.1 Hardware and Electronics

For our robot experiments, we needed a small robot with two temperature sensors. A good solution was to upgrade an out-of-the-shelf robot with an according sensor system. Additional to an extensible I/O-board, the Hemisson robot has six infrared-sensors, which we used for collision avoidance and for the identification of other robots. Three of these sensors point to the front, one to each side and one to the back. The locomotion is realized by two wheels with a differential drive system. In addition, we developed a circuit processing the measured temperature. The circuit with temperature sensors is able to detect temperatures in a range of 140°C. We re-mapped this range to the relevant temperature interval [10°C, 60°C] of honeybees, represented by digital sensor values on the interval $e \in [50, 225]$. Our measurements indicate that the sensor's voltage is approx. linear within this temperature interval. Using an instrumentation amplifier the measured signal was amplified for processing the data in an easier way. The advantage of an instrumentation amplifier compared to the operational amplifier is that the former is regulated by a single resistor without affecting the circuit. Fig. 1A shows the Hemisson with the temperature sensors.

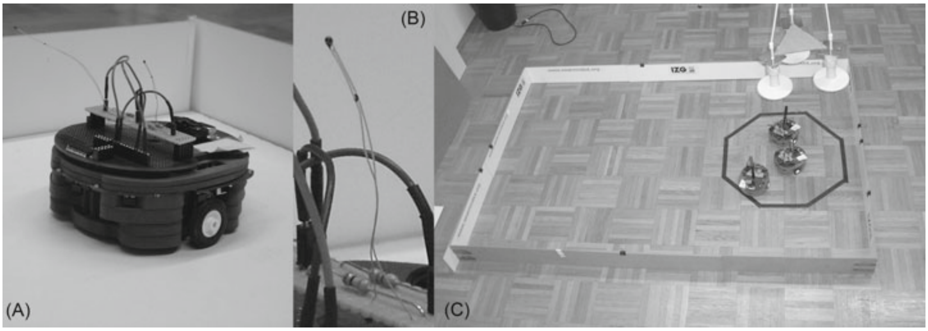


Fig. 1. A: Robot "Hemisson" with temperature sensors. B: Temperature sensor of the Hemisson. C: Robots underneath the heat lamps in the arena.

2.2 Algorithms

Naturally, there are many possibilities to implement the aggregation of autonomous agents at a certain target area. Here, we want to analyze the properties of the BEECLUST algorithm when working with fields of temperature instead of light gradients. This bio-inspired algorithm is different from anything that could be considered to be a standard approach. Actually, it might even be perceived as counter-intuitive. However, we hypothesize that it is robust and efficient in multi-robot scenarios. In order to test this hypothesis we compare the BEECLUST to a rather classic approach of a simple gradient ascent algorithm (GAA).

BEECLUST Algorithm: BEECLUST is adopted from young honeybees' navigation behaviour. In principle there are two strategies how young honeybees find the optimal temperature: In a rather steep gradient field, the individual behaviour of each single bee is effective and dominates. In a rather flat gradient, a single bee is mostly not able to find the spot of optimal temperature. The following group behaviour of young honeybees in such an environment was observed [2]: The bees walk around randomly until they meet another bee. Then they stop and wait. After a certain time the bees walk around randomly again. This process forms clusters of bees. The higher the temperature is, the longer the bees wait. Hence, the cluster sizes and numbers increase over time in the warmer area whereas they decrease in the colder area. To realize such a behaviour the robots have to perform several different tasks: random movement, discrimination between the wall and other robots, avoidance of collisions, measurement of local temperature and calculation of the waiting time.

Our program on the Hemisson robot mimics the minimal individual behaviour extracted from honeybees: A robot drives forward until it detects an object in the front. If this object is an obstacle (e.g., a wall) the robot turns in a random direction by a random angle. This turning angle is uniformly randomly distributed between 40 degrees and 140 degrees in both directions. If the detected object is another robot, the robot stops, measures the temperature, and calculates the time to wait. The robot-to-robot detection is realized by the help of passive sensing using infrared sensors. Passive sensing means, in contrast to active sensing, that no infrared light is emitted. If the sensor receives IR without emitting IR there is only one possibility: The IR-light was emitted by another robot. The dependency of the waiting-time on the measured temperature is also an important factor and is described with a sigmoid curve. The following equation is used to map the sensor values e to waiting times (for details see [2]):

$$w(e) = \frac{w_{\max} e^2}{e^2 + \theta}, \tag{1}$$

where θ indicates the steepness of the resulting curve. See table 1 for characteristics of the robot and other parameters that were used in this work in comparison to parameters of the original BEECLUST [2].

Table 1. Parameters

	temperature	light
speed of the robot	3 cm/s	30 cm/s
range of infrared sensors	4 cm	6 cm
robot size	12 × 13 cm	3 × 3 cm
max. waiting time $w_{\max}$	180 s	66 s
interval of the sensor value e	$[50, 225]$	$[0, 255]$
waiting time function offset θ	21,000	7,000

GAA – Gradient-Ascent Algorithm: The GAA represents a simple straightforward approach to thermotaxis. The collision avoidance is implemented in the same manner as described above. The only difference is that there is no robot-to-robot detection implemented and therefore no waiting-time is calculated. The GAA works as follows: The robot measures the temperature with a frequency of two measurements per second. If one sensor reports a difference of temperature greater than a threshold of 2 in the digital sensor value e (to achieve a certain robustness against fluctuations), the robot changes its direction and starts to turn to the warmer area until the difference between the two sensors has vanished. A difference of 2 units in the digital sensor value corresponds to a difference in the temperature of about 1°C.

2.3 Setup of the Experiment

Experiment 1: To compare both algorithms we used three Hemisson robots and built a rectangular arena as shown in Fig. 1C. The arena had a size of 150cm × 120cm. Over the right side of the arena we placed three infrared heat lamps at a height of 44cm. Each of them had a power of 250 watt. The lamps generated 30°C in a height of 10cm (where the robots measure the temperature) above the target area. The ambient air temperature in the arena (where no heat lamps were) was around 27°C. The target zone underneath the heat lamps was marked with a surrounding black line as seen in Fig. 1C. It covered about an eighth of the arena. Initially, the robots (depending on the scenario: one, two, or three) are positioned at the side of the arena opposite to the target zone. In case of BEECLUST, many robot-to-robot detections occur in the first seconds. However, they do not remain there because the waiting-time in the cold part of the arena is very small (mostly no waiting-time).

It was important to ensure that the ambient air temperature did not significantly change during the experiment. Such changes were prevented by heating up the room to 27°C before our experiments started. To ensure the same initial conditions the temperature was measured before each experiment started. In these experiments, the time each robot spent inside of the target zone was measured. Each experiment lasted 14 minutes.

Experiment 2: In an additional experiment the probability of staying within the target area after a robot-to-robot approach was measured after a time of 25 seconds. In case of the BEECLUST, not every robot-to-robot detection results in a successful detection of the other robot due to noise and blind spots of the infrared sensors. In case of the GAA, two approaching robots interfere with each other and might "push" each other out of the target area when trying to avoid collisions. In this experiment we had five different setups: *a*) one GAA robot, *b*) two GAA robots, *c*) three GAA robots, *d*) two BEECLUST robots and *e*) three BEECLUST robots . The robots started vis-á-vis and where placed 10cm outside of the target area. The third robot in experiment c and e was placed inside of the target area and was programmed to stay immobile. Therefore, it was not included in the determination of the results.

3 Results and Discussion

Fig. 2 shows the results of our three test runs in experiment 1 ($N = 6$ repetitions). Three robots with the GAA have spent 25.4% (mean) of time inside of the target zone. This time span is quite short compared to the mean of 49.7% in the runs with a single GAA robot. This is explained by the consequences of the competition for space underneath the heat lamps: If one robot is inside the target zone and a second robot tries to get into the target zone as well, it may happen that they approach each other. The collision avoidance behaviour might lead to a turn and the robot might leave the target area.

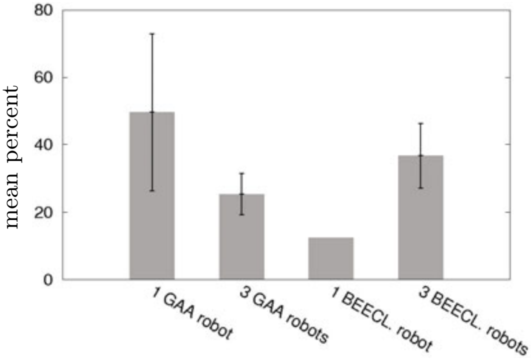


Fig. 2. Mean of the percentage of the total time a robot spent inside of the target zone (see Fig. 1C). Error bars show the standard deviation, $N = 6$ per setting.

One single BEECLUST robot spent an eighth of the time in the target area, because the target area covers an eighth of the arena. This was calculated statistically because one single robot cannot stop underneath the heat lamps. Three BEECL. robots yielded a result with a mean of 36.7%. In comparison to the GAA the swarm-intelligent BEECL. has a higher success rate. If there is no heat source in the arena the three robots do not aggregate for both algorithms.

In test runs with three robots, three cases of robot-to-robot encounters in the target area can occur, which are tested in experiment 2. Fig. 3 shows the relative frequency of a robot to stay in the target area. The accounted final state was measured 25 seconds after the start leaving enough time for a robot to get out of the target zone: One single GAA robot stayed inside of the target zone with a frequency of 0.75 per trial. Two GAA robots remained there with a frequency of 0.67 per trial whereas the frequency was 0.54 per trial in the experiment with three GAA robots. 2 BEECL. robots had a frequency of 0.91 per trial and 3 BEECL. robots remained in the target area with a frequency of 0.83 per trial. Significant differences were determined for the comparisons: 1 BEECL. and all other comparable settings (Chi-square test, $P < 0.0001$), 2 GAA and 2 BEECL. ($P < 0.0330$, $\chi^2 = 4.547$), and 3 GAA and 3 BEECL. ($P < 0.0293$, $\chi^2 = 4.752$).

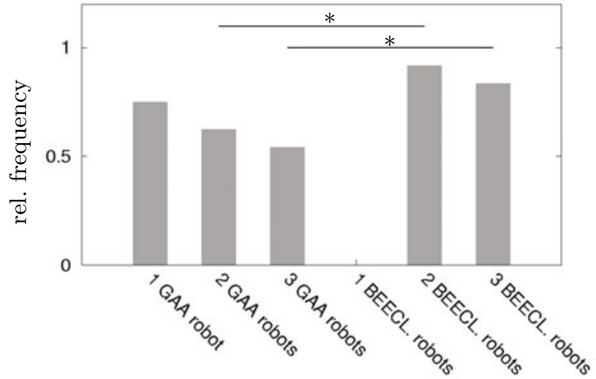


Fig. 3. Comparison of the relative frequency of robots that stay in the target zone for both algorithms and for all possible robot-to-robot encounters, $N = 12$ per setting

The probability of the GAA robots remaining in the target zone decreases with increasing number of robots. It is expected that with a higher number of robots the probability will decrease further because the limited resource "space" in the target area will get more scarce. Although this is true for the BEECLUST robots as well, they have a higher probability of staying in the target area.

4 Conclusion and Outlook

As expected the use of heat instead of light brings along significant challenges in designing an efficient robotic swarm due to different physical characteristics. Differences are observed, e.g., in the shape of the temperature gradient, in the time scale of diffusion and the time delays in the measurement of temperatures. This had to be reflected in the control algorithm by slower motion/turns and longer waiting periods. The time delay of measurement originates from the temperature sensors because they have to heat themselves up to decrease their resistance. If the robot is not exposed to heat long enough it measures a lower temperature and so the calculated waiting-time is shorter. As a consequence the aggregated robot swarm is not as robust as it could be in a light gradient.

A major challenge was the inhomogeneity concerning the spreading of heat and air flows. Due to the fact that hot air ascends, the heat lamps generate a lot of waste heat which heats up the ambient air instead of the desired area in the robot arena. Between experiment runs the initial conditions (defined temperatures at the target area, in the ambient air, etc.) needed to be re-established. Otherwise the air temperature saturates close to the temperature in the target zone and the robots cannot identify the target area anymore. This would influence the effectivity of the algorithms critically. Despite these characteristics, which increase the difficulty of this task, the robots were able to locate the target area beneath the heat lamps using the BEECLUST. This was possible although the configuration of the infrared-sensors of the Hemisson, used for collision-detection and the detection of another robot, have blind spots.

In this paper we focused on building robots which are able to perform thermotaxis. The experiments were used as a proof of concept. For an exhaustive analysis are multiple robots, a larger arena and a longer period of time required. We will continue to use these robots as a model to emulate natural organisms such as honeybees. Furthermore we will test the robustness of bio-inspired algorithms to obstacles and disturbances of the gradient.

Acknowledgements

This work was supported by: EU-IST-FET project 'SYMBRION', no. 216342; EU-ICT project 'REPLICATOR', no. 216240. Austrian Science Fund (FWF) research grants: P15961-B06 and P19478-B16.

References

1. Stabentheiner, A., Schmaranzer, S., Heran, H., Ressl, R.: Verändertes Thermopräferendum von Jungbienen durch Intoxikation mit Roxion-S (Dimethoat). Mitteilungen der Deutschen G. f. allg. u. angew. Entomologie 6, 514–520 (1988)
2. Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., Crailsheim, K.: Get in touch: Cooperative decision making based on robot-to-robot collisions. Autonomous Agents and Multi-Agent Systems 18(1), 133–155 (2008)
3. Şahin, E.: Swarm robotics: From sources of inspiration to domains of application. In: Şahin, E., Spears, W.M. (eds.) SAB 2004. LNCS, vol. 3342, pp. 10–20. Springer, Heidelberg (2005)
4. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Series in Evolutionary Computation. Morgan Kaufmann, San Francisco (2001)
5. Millonas, M.M.: Swarms, phase transitions, and collective intelligence. In: Langton, C.G. (ed.) Artificial Life III, Addison-Wesley, Reading (1994)
6. Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-Organizing Biological Systems. Princeton Univ. Press, Princeton (2001)
7. Jones, J.C., Myerscough, M.R., Graham, S., Oldroyd, B.P.: Honey bee nest thermoregulation: Diversity promotes stability. Science 305, 402–404 (2004)
8. Bujok, B.: Thermoregulation im Brutbereich der Honigbiene apis mellifera carnica. Dissertation, Universität Würzburg (2005)
9. Szopek, M., Radspieler, G., Schmickl, T., Thenius, R., Crailsheim, K.: Recording and tracking of locomotion and clustering behavior in young honeybees (apis mellifera). In: Spink, et al. (eds.) Proc. Measuring Behavior, vol. 6, p. 327 (2008)
10. Schmickl, T., Hamann, H.: BEECLUST: A swarm algorithm derived from honeybees. In: Xiao, Y., Hu, F. (eds.) Bio-inspired Computing and Communication Networks. Routledge, New York (2011)
11. Braitenberg, V.: Vehicles: Experiments in synthetic psychology. MIT Press, Cambridge (1984)
12. Kantor, G., et al.: Distributed search and rescue with robot and sensor teams. In: Proc. 4th Int. C. on Field and Service Robotics, pp. 529–538 (July 2006)
13. Russell, R.: Heat trails as short-lived navigational markers for mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 4, pp. 3534–3539 (1997)

Positively versus Negatively Frequency-Dependent Selection

Robert Morris and Tim Watson

Faculty of Technology, De Montfort University,
Leicester, United Kingdom, LE1 9BH
robert.morris@email.dmu.ac.uk

Abstract. Frequency-dependent selection (FDS) refers to situations where individual fitnesses are dependent (to some degree) on where the individual's alleles lie in the proximate allele frequency distribution. If the dependence is negative – that is, if alleles become increasingly detrimental to fitness as they become increasingly common at a given locus – then genetic diversity may be maintained. If the dependence is positive, then alleles may converge at given loci.

A hypothetical evolutionary model of FDS is here presented, in which the individuals themselves determined – by means of a gene – whether their fitnesses were positively or negatively frequency-dependent. The population ratio of the two types of individual was monitored in runs with different parameters, and explanations of what happened are offered.

Keywords: Frequency-dependent selection, multiple alleles, meta gene.

1 Introduction

Ridley's textbook *Evolution* [6] contains a good entry on FDS, the opening of which is reproduced here. "Frequency-dependent selection occurs when the fitness of a genotype depends on its frequency. It is possible for the fitness of a genotype to increase (positively frequency-dependent) or decrease (negatively frequency-dependent) as the genotype frequency in the population increases."

Many abstract models of FDS have been studied, with the principal aims being to strengthen the theory underpinning these phenomena, and to explore the surrounding space of possibilities. Curtsinger [3] looked at many different selection modes, and found a condition that determined whether or not the system would stably converge. Asmussen and Basnayake [1] studied several models, focussing on the potential for the maintenance of genetic diversity, and Roff [7] looked at maintaining both phenotypic and additive variation via FDS. Bürger [2] performed an extensive analysis of a general model (of which previous known models could be considered special cases) and gave a near-complete characterisation of the equilibrium structure. Schneider [8] carried out a similar study, with multiple alleles and loci, and found no equilibria possible with more than two alleles at a locus. And Trotter and Spencer [9] investigated the potential for maintaining polymorphism taking into account the presence of positive FDS.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 77–84, 2011.

In every previous simulation of FDS, the selection regime has been imposed on the individuals by (essentially) the environment, in order to reproduce real-world conditions. In the present FDS model, the novel step is taken of putting the form of the frequency-dependence under the control of the individuals. In more precise terms, the individuals have a meta-gene in their genotype that dictates whether their fitness will be proportional to how *similar* they are to their neighbours, or how *different* they are to them. (Cf. the meta-genes in [4] and [5].) The purpose here is not to model any known natural phenomena, but to speculatively extend the domain of theoretical FDS in an interesting 'what if?' way.

2 Experiments and Analysis

A standard genetic algorithm was written whose genotype consisted of one meta gene plus a chromosome. The meta gene was a bit, and the chromosome consisted of non-negative integers in a given range. A zero allele in the meta gene told the fitness function to reward that individual for similarity, and a one told the function to reward it for difference. The measures were based on the concept of Hamming distance, and were implemented here in two different ways: pairwise, and population-wide. In the pairwise method, an individual's fitness was calculated by comparing it to a randomly chosen individual from the population (which could have been itself). If the individual in question had a meta gene of zero, its fitness was given by 1 plus the total number of genes[1] – by locus – which it had in common with the other individual, ignoring the meta genes. If the meta gene was a one, the fitness was given by 1 plus the total number of genes which they did *not* have in common. For example, for the individuals <0-21012> and <1-01022>, the fitness of the first w.r.t. the second would be 4 (= 1 + 3), whereas the fitness of the second w.r.t. the first would be 3.

In the population-wide method, an individual's fitness was calculated by comparing it to every individual in the population (including itself). This was performed by applying the pairwise method down the whole population, and tallying up the points.

These were the only components of the fitnesses.

Six runs were executed for every combination of the following parameter ranges: population size = 10 and 100; chromosome length = 1, 4, and 40; allele range = 2 (i.e. bitstrings), 3, and 4; fitness calculation = pairwise and population wide; mutation rate = zero, 'low', and 'high'; crossover = off and on (∼70% across each new population). The chromosomes were initialised randomly, but the meta-bits were set alternately to 0 and 1, to prevent biased starts. The main datum that was tracked during every run was the ratio of the two types of individual in the population per generation – this is what the vertical axes represent in most of the figures. A value of 1 indicates that the population is

[1] The minimal possible fitness was 1 so no individual could have a zero probability of being selected.

dominated by individuals whose fitness is negatively frequency-dependent (hereafter "NFDs"); a value of 0 indicates domination by individuals with positively frequency-dependent fitness (PFDs); a value of 0.5 indicates a 50:50 mixture.

2.1 Two Alleles

When the bitstring results were plotted and compared, it was seen that neither the presence of crossover nor the choice of fitness function (pairwise or population wide) made much difference. The mutation rate was not particularly important either, so long as it was neither vanishingly small nor excessively high. And the population size and the chromosome length only changed the destiny of the system when they were very small. Figure 1 shows what usually happened in the experiments for all but the extreme settings. The PFDs made more copies of themselves than the NFDs from the outset, and the population soon became dominated by PFDs. The system entered an evolutionarily stable state, which mutation and crossover could not overturn.

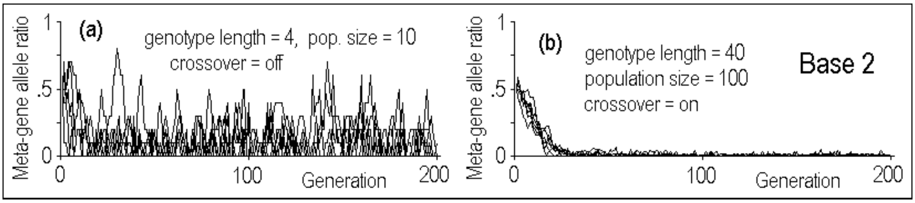


Fig. 1. How the ratio of binary PFDs (meta gene = 0) to NFDs (meta gene = 1) varied over the first 200 generations of two sets of six runs. Plot (a) shows a noisy case, and plot (b) shows a more representative case.

Why the PFDs always defeated the NFDs in base-2 (in non-extreme conditions) can be understood in the following terms. If two bitstrings are generated completely at random, the Hamming distance between them will lie somewhere between zero and their length, according to a bell-shaped probability distribution with a peak at half their length. In other words, they will probably have half their bits in common. This means that in a random initial population of 50% PFDs and 50% NFDs, the mixture of fitnesses found in the PFD group will probably be the same as that of the NFD group. In the first few generations therefore, selection will effectively be random. This means that different individuals will make varying numbers of copies of themselves, so after the first few generations, the population will comprise a number of (near-)homogenous *groups* of PFDs, a number of (near-)homogenous *groups* of NFDs, and a mixture of unique individuals of both type. (Crossover is temporally being ignored here, but when it is factored in it does not change the result.) In the pairwise fitness method, a member of a group will be compared to one of three other kinds of individual: a fellow group member, a member of another group, or one of the singletons. A PFD will get maximum points from a fellow group member, and (essentially) random points from the others. A NFD, on the other hand, will get *minimum* points from a fellow group member, and random points from the others. This means that as the population evolves, the PFDs get progressively fitter and the NFDs get progressively less fit, and at the same time, PFD groups that are similar to each other will grow faster than PFD groups that are more genetically isolated. The inevitable outcome is that the NFDs all die out as the population drives towards uniformity. The same occurs with the population-wide fitness method, because the same fitness mixture is there.

A converged PFD-only population cannot be invaded by an NFD mutant, because such a mutant would have a minimal (or near minimal) fitness compared to the maximal (or near maximal) fitnesses of the PFDs. A diverse PFD-only population is resistant to NFD mutants by an amount that negatively correlates to its diversity: if it is approaching convergence, it will have a high resistance, but if it is very mixed, then an NFD mutant could arise with a fitness comparable to those of the PFDs. However, if an NFD mutant *does* manage to get into a mixed population and starts spreading, the mutant group will die down, for the same reason as their kind dies out from the start.

2.2 Three Alleles

When the base-3 results were plotted, it was seen that the mutation rate and fitness function were similarly (ir)relevant as for base 2, but that the chromosome length and the population size were important, as was crossover in certain circumstances. One of the most important differences between the two-allele and the three-allele systems was how they behaved initially, from a random start. Whereas with two alleles, the population generally converged quickly to zeroes at the meta-locus (i.e. the PFDs dominated), with three alleles the population generally did the opposite, and converged to *ones* at the meta-locus. This was because in every initial population, any two individuals could expect to have on average only $\frac{1}{3}$ of their genes in common (as opposed to the bitstring case, where it was $\frac{1}{2}$). The NFDs could thus expect fitnesses of $\sim\frac{2}{3}$ of the maximum, while the PFDs could only expect $\sim\frac{1}{3}$ of the maximum. Consequently, the NFDs made around twice as many copies of themselves during the first few generations, so quickly wiped out the PFDs. Hence in every run (excepting some with extreme parameters) the populations took themselves into a negatively-frequency-dependent selection regime. (This is also what happens for all larger allele alphabets, so the two-allele situation is exceptional in this regard.)

These NFD convergences did not always last: they often turned out to be merely the first of two phases. When the population was large, and particularly when the chromosome was short as well, the NFD domination seemed very stable. Figure 2 shows two examples of this stability, as well as the fast convergences mentioned previously. No PFDs could invade during the timescales of the experiments, some of which went as far as 5000 generations, so these situations were the complements – in terms of the PFD:NFD ratio – of the two-allele situations (though they were not complementary in terms of diversity, because NFD populations stay diverse while PFD populations converge).

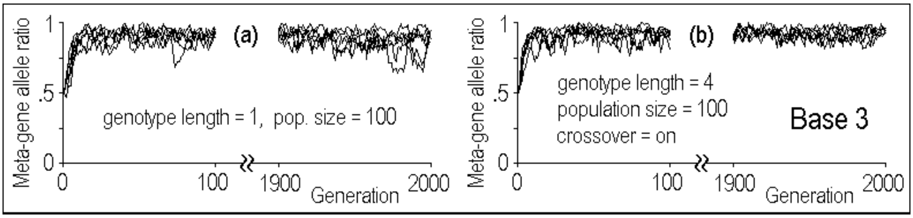


Fig. 2. How the ratio of PFDs to NFDs varied during the first and last 100 generations of two sets of six 2000-generation runs. The NFDs dominate in both cases, partly because the chromosomes were short and the populations large.

Two examples of the second phase are plotted in figure 3. In (a), which is representative of most small-population cases, the NFDs died out nearly as quickly as they took over, whereas in (b) – where the populations were large – it took varying lengths of time for the PFDs to successfully invade, with the longest being around 300 generations. As stated earlier, PFD-domination states, where the genotypes are converged or converg*ing*, are global attractors in this kind of system, and a return to NFD domination may only occur via a vastly improbable mutation and/or selection sequence.

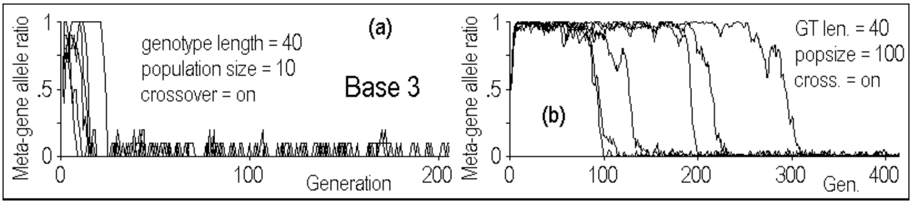


Fig. 3. How the ratio of PFDs to NFDs varied over the first 200 and 400 generations of two sets of six runs. The PFDs eventually dominate in both cases.

2.3 Multiple Alleles

The sets of results gathered for base-4 chromosomes were almost the same as their counterparts in base-3, and preliminary runs in even higher bases indicated that the patterns continue. Explanations of why certain multi-allele populations can support long-term NFD domination, but others cannot, are now offered.

It was found that the configurations most conducive to stable NFD-domination were those of very large populations and allele alphabets, but very short chromosomes, ideally single-locus. Reducing the population size, reducing the number of alleles, and increasing the chromosome length all tended to reduce the length of time the NFDs could survive before before displaced by PFDs. This result can be explained with an example. In a population of 200 NFD-individuals with base-5 single-gene chromosomes (i.e. the genotype comprises 2 genes: the meta gene [0..1] + the chromosomal gene [0..4]) where there are 40 of each allele, every individual's fitness – as measured by the inclusive population-wide method – is 161.

If an individual experienced a mutation in its meta gene, its fitness would be 41: with only $\frac{1}{4}$ the fitness of its neighbours, it would struggle to survive. And if it did survive and spread a little, its group would continue to struggle, because no matter how many copies it made, as long as there were at least two other alleles in the population, those others would be fitter. The experimental reality would be that the mutant would disappear in a generation or two's time, as would any copies it managed to make. An observer would have to wait a long time to see a PFD takeover.

Now, if the allele range is increased, the expected fitness of a PFD mutant decreases, and if the population size is increased, the amount of 'work' it must do to dominate the population increases with it. This is why those settings have the effect they do. Regarding the chromosome length, the effect of increasing it is perhaps counter-intuitive; one might have thought that longer chromosomes would have more capacity to be different from each other, thereby making NFD-domination stabler and longer lasting. Not only is this not the case, it is the opposite of the case, for the following reason. In a well-spaced-out population, at each locus there should be approximately equal ratios of the alleles across the population (so for example, in a population of 100 base-4 individuals, ∼25 individuals should have a 0 as their first gene, ∼25 should have a 1, etc.). When the chromosome is short, the low capacity for difference means that every gene is important, so every gene has 'healthy' selective pressure on it. But when the chromosome is long, a given gene is less important, as it represents only a very small portion of the distance between individuals, so the selective pressure applied on it is relatively light. Consequently, whereas for short chromosomes the local allele ratios are kept under quite tight control, for long chromosomes they can drift and become skewed. This tends to reduce the genetic distance between individuals, making it easier (to some degree) for PFD mutants to establish themselves in the population.

2.4 Crossover

The last parameter to be discussed is crossover. This operator did not discernably change the results in most of the runs, but when the populations and chromosomes were (relatively) large and the number of alleles greater than two, it made a difference. The disappearance of NFDs shown in figure 3(b) did not happen in that same time period in different runs when the only difference in settings was the absence of crossover. This may seem strange at first, when it is considered that crossover does not change population-wide gene frequencies at any loci, and that it is those frequencies that control the fitnesses. Something subtle was happening. In a mutation-only system (with a big population, long chromosomes, and multiple alleles) where NFDs are dominant, the population is usually made up of several roughly-equally-fit groups, within each of which there is homogeneity or near homogeneity. If a group happens to expand, its members' fitnesses dip, and the fitnesses of the non-members rise, so the group usually shrinks back down. (This is a standard dynamic that one finds in populations subject to negatively FDS.) The crucial observation is that when a group expands, causing certain alleles become undesirably frequent at several loci, the undesirable genes are carried by *identifiable individuals*. Evolution can therefore remove these genes by selecting against the individuals that carry them.

When crossover of any type is added, it breaks up the group structure of the population, but this in itself does not have too much impact on the system's behaviour. Crossing over can be described as 'dispersing' or 'mixing up' the alleles – at each locus – across the population. Thus, if any individuals now make extra copies of themselves, the alleles they add to the population are dispersed up and down the loci, so there are *no identifiably-bad individuals* that can be selected against. In other words, instead of selection having 'guilty' individuals it can remove, the guilt is spread across the population, so there are no longer any outstandingly guilty individuals.

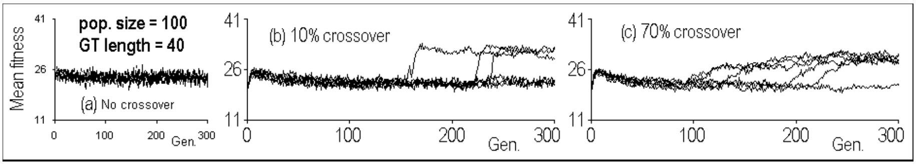


Fig. 4. The mean fitnesses over the first 300 generations of three sets of six runs

To see the exact effect crossover had, the mean fitnesses were plotted for the relevant base-3 runs. Figure 4 shows how they varied for 0%, 10%, and 70% uniform crossover, where the other parameters were the same as those of plot (b) in figure 3. Figure 4(a) shows a case where the groups scenario was played out. The fitnesses quickly rose to around $\frac{2}{3}$ of the maximum – which was to be expected with three alleles – and stayed there. The variability of the values reflects the group-sizes changing as well as the drifting of groups themselves. Plot (c) – which represents the exact same runs as those in figure 3(b) – differs in two keys way to (a). Firstly, there is a gradual drop after the initial rise, and secondly, the curves 'take off' at those times that correspond to the PFD-mutant invasions, as PFD populations are fitter than NFD ones. Plot (b) shows that a very small amount of crossover changes the system's behaviour. It is roughly the same as (c), with the key differences being higher resistances to PFD invasions, and the shorter durations of those invasions when they occur. The former can be attributed to less 'dispersion' of unwanted alleles; the latter shows that crossover slows down invasions, because the dispersals make the PFDs less similar to each other.

Further runs with more alleles and other population sizes suggested that crossover causes the mean fitness to decay geometrically from its high early value to a stable value somewhere above the halfway value. (Also, the 'dip depth' increased with the number of alleles, so the dips in figure 4 are the least severe examples of their kind.) It appears that it is in those resultant regions of stable fitness that selection can positively promote the rarer alleles as effectively as crossover can assist the commoner ones.

3 Conclusion

This paper has sought to present and explain a novel hypothetical model of frequency-dependent selection in which the individuals determine for themselves whether the frequency dependence of their fitness is positive or negative. It was found that in this particular artificial evolutionary system, the important parameters are the population size, the chromosome length, the allele range, and the presence or not of crossover.

When there are only two alleles (the binary case) a random initial population will become stably dominated by individuals whose fitness is positively frequency-dependent. When there are more than two alleles, the population will become dominated by individuals whose fitness is negatively frequency-dependent. *These* converged states vary in their stability, with their durations depending on the parameters. When they end, they end with the imposition of stable positively-FDS across the population, a state whose arrival can be hastened by making any of the following parameter changes: reducing the population size, increasing the chromosome length, and enabling crossover.

References

1. Asmussen, M.A., Basnayake, E.: Frequency-Dependent Selection: The High Potential for Permanent Genetic Variation in the Diallelic, Pairwise Interaction Model. Genetics 125, 215–230 (1990)
2. Bürger, R.: A Multilocus Analysis of Intraspecific Competition and Stabilizing Selection on a Quantitative Trait. Journal of Math. Biology 50, 355–396 (2005)
3. Curtsinger, J.W.: Evolutionary Principles for Polynomial Models of Frequency-Dependent Selection. Proc. Natl. Acad. Sci. USA 81, 2840–2842 (1984)
4. Grefenstette, J.J.: Evolvability in Dynamic Fitness Landscapes: A Genetic Algorithm Approach. In: Proc. of the 1999 Conf. on Ev. Comp., pp. 2031–2038 (1999)
5. Hillman, J.P.A., Hinde, C.J.: Evolving UAV Tactics with an Infected Genome. In: Proc. of the 5th UK Workshop on Comp. Intel., pp. 243–250 (2005)
6. Ridley, M.: http://www.blackwellpublishing.com/ridley/
7. Roff, D.A.: The Maintenance of Phenotypic and Genetic Variation in Threshold Traits by Frequency-Dependent Selection. Journal of Evolutionary Biology 11, 513–529 (1998)
8. Schneider, K.A.: A Multilocus-Multiallele Analysis of Frequency- Dependent Selection Induced by Intraspecific Competition. Journal of Math. Biology 52, 483–523 (2006)
9. Trotter, M.V., Spencer, H.G.: Frequency-Dependent Selection and the Maintenance of Genetic Variation: Exploring the Parameter Space of the Multiallelic Pairwise Interaction Model. Genetics 176, 1729–1740 (2007)

Origins of Scaling in Genetic Code

Oliver Obst[1], Daniel Polani[2], and Mikhail Prokopenko[1]

[1] CSIRO Information and Communications Technology Centre,
P.O. Box 76, Epping, NSW 1710, Australia
[2] Department of Computer Science, University of Hertfordshire, Hatfield, UK
mikhail.prokopenko@csiro.au

Abstract. The principle of least effort in communications has been shown, by Ferrer i Cancho and Solé, to explain emergence of power laws (e.g., Zipf's law) in human languages. This paper applies the principle and the information-theoretic model of Ferrer i Cancho and Solé to genetic coding. The application of the principle is achieved via equating the ambiguity of signals used by "speakers" with codon usage, on the one hand, and the effort of "hearers" with needs of amino acid translation mechanics, on the other hand. The re-interpreted model captures the case of the typical (vertical) gene transfer, and confirms that Zipf's law can be found in the transition between referentially useless systems (i.e., ambiguous genetic coding) and indexical reference systems (i.e., zero-redundancy genetic coding). As with linguistic symbols, arranging genetic codes according to Zipf's law is observed to be the optimal solution for maximising the referential power under the effort constraints. Thus, the model identifies the origins of scaling in genetic coding — via a trade-off between codon usage and needs of amino acid translation. Furthermore, the paper extends the model to multiple inputs, reaching out toward the case of horizontal gene transfer (HGT) where multiple contributors may share the same genetic coding. Importantly, the extended model also leads to a sharp transition between ambiguous HGT and zero-redundancy HGT. Zipf's law is also observed to be the optimal solution in the HGT case.

1 Introduction — Coding Thresholds

The definition and understanding of the genotype-phenotype relationship continues to be one of the most fundamental problems in biology and artificial life. For example, Woese strongly argues against fundamentalist reductionism and presents the real problem of the gene as "how the genotype-phenotype relationship had come to be" [1], pointing out the likelihood of the "coding threshold". This threshold signifies development of the capacity to represent nucleic acid sequence symbolically in terms of a amino acid sequence, and separates the phase of nucleic acid life from an earlier evolutionary stage. Interestingly, the analysis sheds light not only on this transition, but also on saltations that have occurred at other times, e.g. advents of multicellularity and language. The common feature

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 85–93, 2011.

is "the emergence of higher levels of organization, which bring with them qualitatively new properties, properties that are describable in reductionist terms but that are neither predictable nor fully explainable therein" [1].

The reason for the increase in complexity can be identified as *communication* within a complex, sophisticated network of interactions: "translationally produced proteins, multicellular organisms, and social structures are each the result of, emerge from, fields of interaction when the latter attain a certain degree of complexity and specificity" [1, 2]. The increase of complexity is also linked to adding new dimensions to the phase space within which the evolution occurs, i.e. expansion of the network of interacting elements that forms the medium within which the new level of organization comes into existence [1, 2]. An increase of complexity is one of the landmarks of self-organization, typically defined as an increase of order within an open system, without explicit external control.

As pointed out by Ferrer i Cancho and Solé [3], the emergence of a complex language is one of the fundamental events of human evolution, and several remarkable features suggest the presence of fundamental principles of organization, common to all languages. The best known is Zipf's law, which states that the frequency of a word decays as a (universal) power law of its rank. Furthermore, Ferrer i Cancho and Solé observe that "all known human languages exhibit two fully developed distinguishing traits with regard to animal communication systems: syntax [4] and symbolic reference [5]", and suggest that Zipf's law is a hallmark of symbolic reference [3].

They adopt the view that a communication system is shaped by both constraints of the system and demands of a task: e.g., the system may be constrained by the limitations of a sender ("speaker") trying to encode a message that is easy-to-decode by the receiver ("hearer"). In particular, speakers want to minimise articulatory effort and hence encourage brevity and phonological reduction, while hearers want to minimise the effort of understanding and hence desire explicitness and clarity [3, 6, 7]. For example, the speaker tends to choose ambiguous words (words with a high number of meanings), and this increases the interpretation effort for the hearer. Zipf referred to the lexical trade-off as the *principle of least effort*, leading to a well-known power law: if the words within a sample text are ordered by decreasing frequency, then the frequency of the k-th word, $P(k)$, is given by $P(k) \propto k^{-\alpha}$, with $\alpha \approx 1$.

The main findings of Ferrer i Cancho and Solé are that (i) Zipf's law can be found in the transition between referentially useless systems and indexical reference systems, and (ii) arranging symbols according to Zipf's law is the optimal solution for maximising the referential power under the effort constraints.

Combining terminology of Woese and Ferrer i Cancho and Solé allows us to re-phrase these observations as follows: (i) referentially useless systems are separated from indexical reference systems by a coding threshold, and (ii) Zipf's law maximising the referential power under the effort constraints is the optimal solution that is a feature observed at the coding threshold.

In this paper we apply the principle of least effort to genetic coding, by equating, on the one hand, the ambiguity of signals used by "speakers" with codon usage, and, on the other hand, the effort of "hearers" with demands of amino acid translation mechanics. The re-interpreted model confirms that Zipf's law can be found in the transition ("coding threshold") between ambiguous genetic coding (i.e., referentially useless systems) and zero-redundancy genetic coding (indexical reference systems). As with linguistic symbols, arranging genetic codes according to Zipf's law is observed to be the optimal solution for maximising the referential power under the effort constraints. In other words, the model identifies the origins of scaling in genetic coding — via a trade-off between codon usage and needs of amino acid translation.

This application captures the case of the typical, vertical, gene transfer. We further extend this case to multiple inputs, reaching out toward the case of horizontal gene transfer (HGT) where multiple contributors may share the same genetic coding. We observe that the extended model also leads to a sharp transition between ambiguous HGT and zero-redundancy HGT, and that Zipf's law is observed to be the optimal solution again.

2 Horizontal and Stigmergic Gene Transfer

It is important to realize that during the early phase in cellular evolution the proto-cells can be thought of as conglomerates of substrates, that exchange components with their neighbours freely — horizontally [8]. The notion of vertical descent from one "generation" to the next is not yet well-defined. This means that the descent with variation from one "generation" to the next is not genealogically traceable but is a descent of a cellular community as a whole. Thus, genetic code that appears at the coding threshold is "not only a protocol for encoding amino acid sequences in the genome but also an innovation-sharing protocol" [8], as it used not only as a part of the mechanism for cell replication, but also as a way to encode relevant information about the environment. Different proto-cells may come up with different innovations that make them more fit to the environment, and the "horizontal" exchange of such information may be assisted by an innovation-sharing protocol — a proto-code. With time, the proto-code develops into a universal genetic code.

Such innovation-sharing is perceived to have a price: it implies ambiguous translation where the assignment of codons to amino acids is not unique but spread over related codons and amino acids [8], i.e. accepting innovations from neighbours requires that the receiving proto-cell is sufficiently flexible in translating the incoming fragments of the proto-code. Such flexible translation, of course, would produce imprecise copies. However, a descent of the whole innovation-sharing community may be traceable: i.e., in a statistical sense, the next "generation" should be correlated with the previous one. While any individual protein is only a highly imprecise translation of the underlying gene, a consensus sequence for the various imprecise translations of that gene would closely approximate an exact translation of it. As noted by Polani et al. [9], the consensus sequence would capture the main information content of the innovation-sharing community.

Moreover, it can be argued that the universality of the code is a generic consequence of early communal evolution mediated by horizontal gene transfer (HGT), and that thus HGT enhances optimality of the code [8]:

> HGT of protein coding regions and HGT of translational components ensures the emergence of clusters of similar codes and compatible translational machineries. Different clusters compete for niches, and because of the benefits of the communal evolution, the only stable solution of the cluster dynamics is universality.

The work of Piraveenan et al. [10] and Polani et al. [9] investigated particular HGT scenarios where certain fragments necessary for cellular evolution begin to play the role of the proto-code. For example, *stigmeric gene transfer* was considered as an HGT variant. SGT suggests that the proto-code is present in an environmental locality, and is subsequently entrapped by the proto-cells that benefit from such interactions. In other words, there is an indirect exchange of information among the cells via their local environment, which is indicative of stigmergy: proto-cells find matching fragments, use them for coding, modify and evolve their translation machinery, and exchange certain fragments with each other via the local environment. SGT differs from HGT in that the fragments exchanged between two proto-cells may be modified during the transfer process by other cells in the locality.

SGT studies concentrated on the information preservation property of evolution in the vicinity of the "coding threshold", considering a communication channel between a proto-cell and itself at a future time point, and posing a question of the channel capacity constrained by environmental noise. By varying the nature and degree of the noise prevalent in the environment within which such proto-cells exist and evolve, the conditions for self-organization of an efficient coupling between the proto-cell *per se* and its encoding with "proto-symbols" were identified. It was shown that the coupling evolves to preserve (within the entrapped encoding) the information about the proto-cell dynamics. This information is preserved across time within the noisy communication channel. The studies verified that the ability to symbolically encode nucleic acid sequences does not develop when environmental noise φ is outside a specific noise range.

In current work we depart from the models developed by Piraveenan et al. [10] and Polani et al. [9], and rather than considering proto-cell dynamics defined via specific dynamical systems (e.g., logistic maps) subjected to environmental noise, we focus on constraints determined by (i) ambiguous codon usage, and (ii) the demands of amino acid translation mechanics. This allows us to abstract away the specifics of the employed dynamical systems [10, 9], and explain the emergence of a coding threshold from another standpoint that takes into account codon usage and amino acid translation. This, in turn, allows us to extend the model to HGT/SGT scenaria with multiple inputs. Both types of models — dynamical systems based [10, 9] and the one presented here — are able to identify an (order) parameter corresponding to the coding threshold: the environmental noise φ [10, 9] or the effort contribution λ [3]. Both types of models formulate objective functions information-theoretically, following the guidelines of information-driven self-organisation [11–13, 10, 14–17]. The main difference from the dynamical systems based models lies, however, in the ability to detect a power law in the codon usage (lexicon) at the threshold.

3 Model

The Ferrer i Cancho and Solé model [3] is based on an information-theoretic approach, where a (single) set of signals S and a set of objects R are used to describe signals between a speaker and a hearer, and the objects of reference. The relation between S and R (the communication channel) is modelled using a binary matrix $\mathbf{A}$, where an element $a_{i,j} = 1$ if and only if signal s_i refers to object r_j. The effort for the speaker is low with a high amount of ambiguity, i.e. if signal entropy is low. $H_n(S)$ expresses this as a number between 0 and 1:

$$H_n(S) = -\sum_{i=1}^{n} p(s_i) \log_n p(s_i)$$

The effort for the hearer to decode a particular signal s_i is small if there is little ambiguity, i.e. the probability of a signal s_i referring to one object r_j is high. In [3], this is expressed by the conditional entropy

$$H_m(R|s_i) = -\sum_{j=1}^{m} p(r_j|s_i) \log_m p(r_j|s_i)$$

Hearer effort is dependent on probability of each signal and effort to decode:

$$H_m(R|S) = \sum_{i=1}^{n} p(s_i) H_m(R|s_i)$$

A cost function $\Omega(\lambda) = \lambda H_m(R|S) + (1-\lambda) H_n(S)$ is introduced to combine effort of speaker and hearer, with $0 \leq \lambda \leq 1$ trading off the effort between speaker and hearer. To consider effects of different efforts of speaker and hearer, different λ are used to compute the *accuracy of the communication* as the mutual information, $I_n(S,R) = H_n(S) - H_n(S|R)$. This uses matrices evolved for minimal cost $\Omega(\lambda)$, and a simple mutation-based genetic algorithm. In addition, L will denote the number of signals referring to at least one object, i.e. the *effective lexicon size.*

3.1 Extension of the Model and "Readout" Modeling

To model codon usage by several neighbours, we extend the original approach that was using one matrix $\mathbf{A}$, to several matrices $\mathbf{A}_k$, each of which represents a different way of codon usage for the same set of objects (amino acids). In the extended model, a separate matrix $\mathbf{A}_k$ is created and evolved to encode a different communication channel. The cost $\Omega(\lambda)$ is computed for a matrix $\hat{\mathbf{A}}$ that is obtained by averaging over all individual matrices $\mathbf{A}_k$. During evolution,

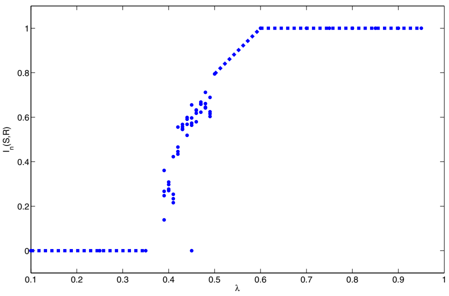


Fig. 1. One 150x150 matrix. The mutual information as function of λ.

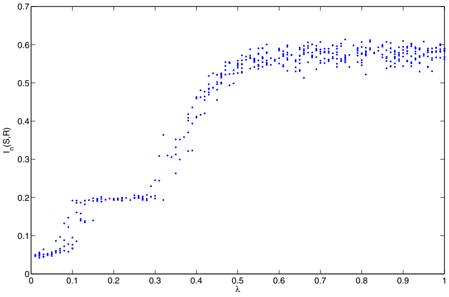


Fig. 2. HGT – four 40x40 matrices. The mutual information as function of λ.

if the cost of $\hat{\mathbf{A}}$, obtained from mutated $\mathbf{A}_k$, is reduced, the mutation is kept. Averaging captures SGT between different sources (variables), and is a specific case of "readout", motivated below.

Shannon information between two variables X and X' is determined as an optimum of knowledge extracted from the state of X about the state of X' *under the assumption that both variables and their joint distribution have been identified beforehand*. In our model, however, the transfer of a message fragment from one proto-cell to another does not, in general, enjoy that advantage, because there are multiple candidates for the source of the message. The stigmergic nature of the message transmission in the HGT scenario does not allow for a priori assumptions of who the sender is and who the receiver is. This implies that there might emerge a pressure to "homogenize" the instantiations of sender and the receiver variables in the sense that, where information is to be shared, an instantiation x in the sender variable X evokes to some extent the same instantiation in the receiver variable X'.

To formalize this intuition, we model the sending (and analogously receiving) proto-cells as mixtures of probabilities, endowed with the "readout" interpretation suggested in [13] which we sketch in the following. Assume a collection of random variables X_k, indexed by an index $k \in \mathcal{K}$ for each sender, where all X_k assume values $x_k \in \mathcal{X}$ in the same sampling set $\mathcal{X}$. For a fixed, given $k \in \mathcal{K}$, one can now determine informational quantities involving X_k in the usual fashion; however, if, as in the HGT case, the sender is not known in advance, one can model that uncertainty as a probability distribution over the possible indices $k \in \mathcal{K}$, defining a random variable K with values in $\mathcal{K}$. If nothing else is known about the sender, one can for instance assume an equidistribution on K.

The *readout* of the collection $(X_k)_{k\in\mathcal{K}}$ is then denoted by X_K which models the random selection of one $k \in \mathcal{K}$ according to the distribution of K which of the X_k, followed by a random selection of an instance $x \in \mathcal{X}$ according to the distribution of X_k. For a Bayesian network interpretation of the readout, see [13]. Formally, the probability distribution of the readout X_K assuming a value $x \in \mathcal{X}$ is given by $\Pr(X_K = x) = \sum_{k\in\mathcal{K}} \Pr(K = k) \cdot \Pr(X_k = x)$.

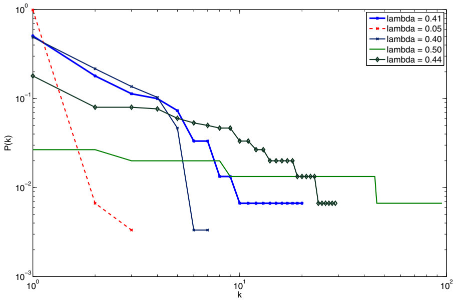


Fig. 3. Single 150x150 matrix. Signal normalised frequency, $P(k)$, versus rank, k.

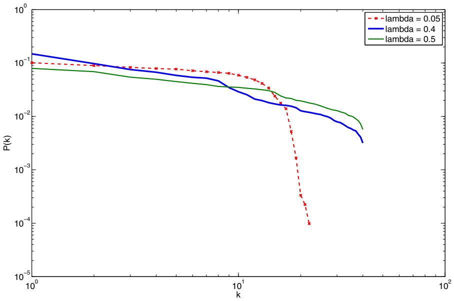


Fig. 4. Four 40x40 matrices. Signal normalised frequency, $P(k)$, versus rank, k.

4 Results

The accuracy of communication $I_n(S, R)$ as a function of λ is shown in Fig. 1 and 2. Figure 1 traces mutual information for a single 150x150 matrix (staying within the model in [3], and studying a typical vertical gene transfer), while Fig. 2 contrasts the dynamics within HGT, simulated with four 40x40 matrices. In both, three domains are distinguishable in the behavior of $I_n(S, R)$ vs. λ. For small values $\lambda < \lambda^*$, $I_n(S, R)$ grows smoothly, before undergoing a sharp transition in the vicinity $\lambda \approx \lambda^*$. Following Ferrer i Cancho and Solé, we observe that single-signal systems (effective lexicon size $L \approx 1/n$) dominate for $\lambda < \lambda^*$: "because every object has at least one signal, one signal stands for all the objects" [3]. Low $I_n(S, R)$ indicates that the system is unable to convey information in this domain (totally ambiguous genetic code). Rich vocabularies (genetic codes with some redundancy), $L \approx 1$, are found after the transition, for $\lambda > \lambda^*$. Full vocabularies are attained for very high λ. The maximal value of $I_n(S, R)$ indicates that the associations between signals (codons) and objects (amino-acids) are one-to-one maps, removing any redundancy in the genetic code. In the HGT case, this is harder to achieve, while the overall tendency is retained.

To investigate transition around $\lambda \approx \lambda^*$, we focus on the lexicon's ranked distribution, and consider the signal's normalised frequency $P(k)$ vs. rank k, for different λ. As expected, Ferrer i Cancho and Solé model shows that "Zipf's law is the outcome of the nontrivial arrangement of word-concept associations adopted for complying with hearer and speaker needs" [3]. Contrasting the graphs (Fig. 3) reveals the presence of scaling for $\lambda \approx \lambda^*$, and suggests a phase transition is taking place at $\lambda^* \approx 0.44$ in the information dynamics of $I_n(S, R)$. This results in a power law ($P(k) = 0.2776k^{-1.0146}$), consistent with $\alpha \approx 1$ reported in [3].

Similar phenomenon is observed for HGT (Fig. 4). Scaling at $\lambda^* = 0.4$ results in the power law ($P(k) = 0.2742k^{-1.0412}$, with $R^2 = 0.9474$), again consistent with $\alpha \approx 1$ in the power law reported in [3] and the one reported for vertical gene transfer above. Thus, the trade-off between codon usage and needs of amino acid translation in HGT results in a nontrivial but still redundant genetic code.

5 Conclusions

We applied the principle principle of least effort in communications and the (extended) information-theoretic model of Ferrer i Cancho and Solé to genetic coding. The ambiguity of signals used by "speakers" was equated with codon usage, while the effort of "hearers" provided an analogue for the needs of amino acid translation mechanics. The re-interpreted model captures the case of the typical (vertical) gene transfer, and confirms presence of scaling in the transition between referentially useless systems (i.e., ambiguous genetic coding) and indexical reference systems (i.e., zero-redundancy genetic coding). Arranging genetic codes according to Zipf's law is observed to be the optimal solution for maximising the referential power under the effort constraints. Thus, the model identifies the origins of scaling in genetic coding — via a trade-off between codon usage and needs of amino acid translation. The extended model includes multiple inputs, representing HGT where multiple contributors may share the same genetic coding. The extended model also leads to a sharp transition: between ambiguous HGT and zero-redundancy HGT, and scaling is observed to be the optimal solution in the HGT case as well.

References

1. Woese, C.R.: A new biology for a new century. Microbiology and Molecular Biology Reviews 68(2), 173–186 (2004)
2. Barbieri, M.: The organic codes: an introduction to semantic biology. Cambridge University Press, Cambridge (2003)
3. Ferrer i Cancho, R., Solé, R.V.: Least effort and the origins of scaling in human language. PNAS 100(3), 788–791 (2003)
4. Chomsky, N.: Language and Mind. Harcourt, Brace, and World, New York (1968)
5. Deacon, T.W.: The Symbolic Species: The Co-evolution of Language and the Brain. Norton & Company, New York (1997)
6. Pinker, S., Bloom, P.: Natural language and natural selection. Behavioral and Brain Sciences 13(4), 707–784 (1990)
7. Köhler, R.: System theoretical linguistics. Theor. Ling. 14(2-3), 241–257 (1987)
8. Vetsigian, K., Woese, C., Goldenfeld, N.: Collective evolution and the genetic code. PNAS 103(28), 10696–10701 (2006)
9. Polani, D., Prokopenko, M., Chadwick, M., Modelling, M.: stigmergic gene transfer. In: Bullock, S., Noble, J., et al. (eds.) Artificial Life XI - Proc. 11th Int. Conf. on the Simulation and Synthesis of Living Systems, pp. 490–497. MIT Press, Cambridge (2008)
10. Piraveenan, M., Polani, D., Prokopenko, M.: Emergence of genetic coding: an information-theoretic model. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 42–52. Springer, Heidelberg (2007)
11. Prokopenko, M., Gerasimov, V., Tanev, I.: Evolving spatiotemporal coordination in a modular robotic system. In: Nolfi, S., Baldassarre, G., et al. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 558–569. Springer, Heidelberg (2006)

12. Polani, D., Nehaniv, C., Martinetz, T., Kim, J.T.: Relevant information in optimized persistence vs. progeny strategies. In: Rocha, L., Yaeger, L., Bedau, M., Floreano, D., Goldstone, R., Vespignani, A. (eds.) Artificial Life X: Proc. 10th International Conference on the Simulation and Synthesis of Living Systems (2006)
13. Klyubin, A., Polani, D., Nehaniv, C.: Representations of space and time in the maximization of information flow in the perception-action loop. Neural Computation 19(9), 2387–2432 (2007)
14. Laughlin, S.B., Anderson, J.C., Carroll, D.C., de Ruyter van Steveninck, R.R.: Coding efficiency and the metabolic cost of sensory and neural information. In: Baddeley, R., Hancock, P., Földiák, P. (eds.) Information Theory and the Brain, pp. 41–61. Cambridge University Press, Cambridge (2000)
15. Bialek, W., de Ruyter van Steveninck, R.R., Tishby, N.: Efficient representation as a design principle for neural coding and computation. In: 2006 IEEE International Symposium on Information Theory, pp. 659–663. IEEE, Los Alamitos (2006)
16. Piraveenan, M., Prokopenko, M., Zomaya, A.Y.: Assortativeness and information in scale-free networks. European Physical Journal B 67, 291–300 (2009)
17. Lizier, J.T., Prokopenko, M., Zomaya, A.Y.: Local information transfer as a spatiotemporal filter for complex systems. Physical Review E 77(2), 026110 (2008)

Adaptive Walk on Fitness Soundscape

Reiji Suzuki and Takaya Arita

Graduate School of Information Science, Nagoya Univesity
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
{reiji,arita}@nagoya-u.jp
http://www.alife.cs.is.nagoya-u.ac.jp/~{reiji,ari}/

Abstract. We propose a new IEC for musical works based on an adaptive walk on a fitness landscape of sounds. In this system, there is a virtual plane that represents the genetic space of possible musical works called fitness soundscape. The user stands on the soundscape, and hears the multiple sounds that correspond to one's neighboring genotypes at the same time. These sounds come from different directions that correspond to the locations of their genotypes on the soundscape. By using the human abilities for localization and selective listening of sounds, the user can repeatedly walk toward the direction from which more favorite sounds come. This virtual environment can be realized by a home theater system with multiple speakers creating "surrounded sound". We report on the basic concept of the system, a simple prototype for musical composition with several functional features for improvement of evolutionary search, and preliminary evaluations of the system.

Keywords: interactive evolutionary computation, musical composition, fitness landscape, soundscape, virtual reality, artificial life.

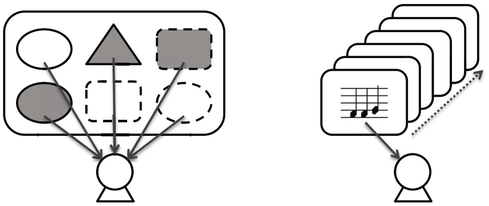
1 Introduction

An interactive evolutionary computation (IEC) has been used for optimizing various artifacts which are not possible to be evaluated mechanically or automatically [1]. Based on subjective evaluations of the artifacts by a human, one's favorite artefacts in the population are selected as parents for the individuals in the next generation. By iterating this process, one can obtain better artifacts without constructing them by oneself.

IECs have been widely applied in various artistic fields. Especially, IECs have been applied in creation of musical works such as musical composition, sound design, and so on [2]. For example, Unemi constructed a musical composition system named SBEAT in which a multi-part music can be generated through simulated breeding [3]. Dahlstedt also proposed a system for synthesizing a sound with interactive evolution, in which each sound has a visual representation which corresponds to the set of parameters of the synthesizer for generating the sound [4].

In IECs for graphical works such as Dawkins's Biomorph [5], the individuals in the population can be evaluated in parallel as shown in Fig. 1 (a). On the other

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 94–101, 2011.



(a) a parallel evaluation of graphical works (b) a sequential evaluation of musical works

Fig. 1. The interactive evolutionary computation with a) parallel or b) sequential evaluations

hand, in IECs for musical works, the user listens to and evaluates each individual in the population one by one, as shown in Fig. 1 (b). This is due to the fact that it is basically thought as difficult to evaluate each individual correctly when the all individuals are played all at once. This is an essential difference of IECs for musical works compared with those for graphical works. Unfortunately, this sequential evaluation of individuals increases the total evaluation time and thus increases the user's temporal cost significantly. It also increases one's cognitive cost in that one needs to remember the features of individuals so as to compare between them. Thus, we often limit the population size small in order to decrease these costs, which brings about the fitness bottleneck [6].

So as to solve these problems, we focus on the cognitive abilities of human for localization and selective listening of sounds. We can localize the direction of sounds, and can concentrate on the sound we like to hear, which is called a *cocktail party effect* [7] in a broad sense. If we utilize these kinds of ability properly, there is a possibility that we can correctly evaluate the individuals in parallel even in the case of IECs for musical works.

We propose a new IEC for musical works based on an adaptive walk on the fitness landscape of sounds. In this system, there is a two-dimensional virtual plane that represents the genotypic space of possible musical works called fitness soundscape. The user stands on the soundscape, and hears the multiple sounds that correspond to one's neighboring genotypes at the same time. The sounds come from different directions that correspond to the locations of their genotypes on the soundscape. This virtual environment can be realized by a home theater system with multiple speakers creating "surrounded sound". By using the ability for localization and selective listening of sounds, the user can repeatedly walk toward the direction from which more favorite sounds come, which corresponds to the evolutionary process of the population in standard IECs.

In this paper, we introduce the basic concept of the proposed system, and a simple prototype system for musical composition. We also propose the several functional features for improvement of evolutionary search. The preliminary experiments showed that the user successfully obtained one's favorite musical work with the help of these features.

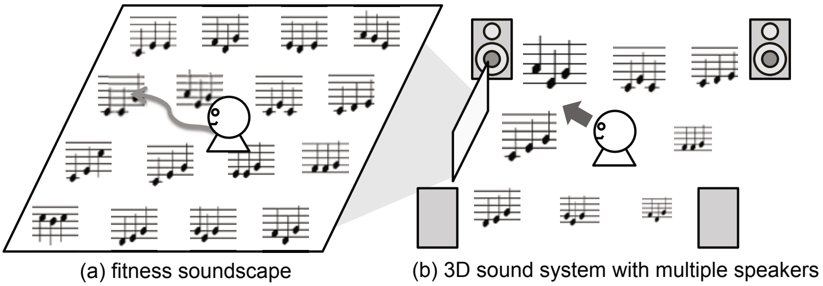


Fig. 2. A conceptual image of adaptive walk on fitness soundscape

2 Adaptive Walk on Fitness Soundscape

We propose an IEC for musical works based on an adaptive walk on a fitness landscape of sounds. Here, we introduce the basic concept of the model.

The conceptual image of the model is shown in Fig. 2. A fitness landscape is often used so as to visualize and intuitively understand evolutionary dynamics of the population [8]. We assume a two-dimensional plane of genotypes that determine the individuals' phenotypes. In this model, the phenotypes represent possible musical works such as musical pieces that are going to be searched, as shown in Fig. 2 (a). Thus, we call this landscape of sounds the fitness soundscape.

The user stands on this plane and hears the sounds of one's neighboring individuals from their corresponding position all at once. This virtual sound environment can be realized by a home theater system with multiple speakers creating "surrounded sound" as shown in Fig. 2 (b).

The height of the fitness landscape is the corresponding fitness value of a genotype on a possible genetic space. The fitness of each genotype on the soundscape can be determined by a subjective evaluation by the user. Thus, the actual shape of the soundscape will be different among users depending on their subjective impression of individuals, and can change dynamically.

The adaptive evolution of the population can be represented by a hill-climbing process on the fitness landscape. We adopt this process on the fitness soundscape as an evolutionary mechanism of the model. The user can walk toward the direction from which one's favorite individuals' sounds come. The user can obtain one's more favorite sounds by repeating this hill-climbing process.

3 Prototype

We constructed a prototype of the system, which includes the several functions for improvement of an adaptive walk on the fitness soundscape. The main part of the system was constructed using Java 3D with JOAL (the Java Bindings for OpenAL API)[1], and we used the software synthesizer TiMidity++[2] for the dynamic

[1] https://joal.dev.java.net/

[2] http://timidity.sourceforge.net/

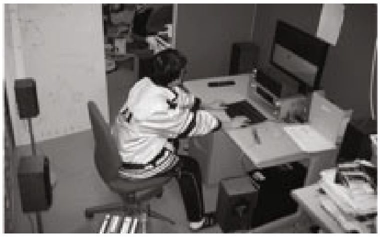


Fig. 3. A snapshot of the system

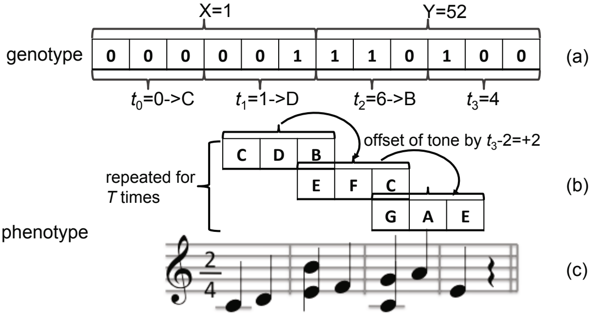


Fig. 4. The genetic description of a musical piece in the case of T=3

generation of the sound files. The real 3D sound environment was composed of a multi-channel home theater system with 7 speakers as shown in Fig. 3. A surround headphone for the home theater system with 5.1 channels can be used instead of this speaker system. We explain each part of the prototype in detail.

3.1 Genetic Description of the Individual

We assumed an evolutionary search of a musical piece as shown in Fig. 4, which is quite simple but sufficient to evaluate and demonstrate the prototype system, especially the parallel search utilizing the cocktail party effect. It also shows the genetic description of a musical piece, which is represented by a bit string with the length 12. Specifically, each genotype can be divided into 4 parts, each of which represents an integer value from 0 to 7. They determine the values of the parameters $t_0, \cdots, t_3$ respectively, as shown in Fig. 4 (a). The first three parameters determine the tones of the initial set of the three notes respectively as shown in Fig. 4 (b). The possible values of the parameter "01234567" correspond to the tones "CDEFGABC". The tones of notes in the next set are defined as those of the corresponding notes in the previous set offset by t_3-2 within the range of possible tones. This subsequent set starts to be played at the same timing of playing the last note in the previous set. By repeating this process, the T sets are generated. Fig. 4 (c) is the final phenotype of this example.

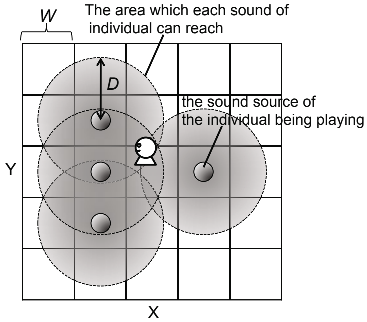

W
The area which each sound of individual can reach
D
the sound source of the individual being playing
Y
X

Fig. 5. An example situation of the soundscape around the user

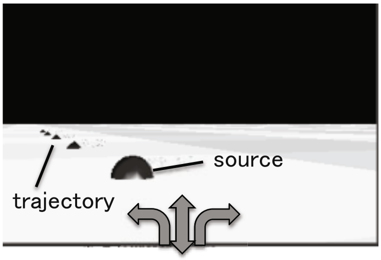


Fig. 6. A three dimensional image of the soundscape

3.2 Fitness Soundscape and Adaptive Walk

We mapped the genotypes described in the previous section into a two-dimensional space using the following simple way. We divided the bit string of each genotype into two parts, and regarded the former part as the x position $(0, \cdots, 2^6 - 1)$ of each genotype and the latter part as its y position $(0, \cdots, 2^6 - 1)$ as shown in Fig. 4 (a).

The fitness soundscape is composed of the $2^6 \times 2^6$ square-shaped areas, and each genotype (phenotype) is assigned to each corresponding area as explained above. Fig. 5 is an example situation of the soundscape around the user. The size of each area is $W \times W$ in units of Java 3D environment. When the user stands on the soundscape, at most 8 individuals on one's neighboring areas are played repeatedly in parallel. The user can determine which individuals are actually played by specifying their relative directions from one's heading by using a control panel (explained later). The sound source of each individual is placed at the center of each corresponding area. It is omnidirectional and its gain decreases linearly to 0 percent at the distance of D. In Fig. 5, the user specified that one can hear the individuals in front center, front right, front left, and rear center. However, one does not hear the sound in front left, because its distance from the user is out of the range of D. It should be noted that we do not play the individual which corresponds to the user's standing area. It is due to the fact that it can make the user difficult to localize other neighboring sounds of individuals because it is too close to the user, besides that it is not necessary for deciding the direction of evolution.

When the user has moved to the next area, the new neighboring individuals around the user's standing area are played. Thus, the user can repeatedly walk around the whole soundscape hearing and evaluating the neighboring individuals. Fig. 6 shows the three dimensional image of the fitness soundscape, which is presented to the user during adaptive walk. The user can grasp the trajectory of one's adaptive walk so far, and the relative positions of the individuals that are played now.

3.3 Functional Features for Enhancement of Adaptive Walk

So as to improve the ability of evolutionary search, we introduced the following functional features into the prototype system. The several properties of these functions can be set up using the control panel of the system shown in Fig. 7.

The adjustment of the scale of the soundscape. The user can adjust the scale of the soundscape by changing the hamming distance between the nearest neighboring genotypes, which is realized by eliminating the areas which correspond to the intermediate genotypes. By decreasing the scale ratio, the user can evaluate more different individuals at the same time, and walk toward a more distant place quickly.

The assingment of the relative positions of individuals being playing. If the all neighboring individuals are always played, there is a possibility that the user can not discriminate between them because they are too much mixed. Thus, we added the set of options that determine the relative directions of the neighboring area in which the individuals can be played. The user can adjust the total number of the individuals being playing and their relative positions from the one's heading by choosing activated individuals from the sorrounding ones in the control panel.

The random delay in the timing of play. So as to make the user easy to distinguish between individuals, we have introduced a randomly determined time interval before playing each individual.

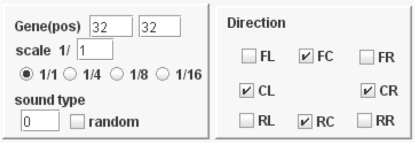
The use of different sound types. We have also added the option whether the all individuals are played using the same sound type or are played using randomly assigned sound types. The difference in the sound types enables the user to listen to each individual more selectively.

4 Preliminary Evaluations

We have conducted the several preliminary experiments using the prototype explained in the previous section. First, we summarize basic evaluations on adaptive walk on the soundscape. By using the multiple speaker systems, we were able to localize the positions of individuals and distinguish between them even when they were played in parallel. We were also able to evaluate individuals, and walk toward the favorite ones. Thus, we can say that the user can search for favorite musical pieces based on one's subjective evaluations using this system.

Fig. 8 shows example trajectories of adaptive walk on the soundscape when the 5 different subjects searched for their favorite musical pieces. We used the settings of parameters: W=5, D=7.5 and T=4. The sounds are played with a randomly determined delay, and we adopted a unique sound type (piano) for all individuals. The relative directions of individuals being playing are front center, rear center, center right, and center left. The subjects were only allowed to use the adjustment of the scale of the soundscape.

Starting from the center of the soundscape, the subjects moved over 10-80 areas, and finally reached their favorite piece whose corresponding position were



Gene(pos) 32 32
scale 1/ 1
● 1/1 ○ 1/4 ○ 1/8 ○ 1/16
sound type
0 □ random

Direction
□ FL ☑ FC □ FR
☑ CL ☑ CR
□ RL ☑ RC □ RR

Fig. 7. The control panel of the prototype

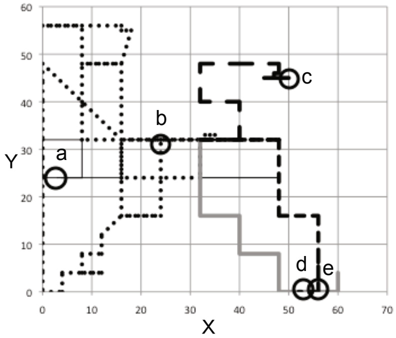


Fig. 8. Example trajectories of evolutionary search on the soundscape

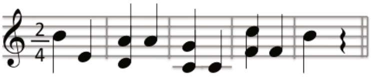


Fig. 9. An evolved musical piece

indicated with a circle on the map. As the figure shows, the trajectories varied significantly, which implies that the aesthetic feeling for musical pieces of the subjects were basically different. It is also interesting that the two subjects finally reached the closer positions (d and e). Fig. 9 is an evolved sound piece which corresponds to the position c (50, 45) in Fig. 8.

It also turned out the following points with regard to the functional features. By decreasing the scale of the soundscape during early period of the search, the user could quickly reach one's favorite areas in a long distance from the initial area. It was not easy to localize the individuals in backward positions of the user compared with those in other relative positions. Thus, there is a possibility that we can improve the ability for selective listening by adjusting the number or assignment of the individuals being playing in backward positions. Also, the random delay in the timing of play and the use of different sound types made the users much easier to distinguish between individuals. However, the latter sometimes affected the intuitive impressions of the individuals strongly, and thus affected their evaluations.

In addition, a searching process on the soundscape was itself a new experience, which implies that this system can be extended to a kind of installation.

5 Conclusion

We have proposed a new interactive genetic algorithm for musical works utilizing the human abilities of localization and selective listening of sounds, which is based on adaptive walk on the fitness landscape of sounds. We constructed a prototype of the system for simple musical pieces, and conducted a preliminary evaluation of it. It was confirmed that the users were able to search for their favorite pieces by using this system. We also proposed the several functional features for improving evolutionary search.

Future work includes the more detailed evaluation of the system, the improvement of evolutionary search, and the application of this system to the interface of content selection in portable music players.

Acknowledgements

This work was supported in part by a grant from the Sound Technology Promotion Foundation.

References

1. Takagi, H.: Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. Proceedings of the IEEE 89(9), 1275–1296 (2001)
2. Husband, P., Copley, P., Eldridge, A., Mandelis, J.: An Introduction to Evolutionary Computing for Musicians. In: Miranda, E.R., Biles, J.A. (eds.) Evolutionary Computer Music, pp. 1–27 (2007)
3. Unemi, T.: SBEAT3: A Tool for Multi-part Music Composition by Simulated Breeding. In: Proceedings of the Eighth International Conference on Artificial Life, pp. 410–413 (2002)
4. Dahlstedt, P.: Creating and Exploring Huge Parameter Spaces: Interactive Evolution as a Tool for Sound Generation. In: Proceedings of the International Computer Music Conference 2001, pp. 235–242 (2001)
5. Dawkins, R.: The Blind Watchmaker. Longman, Harlow (1986)
6. Biles, J.A.: GenJam: A Genetic Algorithms for Generating Jazz Solos. In: Proceedings of the 1994 International Computer Music Conference (1994)
7. Cherry, E.C.: Some Experiments on the Recognition of Speech with One and with Two Ears. Journal of the Acoustical Society of America 25, 975–979 (1953)
8. Wright, S.: The Roles of Mutation, Inbreeding, Crossbreeding, and Selection in Evolution. In: Proceedings of the Sixth International Congress on Genetics, pp. 355–366 (1932)

Breaking Waves in Population Flows

George Kampis[1,2] and Istvan Karsai[3]

[1] Collegium Budapest, Institute for Advanced Study,
Budapest, Hungary
[2] Eötvös University, Department of Philosophy of Science,
Budapest, Hungary
[3] Department BISC, East Tennessee State University,
Johnson City, TN, USA
gkampis@colbud.hu

Abstract. We test the controversial ideas about the role of corridors in fragmented animal habitats. Using simulation studies we analyze how fragmentation affects a simple prey-predator system and how the introduction of openings that connect the habitats changes the situation. Our individual based model consists of 3 levels: renewable prey food, as well as prey and predators that both have a simple economy. We find, in line with intuition, that the fragmentation of a habitat has a strong negative effect especially on the predator population. Connecting the fragmented habitats facilitates predator (and hence prey) survival, but also leads to an important counterintuitive effect: in the presence of a high quality predator, connected fragmented systems fare better in terms of coexistence than do unfragmented systems. Using a frequency domain analysis we explain how corridors between sub-habitats serve as "wave breakers" in the population flow, thus preventing deadly density waves to occur.

Keywords: predator prey systems, animal corridors, frequency domain analysis.

1 Introduction

Wave patterns are common in biology. Spatio-temporal waves have been demonstrated in many population models. In particular, various waveforms including spiral waves can be generated in simple predator prey systems [e.g., 1]. The stability and coexistence of populations is often closely related to the existence and behavior of wave patterns. High amplitude oscillations or transients tend to destabilize systems by generating depleted resource conditions. Therefore, the understanding and active control of population waves is of high importance in a number of contexts. For example, fragmented habitats tend to produce high densities that often lead to fatal oscillations.

General spatial models in ecology, including island biogeographic models [2] as well as meta-population models [3, 4] predict that movements between patches will increase effective population size and persistence in general. Haddad & Tewksbury [5] reviewed major ecology and general biology journals from 1997 to 2003 and found only 20 studies to test the corridors' effects on populations and diversity. They concluded that the current evidence offers only tentative support for the positive

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 102–109, 2011.

effects of corridors, and much more work on population and community responses is needed, especially, that it is important to study the mechanisms and conditions under which we can expect corridors to impact populations. The authors also predicted an increasing importance of individual based models that can aid empirical studies by focusing on the effects of different life history parameters [5].

In this paper we present and agent based model to study the oscillatory behavior of populations in various fragmented and connected fragmented habitats, respectively. The model utilizes an approach inspired by studies of excitable media, hydrodynamics, and granular flows. Hydrodynamic analogies can often help understand wave phenomena in different domains. Wave control tools (such as wave breakers and barriers) are being extensively used in natural flow systems such as rivers, ocean shores, etc. Also, granular flows have been successfully applied to the modeling and control of the behavior of humans in mass situations such as when escaping from fire, in high traffic or in mass demonstrations [6] . We utilize a similar metaphor in the present model, where population waves are shown to be controllable by a combination of barriers and restricted passages (i.e., corridors).

2 The Model

The model (written in NetLogo 3.14) uses a simple set of assumptions:

- The system consists of predator and prey individuals, which feed on biotic resources. Prey food has an autonomous growth dynamics with saturation.
- The model is spatially explicit, consisting of *n* x *n* locations. A prey individual consumes a food token on its current location, if there is one. Similarly, the predator consumes a single prey individual if prey is found at the given location. All these behaviors are translated into a common currency, "energy". Consumption ensures a certain amount of energy, i.e., $Gain_{PY}$ and $Gain_{PD}$ for the prey and predator, respectively. The consumed individual dies, and it is removed from the system. Death also happens if the energy level of an individual reaches zero.
- At each time step every individual is forwarded by a constant amount *Fd* (this speed is assumed identical for both predator and prey). Each move occurs in a uniformly selected random direction taken from the interval $\pm T$ (the degree of turning in degrees relative to the current orientations). One step costs exactly one energy token (stored "energy" thus directly converts to lifetime).
- The habitat is modeled as a rectangular area with reflecting boundaries. Each position (except the borders) can be empty or occupied by an arbitrary number of individuals (except that prey food that can only be on or off at a given location).
- Reproduction is asexual and occurs with probability R_{PY} and R_{PD}, for prey and predator. Upon reproduction, a new individual of the given type is produced, with a random spatial orientation. Energy tokens of the parent individual will be shared evenly with the offspring. Prey food regenerates in *K* steps.
- Prey and predator start from random initial positions as well as orientation and receive a randomized amount of startup energy (between *0* and E_{PY} or E_{PD} respectively). At every discrete time step, the following sequence of actions is performed for each individual organism in a dynamically randomized order: move, consume available resources, reproduce, and die if energy is out.

Borders with reflecting walls implement fragmentation in the habitat. Borders (similar to real roads, canals, or fences) do not decrease the overall habitat size significantly, and behave in the same way, as does the outer boundary. Corridors are implemented as openings in the walls where the organisms can pass through freely. Corridors have zero length (beyond the wall thickness) and are wide enough to permit several organisms to cross at the same time.

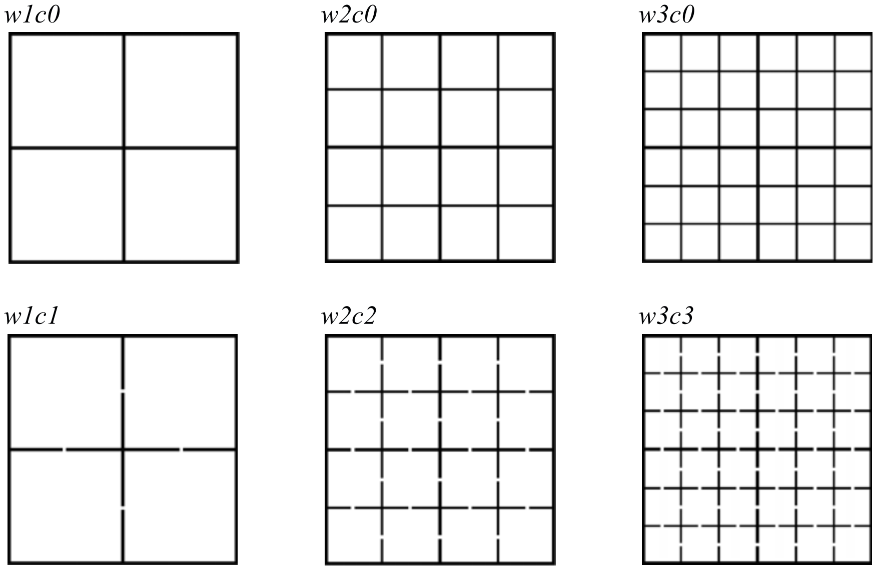


Fig. 1. Fragmentation schemes studied in the paper: *wXcY* is an *X* walls and *Y* corridors system

The main numerical control parameter of the model is $Gain_{PD}$, i.e., the amount of energy gained by the predator when a prey individual is consumed (this can be thought of as representing a "quality" parameter for the predator: thus, the parameter is used as an umbrella descriptor for many direct and indirect relationships between several individual predator traits). In the model, high levels of predator gain and the resulting higher energy reserves of the predators imply a higher expected predator lifetime, which corresponds to an efficient natural predator that can control a large area.

An additional, and most important, control parameter is the number of boundaries and openings in the system. We studied several different combinations (see Fig. 1.) Baseline settings define an initial area large enough to support a high population size (about *10,000* individuals) of prey and predator (Table 1.). Under the baseline conditions, in the unfragmented area (*w0c0*) the prey population always persists indefinitely without the predators, while prey and predator tend to coexist together.

In the experiments, the value of $Gain_{PD}$ and the number of boundaries and corridors were varied, while all other parameters were kept unchanged. Many of these parameters have various effects on dynamic properties that we cannot study in this paper. Treatments consisted of an exhaustive combination of the values $Gain_{PD}$ = 10 (baseline), 30, 50 and 70, with a complete set of boundary-corridor systems as presented on Fig.1. (For each treatment, we started simulation runs with a delay t=50 in initial prey food re-growth, to avoid extreme initial prey densities and follow-up extinctions.) A high number (≥ 50) of simulation runs with different random seeds and $t = 4,596$ time steps were carried out for each treatment. Data were analyzed and plotted using the R statistical package.

Table 1. Model parameters and initial values used in the simulations

Initial prey N_{PY} = 1,000; predator N_{PD} = 100	n area size = 200
Fertility $R_{PY} = R_{PD}$= 15 %	Corridor width = 3
Motion speed Fd = 0.9;	Max. turning T = 50
$Gain_{PY}$ = 4, $Gain_{PD}$ = varied (10, 30, 50, 70)	Regeneration time K = 5
Initial energy maximum $E_{PY} = 2*Gain_{PY}$, $E_{PD} = 2*Gain_{PD}$	

3 Results and Discussion

The various effects of fragmentation and corridors exerted on the quantitative behavior of coexistence dynamics has been explored in (Karsai and Kampis 2011, submitted). The basic – and seemingly counterintuitive – finding is that, although fragmentation decreases the chance of survival for the predator (as expected), yet a combined effect of fragmentation and corridors produces a situation favoring the coexistence and stable dynamics of prey and predator above the chance of coexistence found in the unfragmented system. In the present paper we study the oscillatory behavior and the effect of boundaries and corridors on the emerging spatio-temporal waves understood as population oscillations. Accordingly, the main results reported here are based on a frequency domain analysis.

3.1 Qualitative Behavior

In an unfragmented habitat (Fig. 2.a), massive waves (typical for high density populations) together with a high risk of extinction are experienced at high values of predator gain. In isolated fragmented habitats (Fig. 2.b), under the same conditions a system of short-lived dynamic transients dominates the system in each isolated fragment, leading to the extinction of the predator or the prey (and then both of them) in almost every trial and any degree of fragmentation applied. Wave phenomena are not strongly expressed in these cases, because of the lack of a stationary domain of existence and also due to the strongly limited size of the isolated fragments.

In a connected-fragmented system, however, an entirely new phenomenon is found. “Broken” waves appear that can percolate through boundaries via the openings and tend to form a new seed for newer waves (Fig. 2c.). The typical mechanism is that some prey individuals get into (“escape” into) new territories, followed by the predator somewhat later. The new territories often lack predators as a result from earlier overexploitation. Hence, a new territory can be freely repopulated by the prey, harvested by the predator later; and so on. Unlike in isolated fragments, in connected fragments the local extinction dynamics does not lead to a fatal population level consequence. New growth and new oscillations can start in neighboring fragments and spread all over in a recursive fashion. In the following, we analyze these behaviors using Fourier analysis to quantify the nature of the summarized effects.

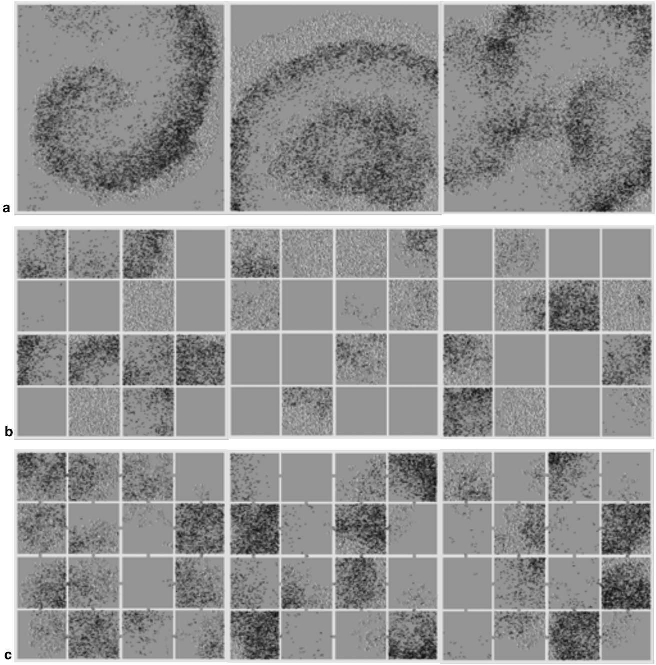


Fig. 2. Population distributions in unfragmented: a, fragmented: b; fragmented and connected: c habitats. Prey food (*grey*), prey (*white*) and predator (*black*).

3.2 Frequency Domain Analysis

Prey behavior is clearly derived from predatory effects. It will, therefore, be instructive to focus on the behavior of the population of predators.

Recurrence plots for *N=1024* steps are shown on Fig. 3. To appreciate these recurrent plots, note that a perfectly regular plot (for instance, the recurrence plot of a sinus wave) shows a regular, reticular pattern. Transients are visible as stripes and smudges. Disregarding transient irregularities, Fig. 3. shows a clear overall pattern. The upper row shows the unfragmented situation. The baseline case $Gain_{PD}$ = 10 yields a quite perfect regular recurrence pattern, showing the dominance of a single frequency and its higher harmonics. Increasing $Gain_{PD}$ (going right) leads to slower and less sharp recurrences, visualized as a decay of the original regular reticular pattern. Below the upper row, we show similar plots for a feasible fragmented connected habitat (*w2c2)*. Here, even at higher values of $Gain_{PD}$, the recurrence structure is maintained in an undamaged and sharp condition.

We also performed a (Fast) Fourier Transformation on the predator records for the various habitat types as shown in Fig. 1, using *N=4096* steps, and discarding the initial transients (*t<500*). Plots vary in vertical resolution but are identical in the horizontal axis, showing the actual frequencies (i.e., modes). At the base level of predator gain (*$Gain_{PD}$=10*), an invariant, single frequency peak characterizes the dynamics, which is independent from the number of fragments and corridors (Fig. 4, upper row). This invariant frequency arises from the interaction between the population and the spatial environment, parameterized here as a special case.

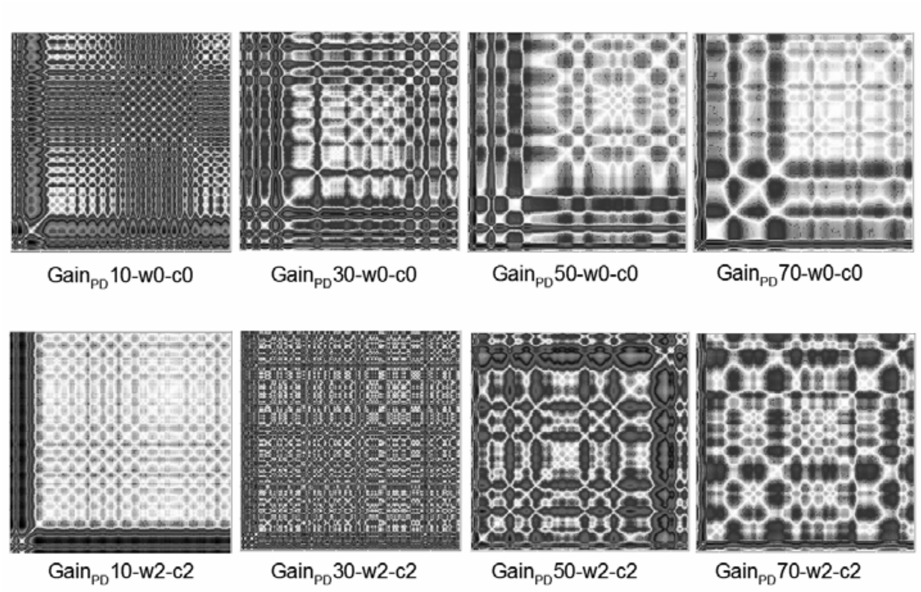


Fig. 3. Recurrence plots of various unfragmented viz. connected fragmented systems. For high quality predators (large *$Gain_{PD}$*), connected fragmented systems show sustained oscillations.

Fragmented isolated habitats show obligate extinction transients in the high predator gain region (i.e., *$Gain_{PD} \geq 50$*). Because of the universality of the extinction transients, a systematic frequency analysis of these systems would not be informative. In the unfragmented case, however, we see a different situation. We can observe that the increase of *$Gain_{PD}$* leads to a characteristic slowdown (Fig. 4., lower row) and a heavy, low-frequency tail, appearing due to various transients that arise from irregular oscillations with spatial inhomogeneity (as seen in Fig. 2.a), topped by a zero frequency component that corresponds to the extinctions (Fig.4., lower row, right); this combined tail finally suppresses the periodic signal (*$Gain_{PD}$* =70, *w3c3*).

In contrast, fragmented connected habitats show the most interesting behavior. We observe that predators respond to increasing fragmentation and to growing *$Gain_{PD}$* differently. The opening of corridors introduces a left shift and a wider base in the frequency distribution; this means that many new frequencies emerge with small amplitude but also that the waves are slowing down (Fig. 5, throughout). The overall system dynamics also depends on the degree of fragmentation via the appearance of

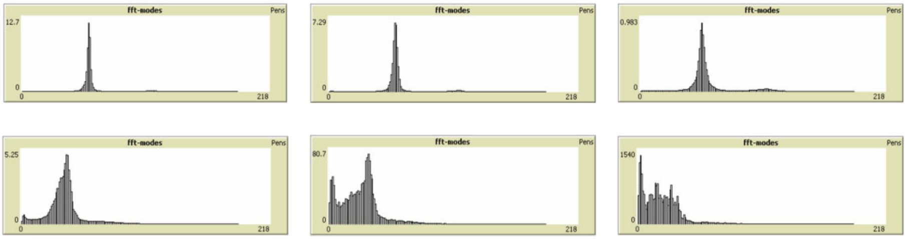


Fig. 4. Connected fragmented habitats I. Upper row: Sharp oscillations in the baseline case (Gain_{PD}=10). Left to right: *w0c0*, *w1c1*, and *w3c3*. Lower row: unfragmented (*w0c0*) case with increasing predator gain (Gain_{PD}=30, 50 and 70), showing low viz. zero frequency tail.

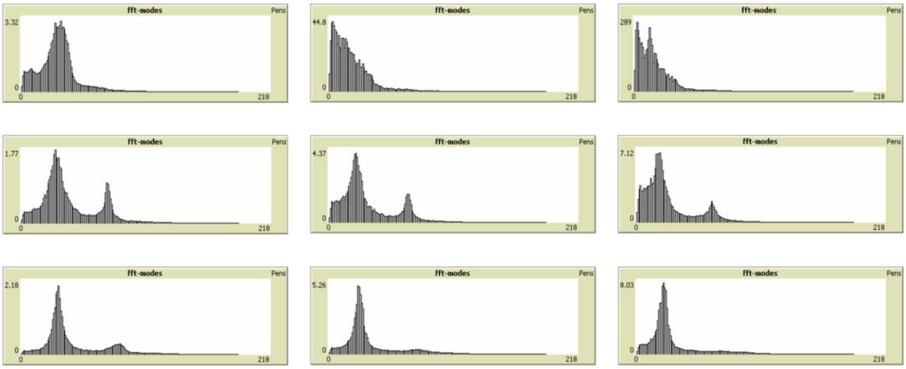


Fig. 5. Connected fragmented habitats II. Left to right: increasing predator gain ($Gain_{PD}$=30, 50 and 70). Top to bottom: increasing fragmentation: *w1c1*, *w2c2* and *w3c3*.

global harmonics that survive despite local extinctions (Fig. 5, middle and lower rows) (A notable contingency is the occurrence of higher harmonics when fragment size supports multiple oscillations, see Fig. 5, middle row). The quality of the predator ($Gain_{PD}$) also has a profound effect here. As in the unfragmented case, increasing predator quality leads to a left shift (i.e., slowdown) on the frequency plot, and the same low-frequency tail appears. This tail is again associated with the transients and extinctions in the same way as in the unfragmented case. However, this time we deal with an asynchronous process, which is the result of the independent dynamics of the several fragments, and accordingly, we see that the local transients do not destroy the main population harmonics. Indeed, in the connected-fragmented case, increasing fragmentation (Fig. 5, top to bottom) increasingly *restores* the original harmonics.

Thus, we observe that two interwoven effects arise from fragmentation and predator quality. A worst combination case is seen on Fig. 5, upper row (viz. *w1c1*), where fragmentation biases the system towards extinctions in the individual fragments (shown by the notorious low-frequency tail), but at the same time, the positive effect arising from corridors (Fig. 5, middle and lower rows) is not strong enough yet to support the survival of the lead frequency in the combined system of fragments.

4 Conclusions and Interpretation

There is a natural analogy with wave-breakers here. Wave-breakers are obstacles that disrupt otherwise fatal see or river waves, dispersing the effects of energy. Here, similarly, fragmentation barriers can break population waves – however, of course, here the end effect is the opposite on the system; exactly by breaking the otherwise fatal waves can the population escape and via the "leaking" of corridors a sufficient amount of organisms can survive, to build up new waves, to be broken up again, etc.

This mechanism explains our otherwise highly counterintuitive main finding that more fragmentation can be better than little or – what is the least expected – no fragmentation at all. The key is connectedness in fragmentation: the higher the fragmentation, the more effective the wave breakers, and via the connections, the more efficient the regeneration process of subpopulations.

References

1. Boerlijst, M.C., Lamers, M.E., Hogeweg, P.: Evolutionary consequences of spiral waves in a host-parasitoid system. Proceedings of the Royal Society of London Series B Biological Sciences 253(1336), 15–18 (1993)
2. MacArthur, R.H., Wilson, E.O.: The theory of island biogeography. Princeton University Press, Princeton (1967)
3. Levins, R.: Some demographic and genetic consequences of environmental heterogeneity for biological control. Bulletin of the Entomological Society of America 15, 237–240 (1969)
4. Hanski, I.: Metapopulation ecology. Oxford University Press, Oxford (1999)
5. Haddad, N.M., Tewksbury, J.J.: Impacts of corridors on populations and communities. In: Crooks, K., Sanjayan, M. (eds.) Connectivity Conservation, pp. 390–415. Cambridge University Press, Cambridge (2006)
6. Helbing, D.: A fluid-dynamic model for the movement of pedestrians. Complex Systems 6, 391–415 (1992)
7. Karsai, I., Kampis, G.: Connected fragmented habitats facilitate stable coexistence dynamics. Ecological Modelling 222, 447–455 (2011)

Symbiosis Enables the Evolution of Rare Complexes in Structured Environments

Rob Mills and Richard A. Watson

Natural Systems Research Group, University of Southampton, UK
rmm05r@ecs.soton.ac.uk

Abstract. We present a model that considers evolvable symbiotic associations between species, such that one species can have an influence over the likelihood of other species being present in its environment. We show that this process of 'symbiotic evolution' leads to rare and adaptively significant complexes that are unavailable via non-associative evolution.

1 Introduction

The organisation of biological systems depends strongly on the interactions between the organisms that share environments [1]. These interactions not only shape the selective forces on organisms, but in addition can be subject to change themselves. Symbiosis, the collaboration between organisms of different types [2], is very common in nature [3, 4]. Symbiotic associations have the capability of altering an organism's biotic environment: the selective context in which it will appear. This modification in environment is clear where symbiosis is taken to its logical conclusion, symbiogenesis: where the symbionts involved become reproductively inseparable [5]. This has the potential for evolutionary significance: symbiogenesis is thought to be implicated in some of the major transitions in evolution [6, 7]. There are also less extreme symbioses between free-living species that still a have significant impact on their likely biotic environments [1].

Given the ubiquity of symbiotic associations, and the assumption that such relationships can modify the likelihood of co-occurrence between species, we ask the following questions: What kind of association formation mechanism can lead to the evolution of complexes that are unevolvable in the absence of associations? Under what conditions is such a distinction available?

In this paper we describe a model framework of symbiotic evolution, where the ecosystem composition adapts rapidly according to local dynamics, symbiotic associations adapt gradually between co-occurring species in the ecosystem, and the associations in turn modify local dynamics. We investigate an implementation of this framework that idealises the separation in timescales of these two adaptive processes. We find that evolving symbiotic associations on a slower timescale than the dynamical changes in ecosystem composition is sufficient to lead to adaptively significant complexes with many dependencies resolved, and are unavailable without association formation. We use a structured adaptive landscape, where only very specific changes are sufficient to traverse the

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 110–117, 2011.

ruggedness. The challenge then for associative evolution is to evolve compatible groupings that resolve conflicts, and transfer competition to a higher level, which is sufficient to traverse this ruggedness. This provides insight into why there is a distinction in the evolutionary outcomes with and without association formation. In other work in this volume [8], we show conditions under which individual selection leads to the reinforcement of associations between individuals of different species that co-occur in the ecosystem dynamics. In the present paper, we use a higher-level model where associations evolve between species according to their co-occurrence. This allows us to examine the adaptive significance of the coupled processes, when compared to non-associative evolution.

Prior models have been suggested that investigate the evolution of symbiogenic encapsulation [10–12], or abstractions thereof [13]. These models use a variety of mechanisms to determine the suitability of symbiogenic joins, including Pareto dominance [10], context-optimality [13], and maximising reciprocal synergy [12]. In [14] the evolution of 'observers' provides groupings to coarse-grain an adaptive landscape. A related approach is applied to physical models in [15]. As noted, we use different temporal scales for ecosystem dynamics and association formation. In evolutionary computation, [16] uses results of multiple hill climbing runs to build a model of dependencies, which provides a similar timescale separation to successfully solve hierarchical problems. Memetic algorithms [17] also use search at two levels, but importantly, neither search process modifies the variational units for the other.

2 Modelling Symbiotic Variation

In our model ecosystem there are many species. Since we want to investigate the effect of evolving the interactions *between* species, the species themselves have a trivial representation. For each of N niches in the environment, two particular species compete to inhabit it. There are $2N$ species in total. Each species also has a set of association strengths, one for each of the other $2N-1$ species. These associations can evolve, whereas their niche is immutable.

Initially the ecosystem has a random constitution of N species, such that all niches are occupied. All association strengths are initialised at zero. Changes in the ecosystem composition occur by following the system dynamics – by introducing random migrants that compete with the current ecosystem occupants. The symbiotic associations strengthen between species that are present in the ecosystem at any given timestep. Within a reasonably short timescale, no further changes in ecosystem composition occur: the ecosystem is at a locally stable state. Periodically, the ecosystem is perturbed such that its constitution is randomised, and local dynamics will again cause ecosystem changes. Note that these perturbations do not modify the symbiotic associations that have evolved.

The changes in the ecosystem composition are effected by allowing a random species to immigrate, and if the overall utility of the ecosystem is higher with this migrant than without, it is retained. When the migrant species has non-zero associations with other species, these are interpreted as probabilities that those

Table 1. Symbiosis Model Main Procedure

1. allow d demes to run to their local attractor
2. measure the co-occurrence of each pair of species within all deme attractors, and reinforce symbiotic associations according to Eqn. 1.
3. randomise each ecosystem composition and go to step 1.

other species will migrate at once. Thus, if a pair of species has symbiotic associations of maximal strength, they will always migrate together. These associations have the effect of correlating the possible variation in ecosystem composition.

We use an external fitness function to define the overall utility of each ecosystem composition, and this is used in a 'black box' manner: a utility value is only defined if all niches are occupied. However, by comparing two compositions that only differ by the occupant of a single niche, we identify which of the alternatives is more suited to the current context, thus effecting individual-level selection. The case is similar when a migrant group modifies multiple niches. Therefore, this reveals the context-dependent utility contribution of a migrant group.

In order to examine the behaviour we instantiate a model within this framework, with two additional assumptions: i) the length of time the system spends at local attractors dominates the time spent in transients; ii) instead of modifying the associations gradually over several trajectories, we use several independent demes in parallel, and modify the associations in proportion to the co-occurrence of species across the ensemble of demes. Using only attractor states to inform association evolution makes explicit the separation of timescales between changes to association and ecological changes.

2.1 Model M-S: Continuous Associations, Parallelised

If the ecosystem spends most of its time at attractors (or close to), a suitable approximation is to only modify the associations according to the species that are present at attractors. Procedurally, this model is described in Table 1. Following the dynamics of the ecosystem is a simple process that takes into account the symbiotic associations. The procedure is described in Table 2.

Observed co-occurrence, $O_{i,j}$, is calculated from the proportion of demes where both species i and j are present. The expected co-occurrence frequency is calculated from the product of univariate frequencies: $E_{i,j} = X_i X_j$. Using these values we construct a deviance from expected metric (essentially, a measure of surprise with respect to $E_{i,j}$). Associations are formed according to the rule in Eqn 1, where h is the threshold below which all $S_{i,j}$ values are set to zero.

$$S_{i,j} = \begin{cases} \frac{O_{i,j} - E_{i,j}}{\min(X_i, X_j) - E_{i,j}}, & \text{if } h \cdot \frac{O_{i,j} - E_{i,j}}{\min(X_i, X_j) - E_{i,j}} < O_{i,j} \leq \min(X_i, X_j) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The exit condition in step 6) when following dynamics can either be to make a pre-specified number of migrations, or alternatively to wait for P trial migrations

Table 2. Symbiosis-Informed Ecosystem Dynamics

```
1. Evaluate the initial ecosystem composition ($\rightarrow f_p$)
2. Form a migrant group $g$:
    (a) Randomly select a migrant species $m$, and add to $g$
    (b) Select without replacement a random species $x \neq m$
    (c) With probability $S_{m,x}$, add $x$ to $g$, unless that niche is already filled in $g$
    (d) If any species has not been sampled, goto (b)
3. Temporarily introduce $g$ to the ecosystem, allowing $g$ to take precedence over
   the current occupants
4. Evaluate the modified ecosystem composition ($\rightarrow f_m$)
5. If $f_m \geq f_p$, allow the migrant group $g$ to remain permanently, and set $f_p \leftarrow f_m$
6. If exit conditions not met, go to step 2).
```

without a change in composition. Note that in step 5), if the overall utility of the modified ecosystem is not higher than the original ecosystem, *all* species changes are reverted. Implications of this are discussed further in Sec. 4.

3 Simulation Experiments

We investigate the behaviour of the described models, and in particular, the complexes that evolve. We measure the evolutionary timescales required for each model to find the globally optimal configuration in the entire landscape (examined over several system sizes). The control model (M-C) is equivalent to M-S except that no associations are evolved – the S matrix is held at 0 throughout.

Watson and Jansen [18] introduce a synthetic problem class where instances comprise several large modules of binary variables, or ‘building blocks’. Each module has two optima, one of higher utility than the other, both with equal sized basins. These are concatenated with no inter-module dependencies to construct the full problem. Eqn. 2 defines the utility contribution of a single module.

$$f(x) = \begin{cases} k & \text{if } U(x) = k \\ \frac{U(x)}{2} & \text{if } k > U(x) > \frac{k}{2} \\ \frac{(k - U(x))}{2} & \text{else} \end{cases} \tag{2}$$

Given that each of Z modules has k variables, x is a configuration of variables within that module, and $U(x)$ is the unitation (number of variables set to ‘1’). We set $k = Z = \sqrt{N}$, such that the size of the modules scales with the size of the system, and thus refer to this problem as the scalable building blocks (SBB) problem. Each variable in the problem corresponds to a niche, and we refer to the two species that can occupy this niche as the ‘0-species’ and ‘1-species’ – giving a total of $2N$ different species in the ecosystem. The resultant landscape is very rugged, with 2^Z locally optimal configurations. Of these configurations, only one is globally optimal: when all niches are occupied by the 1-species. From any local optimum, the nearest configuration of higher utility differs in k niches – all within one module that is currently entirely occupied with 0-species.

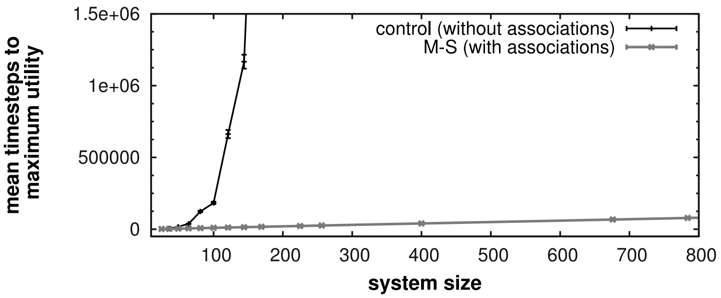


Fig. 1. Timesteps that each model takes before visiting the highest utility configuration in the SBB landscape. The symbiotic model visits this configuration with ease, whereas the control model very rarely does (thus requiring exponentially many epochs to sample the appropriate basin). The parameter settings are as follows: for M-S, we set $d = 50$, $h = 0.6$; In both M-S and the control M-C, following the dynamics uses $2N$ migrations, sampling on the initially selected species m without replacement; each model continues to restart until the globally optimal configuration is sampled. 30 repeats are performed for M-S, 100 repeats are performed for M-C (on account of higher stochasticity).

4 Discussion

As Fig. 1 shows, the symbiosis model provides an efficient processes that finds the very particular configuration with highest overall utility. This is in contrast to the control model, where the number of timesteps required to find the same configuration increases exponentially with the system size. The comparison indicates the significance of the result that M-S can find such a configuration.

The symbiotic model is very efficient at discovering the globally optimal configuration in this landscape. How is this so? First, consider the attractors visited by the initial set of demes. In each one, some modules will be occupied by all-0, and some by all-1 species – but no single attractor will have all-1 in all modules. Whenever a 1-species occurs, it always co-occurs with other 1-species in the other niches *in that module.* Note that it is not the case that 1-species always occur in any particular module: the all 0-species attractor also has a 50% basin. Furthermore, there is no correlation between the attractor that each module finds, since the landscape is separable. Associations are formed between species that co-occur frequently across the set of demes. Thus, within any module strong symbioses will form amongst all k of the 1-species, and likewise amongst all k of the 0-species. No associations will form between 0-species and 1-species that occupy niches in the same module, since at attractors there are no co-occurrences. The between-module co-occurrences are predicted by the univariate frequencies (*i.e.,* there is little or no surprise), so no associations will form here either. Fig. 2 (a) shows the resultant associations from the described process.

Now let us consider the local dynamics with these associations. Each migration is likely to introduce all of the compatible species in a particular module. This transfers competition to the module-level. To start with, introducing either an all-0 group or an all-1 group will remain, since both are significantly higher utility

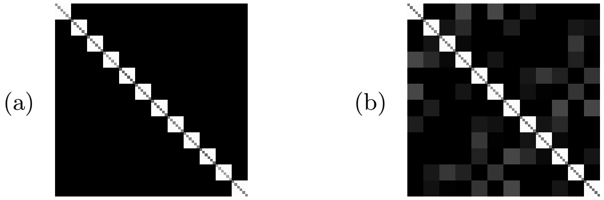


Fig. 2. Symbiosis matrices for a 72 species (N=36, k=6) ecosystem. (a) Ideal target to represent the landscape structure. (b) Calculated from the surprise metric (Eqn. 1) on an example set of local optima, without a threshold (*i.e.,* h=0). This leads to a largely accurate representation, with some spurious interactions. An appropriate threshold can recover the target – see text. Lighter shades indicate stronger associations.

than a random composition in the corresponding niches. However, as soon as an all-1 group is introduced, it has exactly the right variation to move to the highest utility possible in that module, and will thus replace any other composition. When appropriate all-1 groups have migrated into each of the modules, the overall utility will be maximised. After the associations develop as described above, this reliably occurs.

While each of the configurations that is discovered in one of the demes has N species present, the subsequent competition is not directly between these local optima. The associations that evolve form groups of k members, corresponding to each module, as described. This is important for a significant result. Because the module sub-functions are independent, these small groups are both sufficient and selectively efficient in the sense that they create Z independent competitions between the two sub-solutions in each module, rather than a single competition between all 2^Z possible local optima [18].

The success of the symbiosis model depends on the formation of associations appropriate to construct these per-module competitions. Fig. 2 (a) shows the target associations, which comprises strong associations between all compatible species within each module, and between-module associations. Frame (b) shows an example calculation of the surprise metric, without a threshold applied (Eqn. 1). In addition to the correct within-module interactions, this measure indicates some spurious interactions. Using Eqn. 1 with $h = 0.6$ is sufficient to recover just the appropriate associations, as in frame (a).

There are several control models that we could have used, but would any have a better chance than M-C? Selecting on groups at the level of entire ecosystems does not allow one ecosystem configuration to have any correlation with the next. Without the ability to follow fitness gradients, all possible ecosystems must be enumerated. In principle, migrations of uncorrelated groups of species can move between the local optima that defeat single-migrant dynamics. However, to move between local optima in the SBB landscape, the k species that are in one module must all change at once. The number of different possible k-species groups is $2^k\binom{N}{k}$, so any uncorrelated group formation process will require a prohibitively large number of attempts before finding the exact group necessary to move from one local optimum to another of higher utility. Even if the decomposition were somehow known, randomly forming a group of the particular membership required is still exponential in the module size, which scales with the system size. Considering all 2^Z local attractors as M-C does is therefore the fastest control when $k = Z$, despite it not requiring any structural information. Therefore, any reasonable control that does not evolve symbiotic associations will not reliably find ecosystems of the same level of utility.

As described above, we use a mechanism that reinforces associations at the inter-species level – based on the surprise in the co-occurrence of species across several demes, with respect to the expected frequencies predicted from individual occurrence levels. The proposed mechanism is simpler than those suggested in previous models [10–12], and gives rise to qualitatively distinct results. In other work, we use a model that has an explicit population within each species, and each individual can evolve species-specific symbiotic associations (*i.e.,* the associations are individual traits) [8]. We use this model to explore the types of population structure that lead to the evolution of adaptively significant associations. In particular, we show that the high-level mechanism used in M-S need not be imposed at the species level, but can in fact can be manifested via the evolution of individual traits. As in the present model, the evolved associations create higher-level groupings and selective units.

Note that due to the transparent simplicity of the SBB landscape, it is actually possible to simplify M-S to only use 'all-or-nothing' associations in this case. However, such a simplification narrows the applicability. Elsewhere we have investigated landscapes in which associations with intermediate strengths find high-utility configurations that cannot be found with 'all-or-nothing' associations [19, 20]. It is worth noting that the algorithm proposed in [16] is able to efficiently solve hierarchical problems with some similar abstractions, in particular by model building from information at local optima.

In our model, we apply selection on migration groups such that an entire group is rejected if overall utility is not improved. This effectively causes the units of variation to be synonymous with the units of selection. An alternative scheme might allow groups to migrate together, but select on individual species. We suggest that because the individual selection is performed in the new context, with the entire migration group, the ultimate changes in ecosystem composition will not be significantly different than if selecting on groups as an entire unit. Recall that migration groups typically comprise species that were frequently found to co-occur in locally stable contexts. Thus, the individual species would be selected in the context of the particular group. We leave the verification that both schemes have qualitatively equivalent results for future work.

We have presented a model of the evolution of symbiotic associations where a separation of the temporal scales of changes in ecosystem composition and changes in species associations leads to the evolution adaptively significant complexes. This is in contrast to previous models of symbiosis with similar timescales for both levels of adaptation, and results in a simpler and more biologically plausible model. Provided that the ecosystem is perturbed at a low frequency such that the transients are shorter than the average time between perturbations, the associations that form give rise to specific species groupings that can traverse rugged landscapes.

Acknowledgements. Thanks to Jason Noble, Simon Powers, Johannes van der Horst, and Devin Drown for useful discussions.

References

1. Thompson, J.N.: The Coevolutionary Process. Chicago (1994)
2. Mayr, E.: What Evolution is. Phoenix, London (2001)
3. Dunbar, H.E., Wilson, A.C.C., Ferguson, N.R., Moran, N.A.: Aphid thermal tolerance is governed by a point mutation in bacterial symbionts. PLoS Biology 5(5), e96 (2007)
4. Margulis, L.: The Symbiotic Planet. Phoenix, London (1998)
5. Khakhina, L.N.: Concepts of Symbiogenesis. Yale University Press, New Haven (1992)
6. Maynard Smith, J., Szathmáry, E.: The major transitions in evolution. Oxford University Press, Oxford (1995)
7. Margulis, L., Dolan, M.F., Guerrero, R.: Origin of the nucleus from the karyomastigontin amitochondriate protists. PNAS 97(13), 6954–6959 (2000)
8. Watson, R.A., Palmius, N., Mills, R., Powers, S., Penn, A.: Can selfish symbioses effect higher-level selection? In: ECAL (2009)
9. Mills, R., Watson, R.A.: Variable discrimination of crossover versus mutation using parameterized modular structure. In: GECCO, pp. 1312–1319 (2007)
10. Watson, R.A., Pollack, J.B.: A computational model of symbiotic composition in evolutionary transitions. Biosystems 69(2-3), 187–209 (2003)
11. Defaweux, A., Lenaerts, T., van Hemert, J.I.: Evolutionary transitions as a metaphor for evolutionary optimisation. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 342–352. Springer, Heidelberg (2005)
12. Mills, R., Watson, R.A.: Symbiosis, synergy and modularity: Introducing the reciprocal synergy symbiosis algorithm. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 1192–1201. Springer, Heidelberg (2007)
13. de Jong, E.D., Watson, R.A., Thierens, D.: On the complexity of hierarchical problem solving. In: GECCO, pp. 1201–1208 (2005)
14. Philemotte, C., Bersini, H.: A gestalt genetic algorithm: less details for better search. In: Procs. GECCO, pp. 1328–1334 (2007)
15. Houdayer, J., Martin, O.C.: Renormalization for discrete optimization. Physical Review Letters 83, 1030–1033 (1999)
16. Iclanzan, D., Dumitrescu, D.: Overcoming hierarchical difficulty by hill-climbing the building block structure. In: GECCO, pp. 1256–1263 (2007)
17. Krasnogor, N., Smith, J.: A tutorial for competent memetic algorithms. IEEE Transactions on Evolutionary Computation 9(5), 474–488 (2005)
18. Watson, R.A., Jansen, T.: A building-block royal road where crossover is provably essential. In: GECCO, pp. 1452–1459 (2007)
19. Mills, R., Watson, R.A.: Adaptive units of selection can evolve complexes that are provably unevolvable under fixed units of selection (abstract). In: ALIFE XI, p. 785 (2008)
20. Mills, R., Watson, R.A.: Dynamic problem decomposition via evolved symbiotic associations (in preparation)

Growth of Structured Artificial Neural Networks by Virtual Embryogenesis*

Ronald Thenius, Michael Bodi, Thomas Schmickl, and Karl Crailsheim

Artificial Life Laboratory of the Department of Zoology
Karl-Franzens University Graz
Universitätsplatz 2, A-8010 Graz, Austria
ronald.thenius@uni-graz.at

Abstract. In the work at hand, a bio-inspired approach to robot controller evolution is described. By using concepts found in biological embryogenesis we developed a system of virtual embryogenesis, that can be used to shape artificial neural networks. The described virtual embryogenesis has the ability to structure a network, regarding the number of nodes, the degree of connectivity between the nodes and the amount and structure of sub-networks. Furthermore, it allows the development of inhomogeneous neural networks by cellular differentiation processes by the evolution predispositions of cells to different learning-paradigms or functionalities. The main goal of the described method is the evolution of a logical structure (e.g., artificial neural networks), that is able to control an artificial agent (e.g., robot). The method of developing, extracting and consolidation of an neural network from a virtual embryo is described. The work at hand demonstrates the ability of the described system to produce functional neural patterns, even after mutations have taken place in the genome.

1 Introduction

Structured artificial neural networks (ANNs) are advantageous in many ways: For example, they provide highly interesting auto-teaching structures as mentioned by Nolfi in [1] and [2]. In these works an ANN is described, which consists of two subnets, a "teaching net" and a "controlling net". If such a network is shaped by an artificial evolutionary process it has "genetically inherited predispositions to learn", as described by the authors. The ability of an ANN to evolve such substructures is one of the main goals of the virtual embryogenetic approach described in this work.

Other concepts, like the influence of body (and neural controller shape) onto the function of the controller are described in [3] and [4]. The main ideas of these publications are to describe the significant influence of the morphological shape of the agent (or robot) on the learning and controlling process. As described later

* Supported by: EU-IST FET project I-Swarm, no. 507006; EU-IST-FET project SYMBRION, no. 216342; EU-ICT project REPLICATOR, no. 216240.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 118–125, 2011.

in this article, the positions of sensor input nodes and actuator output nodes within the virtual embryo's growing body corresponds the positions of sensors and actuators in the real robot ("embodiment" of controller).

Other interesting approaches to the challenge of shaping of ANNs (with focus on the French flag problem described in [5]) are described in [6]: The function of a node is not determined there, but can be shaped by the genome. Another work, that deals with the problem of differing functions within an ANN, is [7], in which different predispositions for learning are implemented by "virtual adaptive neuro-endocrinology". Within such a network, different types of cells exist: gland-cells, which influence the learning of the network, and regular cells, which are influenced by the gland-cells.

In [8] the process of virtual embryogenesis is described (see figure 1). It also describes how the feedback between morphogenes and the interaction between morphogenes and the genome act together and define the final shape of the embryo. Based on this work we show how to extract, how to optimize and how to use an ANN from a grown virtual embryo. In addition we discuss the advantages of such networks for future evolutionary adaptation.

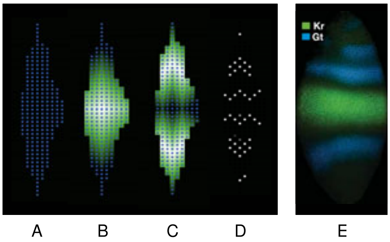


Fig. 1. Comparison of virtual embryogenesis in our model and real-world embryogenesis (from [8]): A: Virtual embryo, consisting of cells (dots); B: Morphogene gradient in the embryo; C: Gradient of another morphogene, inducing cell differentiation. D: Embryo consisting of differentiated cells (white dots) and non-differentiated cells (invisible) ; E: Natural example of gene expression: Activity domains of gap genes in larva (lateral view) of *Drosophila m.* (from [9]; 'Kr' and 'Gt' indicate gene activity.)

2 Method

2.1 Concept of Adapting Virtual Embryogenesis for Controller Development

In the process of virtual embryogenesis, described in [8], an embryo consists of cells, which develop to nodes in the ANN during the embryological process. These cells duplicate, specialise, emit chemical substances (morphogenes), die, or build links to other cells depending local status variables. Links between cells represent neural connections between the nodes of the resulting ANN. Cells can be "pushed around" in space, due to growth processes of the embryo, but they have no ability for active movement. Cellular actions are coded in the genome (collection of single genes) of the embryo. Genes are triggered by presence of virtual morphogenes. One possible effect of gene-activation is the production of another morphogene. This way a network of feedbacks emerges, which leads to a self-organised process, that governs the growth of the embryo. The developed network is analysed and translated into a data structure which is compatible to a standard ANN-interpreter.

The overall process of growing and extracting an ANN from an embryo is a procedure of 9 distinct steps (figure 2): A genome (figure 2a) that is able to perform complex operations including operations regarding the interpretation of the genome (figure 2b) codes for the growth process of a virtual embryo (figure 2c). For details of this process see [8]. Some cells of this virtual embryo are able to develop links to other cells in the embryo and thus become neural cells. (figure 2d). After the embryogenetic process, the ANN is extracted from the virtual embryo (figure 2e). "Dead-end" connections within the network that might develop during embryological process are removed for the sake of calculation speed (figure 2f). The ANN is then linked to sensor- nodes and actuator nodes according to the definitions of the robot in which the network has to perform (figure 2g). After this step, the network is translated into a data structure, that can be parsed by an ANN interpreter (figure 2h) and tested in the robot or in the simulation environment (figure 2i).

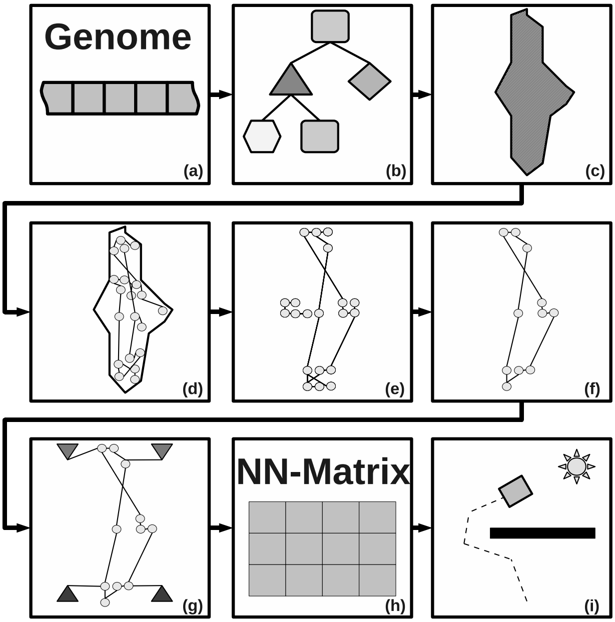


Fig. 2. Concept of shaping ANNs by virtual embryogenesis

3 Results

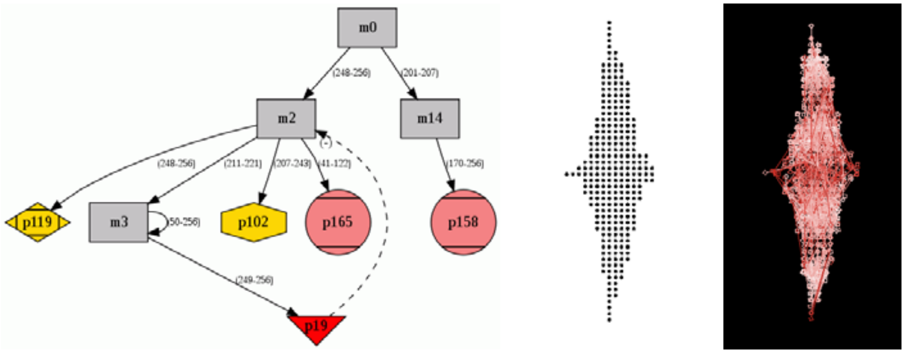
3.1 Growth of Neurons

One of the most important steps on the way from the initial genome to a robot controller is the definition and linkage of neural cells within the virtual embryo. Some genes trigger the growth of neural links of a cell. In figure 3(c) a simple multi-layered neural structure is shown. Due to locally differing concentrations of morphogenes it is possible that sub-networks grow within different parts of the embryo. Due to the possibility to modify the internal status variables of neural cells (as mentioned in [8]), these sub-networks have different properties, for example, in their pre-disposition for learning or in their function. The possibility to structure neural networks this way in a genome controlled self-organised process is one of the main advantages of the method described here. Such processes can be modified by artificial evolution.

The genetic structure shown in figure 3(a) contains two genes ("p165" and "p185"), which code for proteins that lead to a linkage of neural cells. The protein, coded by the gene "p165", is part of the genetic substructure (beginning with the morphogene coding gene "m2") that controls the vertical growth of the embryo (final shape of the embryo depicted in figure 3(b)). Due to its position, this gene induces the building of "long distance connections" all over the embryo (depicted by red lines in figure 3(c)). This process already takes place very early in the growth process of the virtual embryo, so that linked cells move into different directions and the neural connections reach throughout the whole embryo. The protein resulting from the triggering of the gene "p185" leads to the development of the spatially concentrated sub-networks. The sub-networks with high density are depicted by white lines in figure 3(c).

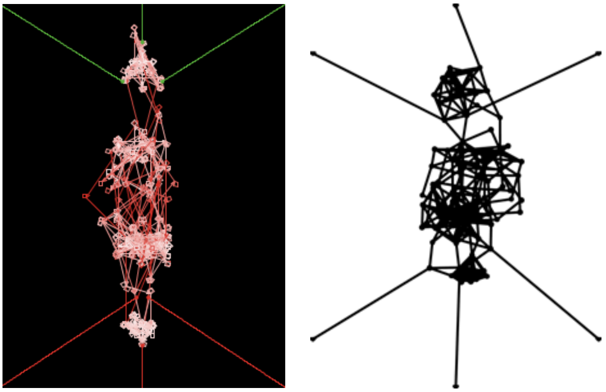
3.2 Translation

After the virtual embryo has stopped to grow, it is necessary to extract the ANN from the embryo and to translate it into a structure that is readable for a standard neural network interpreter. Before that, the network gets consolidated: All neural cells that have no input or output are removed from the network . This "consolidation process" is an optimisation step which prevents unnecessary structures (like "dead end" connections). In a second step, the ANN has to be linked to the inputs (sensors) and outputs (actuators) of the robot (or other agent), controlled by the network (figure 4). The cells that get connected to inputs and outputs are selected by their position in the virtual embryo corresponding to the position of sensors and actuators on the robot. In a third step the finalised neural network is translated into a structure of one-dimensional and two-dimensional arrays that can be parsed by a standard neural network interpreter. After this step the embryogenetically shaped ANN is ready for upload to the robot.



(a) Genome (b) Embryo (c) Neural cells

Fig. 3. Growth of an ANN in a virtual embryo. (a): The genome defining the number and location of neural sub-networks in the embryo via self-organizing processes (for details please see [8]). (b): Shape of the embryo, neural connections are not drawn. (c): Screenshot of the neural network in the embryo. White lines indicate "short distance connections" within local sub-networks, red lines indicate "long distance connections" that reach throughout the whole embryo and connect the sub-networks with each other.



(a) Network inside embyro (b) Unflolded network

Fig. 4. Extracting and optimising the ANN. (a): A neural network inside of a virtual embryo. (b): Unfolded neural network. During virtual embryogenesis "dead-end" connections can occur, which consist of neural cells that do not have input and output. These cells are disadvantageous to the efficiency of the resulting ANN. Yellow circles indicate neural cells, lines indicate connections between cells.

4 Discussion

4.1 Usability of Virtual Embryogenesis for Shaping of ANNs and Robot Controllers

One important ability of the described system, on which an artificial evolutionary process works, is the stability against lethal mutations, along with the ability to forward changes from the genetic level to the phenotypic (morphological) level. As shown in figure 5, different embryo shapes emerge from one “ancestor” by applying small changes to the genome. One important feature is, that these new shapes do not look completely different from its ancestor, but differ only slightly (“self-similarity’). These changes do not only take place on the level of shape, but also on the level of resulting ANNs. As shown in figure 6, changes in the genome lead also to changes in the structure of the resulting network (i.e., the robot controller). This way the number of layers, the number of cells within a layer, the micro-structure of a layer, etc. can change. Please notice, that this ability of the described system leads to an increased number of “viable ” offspring of an given “ancestor” in an evolutionary system . We expect this self-similarity to allow for exploration of larger areas of the fitness landscape in less time, due to higher possible mutation rates, and better variations of the offspring due to mutation.

Using the described technique of virtual embryogenesis enables us to develop embryos from genomes (see figure 1). Within these embryos we are able to let grow ANNs, whereby the shape, the degree of connectedness and differentiation processes within the ANN are determined during the developmental process. This growth process is controlled by the genome and uses a system of feedbacks and delays. Especially the differentiation of cells within the embryo (see figure 1 D) is important. It allows us to evolve complex structures, constructed out of sub-structures: many local networks within one bigger network. The described system allows not only evolution of morphological structures: In addition internal functions or predispositions of cells or groups of cells get optimized. Within a neural network different kinds of learning methods could applied (like mentioned in [1] and [2]). In addition different subnets within one neural network could learn with different speeds, so they are able to adapt to processes that take place in different time scales in the environment of the agent. It is possible to structure an ANN into different sub-networks, that are able to specialise for different functionalities by being receptive to different external reward functions, that influence the learning function. Also different functions within structured ANNs sub-networks can be determined, for example sensor fusion or task-management.

5 Conclusion

Our method of evolving ANNs by simulating embryogenesis elaborates on concepts from the field of evo-devo [10]. This approach produces results that are comparable to the products of natural developmental processes (figure 1). Within an embryo we are able to grow ANNs, consisting of different sub-networks. If

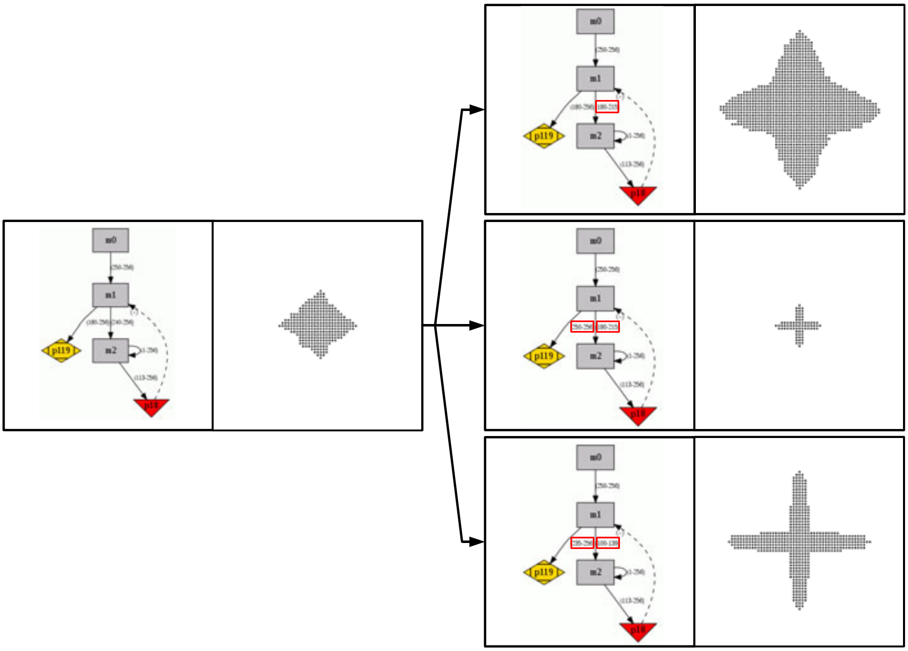


Fig. 5. Variations in the genome lead to variations in shape: If the genome of one "ancestor" (left sub-figure) is changed slightly, resulting embryos differ slightly in shape (right sub-figures). Changes in the genome are marked with red boxes.

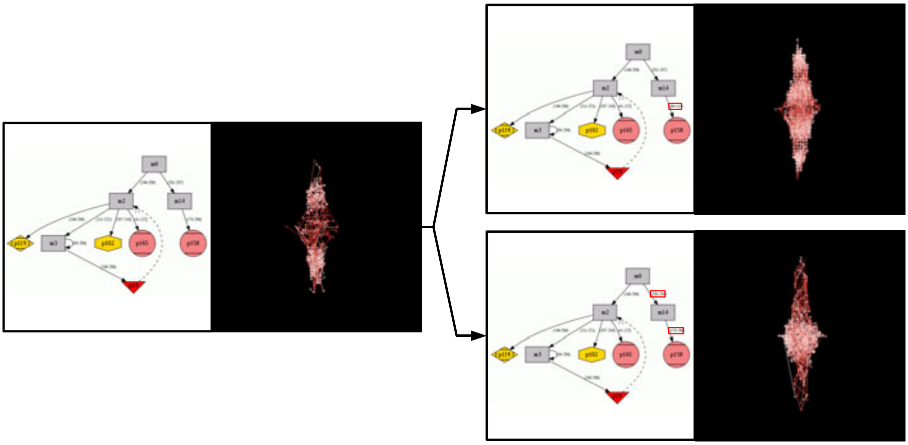


Fig. 6. Variations in the genome lead to variations in the shape of the resulting ANN: If the genome of on "ancestor" (left sub-figure, network with 2 layers) is changed, the resulting networks also change (right sub-figures, networks with 1, respectively 4 layers)

the genome of a virtual embryo is mutated, the described system leads to functional intact offspring, slightly differing from its "ancestor" (figure 6 and 5). We also show the possibility how to differentiate groups of cells (figure 1D), what is comparable to tissue development in biology.

In the next step we plan to combine the presented virtual embryogenesis with artificial evolution. The quality of the neural network (e.g., regarding learning abilities, learning speed or ability to adapt to new situations) will be used for a fitness function. This way novel and efficient ANN-structures are planned to be evolved. Due to the possibility of the described system to develop structured ANNs, we plan to investigate the processes of evolution of hierarchical brain-structures, and this way to learn more about the biological evolution of brains in real-world animals. Further, we expect this technique to be excellent for shaping robot controllers, due to the fact that the evolved ANNs can be as variable in their structure and function as the virtual embryos, in which they develop.

References

1. Nolfi, S., Parisi, D.: Auto-teaching: Networks that develop their own teaching input. In: Deneubourg, J.L., Bersini, H., Goss, S., Nicolis, G., Dagonnier, R. (eds.) Proceedings of the Second European Conference on Artificial Life (1993)
2. Nolfi, S., Parisi, D.: Learning to adapt to changing environments in evolving neural networks. Adaptive Behavior 5(1), 75–98 (1997)
3. Pfeifer, R., Bongard, J.C.: How the body shapes the way we think: A new view of intelligence (Bradford Books). The MIT Press, Cambridge (2006)
4. Pfeifer, R., Iida, F., Bongard, J.: New robotics: Design principles for intelligent systems. Artificial Life 11(1-2), 99–120 (2005)
5. Wolpert, L.: Principles of development. Oxford University Press, Oxford (1998)
6. Miller, J.F., Banzhaf, W.: Evolving the program for a cell: From french flags to boolean circuits. In: On Growth, Form and Computers, pp. 278–302. Academic Press, London (2003)
7. Timmis, J., Neal, M., Thorniley, J.: An adaptive neuro-endocrine system for robotic systems. In: IEEE Workshop on Robotic Intelligence in Informationally Structured Space. Part of IEEE Workshops on Computational Intelligence, Nashville (2009) (in press)
8. Thenius, R., Schmickl, T., Crailsheim, K.: Novel concept of modelling embryology for structuring an artificial neural network. In: Troch, I., Breitenecker, F. (eds.) Proceedings of the MATHMOD (2009)
9. Jaeger, J., Surkova, S., Blagov, M., Janssens, H., Kosman, D., Kozlov, K.N., Manu, Myasnikova, E., Vanario-Alonso, C.E., Samsonova, M., Sharp, D.H., Reinitz, J.: Dynamic control of positional information in the early *Drosophila* embryo. Nature 430, 368–371 (2004)
10. Müller, G.B.: Evo-devo: Extending the evolutionary synthesis. Nature Reviews Genetics 8, 943–949 (2007)

The Microbial Genetic Algorithm

Inman Harvey

Evolutionary and Adaptive Systems Group, Centre for Computational Neuroscience and Robotics, Department of Informatics, University of Sussex, Brighton BN1 9QH, UK
inmanh@sussex.ac.uk

Abstract. We analyse how the conventional Genetic Algorithm can be stripped down and reduced to its basics. We present a minimal, modified version that can be interpreted in terms of horizontal gene transfer, as in bacterial conjugation. Whilst its functionality is effectively similar to the conventional version, it is much easier to program, and recommended for both teaching purposes and practical applications. Despite the simplicity of the core code, it effects Selection, (variable rates of) Recombination, Mutation, Elitism ('for free') and Geographical Distribution.

Keywords: Genetic Algorithm, Evolutionary Computation, Bacterial Conjugation.

1 Introduction

Darwinian evolution assumes a population of replicating individuals, with three properties: Heredity, Variation, and Selection. If we take it that the population size neither shrinks to zero, nor shoots off to infinity -- and in artificial evolution such as a genetic algorithm we typically keep the population size constant -- then individuals will come and go, whilst the makeup of the population changes over time.

Heredity means that new individuals are similar to the old individuals, typically through the offspring inheriting properties from their parents. Variation means that new individuals will not be completely identical to those that they replace. Selection implies some element of direction or discrimination in the choice of which new individuals replace which old ones. In the absence of selection, the population will still change through random genetic drift, but Darwinian evolution is typically directed: in the natural world through the natural differential survival and mating capacities of varied individuals in the environment, in artificial evolution through the fitness function that reflects the desired goal of the program designer.

Any system that meets the three basic requirements of Heredity, Variation and Selection will result in evolution. Our aim here is to work out the most minimal framework in which that can be achieved, for two reasons: firstly theoretical, to see what insights can be gained when the evolutionary method is stripped down to its bare bones; secondly didactic, to demystify the Genetic Algorithm (GA) and present a version that is trivially easy to implement.

2 GAs Stripped to the Minimum

The most creative and challenging parts of programming a GA are usually the problem-specific aspects. What is the problem space, how can one best design the

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 126–133, 2011.

genotypic expression of potential solutions or phenotypes, and the genotype-phenotype mapping, how should one design an appropriate fitness function to achieve ones goals?

Though this may be the most interesting part, it is outside the remit of our focus here on those evolutionary mechanisms that can be generic and largely independent of the specific nature of any problem. So we will assume that the programmer has found some suitable form of genotype expression, such that the genetic material of a population can be maintained and updated as the GA progresses; for the purposes of illustration we shall assume that there is a population size P of binary genotypes length N. We assume that all the hard problem-specific work of translating any specific genotype in the population into the corresponding phenotype, and evaluating its fitness, has been done for us already.

Given these premises, we wish to examine the remaining GA framework to see how it can be minimized.

2.1 Generational and Steady State GAs

Traditionally GAs were first presented in generational form. Given the current population of size P, and their evaluated fitnesses, a complete new generation of the same size was created via some selective and reproductive process, to replace the former generation in one fell swoop. A GA run starts with an initialized population, and then loops through this process for many successive generations. This would roughly correspond to some natural species that has just one breeding season, say once a year, and after breeding the parents die out without a second chance.

There are many natural species that do not have such constraints, with birth and death events happening asynchronously across the population, rather than to the beat of some rhythm. Hence the Steady State GA, which in its simplest form has as its basic event the replacement of just one individual from P by a single new one (a variant version that we shall not consider further would replace two old by two new). The repetition of this event P times is broadly equivalent to a single generation of the Generational GA, with some subtle differences.

In the Generational version, no individual survives to the next generation, although some will pass on genetic material. In the Steady State version, any one individual may, through luck or through being favoured by the selective process, survive unchanged for many generation-equivalents. Others, of course may disappear much earlier; the average lifetime of each individual will be P birth/death events, as in the generational case, but there will be more variance in these lifetimes.

There are pragmatic considerations for the programmer. The core piece of code in the Steady State version, will be smaller, although looped through P times more, than in the Generational GA. The Generational version needs to be coded so that, in effect, P birth/death events occur in parallel, although on a sequential machine this will have to be emulated by storing a succession of consequences in a temporary array; typically two arrays will be needed for this-generation and next-generation. If the programmer does have access to a cluster of processors working in parallel, then a Steady State version can farm out the evaluation of each individual (usually the computational bottleneck) to a different processor. The communication between processors can be very minimal, and the asynchronous nature of the Steady State version means that it will not even be necessary to worry too much about keeping different processors in step.

These are suggestive reasons, but perhaps rather weak, subjective and indeed even aesthetic, for favouring a Steady State version of a GA. A stronger reason for doing so in a minimalist GA is that it can exploit a very simple implementation of selection.

2.2 Selection

Traditionally but regrettably, newcomers to GAs are usually introduced to 'fitness-proportionate selection', initially in a Generational GA. After the fitnesses of the whole current population are evaluated by whatever criterion has been chosen, and specified as real values, these values are summed to give the size of the reproductive 'cake'. Then each individual is allocated a probability of being chosen as a parent according to the size of its own slice of that cake.

This method has three virtues: it will indeed selectively favour those with higher fitnesses; it has some tenuous connection with the different biological notion of fitness (although in the biological sense any numbers associated with fitness are based on observing, after the event, just how successful an individual was at leaving offspring; as contrasted with the GA approach that reverses cause and effect here); and lastly, it happens to facilitate, for the theoreticians, some mathematical theorem-proving. But pragmatically, this method often leads to unnecessary complications.

What if the chosen evaluation method allocates some fitnesses that are negative? This would not make sense under the biological definition, but where the experimenter has chosen the evaluation criteria it may well happen. This is one version of the general re-scaling issue: different versions of a fitness function may well agree in ordering of different members of the population, yet have significantly different consequences. One worry that is often mentioned is that, under fitness-proportionate selection, one often sees (especially in a randomized initial population) many individuals scoring zero, whilst others get near-zero; though this may reflect nothing more than luck or noise, the latter individuals will dominate the next generation to the complete exclusion of the former.

These issues motivated many complex schemes for re-scaling fitnesses *before* then implementing fitness-proportionate selection. But -- although it was not initially reflected in the standard texts -- many practitioners of applying GAs to real-world problems soon abandoned them in favour of rank-based methods.

2.3 Rank-Based and Tournament Selection

The most general method of rescaling is to use the scores given by the fitness function to order all the members of the population from fittest to least fit; and thereafter to ignore the original fitness scores and base the probabilities of having offspring solely on these relative rankings. A common choice made is to allocate (at least in principle) 2.0 reproductive units to the fittest, 1.0 units to the median, and 0.0 units to the least fit member, similarly scaling pro rata for intermediate rankings; this is linear rank selection. The probability of being a parent is now proportional to these rank-derived numbers, rather than to the original fitness scores.

There are in practice some complications. If as is common practice P is an even number, the median lies between two individuals. The explicit programming of this technique requires some sorting of the population according to rank, which adds further complexity. Fortunately there is a convenient trick that generates a similar outcome in a much simpler fashion.

If two members of the population are chosen at random, their fitnesses compared, and the Winner selected, then the probability of the Winner being any specific member of the population exactly matches the reproductive units allocated under linear rank selection. This is easiest to visualize if one considers P such tournaments, in succession using the same population each time. Two individuals are chosen at random each time, so that each individual can expect to participate in two such tournaments. The top-ranker will win both its tournaments, for 2.0 reproductive units; a median-ranker can expect to win one, lose one, for 1.0 units; and the bottom-ranker will lose both its tournaments, for 0.0 units. A minor difference between the two methods will be more variance in the tournament case, around the same mean values.

Tournament selection can be extended to tournaments of different sizes, for instance choosing the fittest member from a randomly selected threesome. This is an example of *non-linear* rank selection. From both pragmatic and aesthetic perspectives, Tournament Selection with the basic minimum viable tournament size of two has considerable attractions for the design of a minimalist GA.

2.4 Who to Breed, Who to Die?

Selection can be implemented in two very different ways; either is fine, as long as the end result is to bias the choice of those who contribute to future generations in favour of the fitter ones. The usual method in GAs is to focus the selection on who is to become a parent, whilst making an unbiased, unselective choice of who is to die. In the standard Generational GA, every member of the preceding generation is eliminated without any favouritism, so as to make way for the fresh generation reproduced from selected parents. In a Steady State GA, once a single new individual has been bred from selected parents, some other individual has to be removed so as to maintain a constant population size; this individual is often chosen at random, again unbiased.

Some people, however, will implement a method of biasing the choice of who is removed towards the less fit. It should be appreciated that this is a *second* form of selective pressure, that will compound with the selective pressure for fit parents and potentially make the combined selective pressure stronger than is wise. In fact, one can generate the same degree of selective pressure by biasing the culling choice towards the less fit (whilst selecting parents at random) as one gets by the conventional method of biasing the parental choice towards the more fit (whilst culling at random).

This leads to an unconventional, but effective, method of implementing Tournament Selection. For each birth/death cycle, generate one new offspring with random parentage; with a standard sexual GA, this means picking both parents at random, but it can similarly work with an asexual GA through picking a single parent at random. A single individual must be culled to be replaced by the new individual; by picking two at random, and culling the *Loser*, or least fit of the two, we have the requisite selection pressure. One can argue that this may be closer to many forms of natural selection in the wild than the former method.

Going further, we can consider a yet more unconventional method, that combines the random undirected parent-picking with the directed selection of who is to be culled. Pick two individuals at random to be parents, and generate a new offspring from them; then use the same two individuals for the tournament to select who is culled -- in other words the weaker parent is replaced by the offspring.

It turns out that this is easy to implement, is effective. This is the underlying intuition behind the Microbial GA, so called because we can interpret what is happening here in a different way -- Evolution without Death!

2.5 Microbial Sex

Microbes such as bacteria do not undergo sexual reproduction, but reproduce by binary fission. But they have a further method for exchanging genetic material, *bacterial conjugation*. Chunks of DNA, plasmids, can be transferred from one bacterium to the next when they are in direct contact with each other. In the conventional picture of a family tree, with offspring listed below parents on the page, we talk of vertical transmission of genes ‘down the page’; but what is going on here is horizontal gene transfer ‘across the page’. As long as there is some selection going on, such that the ‘fitter’ bacteria are more likely to be passing on (copies of) such plasmids than they are to be receiving them, then this is a viable way for evolution to proceed.

We can reinterpret the Tournament described above, so as to somewhat resemble bacterial conjugation. If the two individuals picked at random to be parents are called A and B, whilst the offspring is called C, then we have described what happens as C replacing the weaker one of the parents, say B; B disappears and is replaced by C. If C is the product of sexual recombination between A and B, however, then ~50% of C’s genetic material (give or take the odd mutation) is from A, ~50% from B. So what has happened is indistinguishable from B remaining in the population, but with ~50% of its original genetic material replaced by material copied and passed over from A. We can consider this as a rather excessive case of horizontal gene transfer from A (the fitter) to B (the weaker).

3 The Microbal GA

We now have the basis for a radical, minimalist revision of the normal form of a GA, although functionally, in terms of Heredity, Variation and Selection, it is performing just the same job as the standard version.

This is illustrated in Fig. 1. Here the recombination is described in terms of ‘infecting’ the Loser with genetic material from the Winner, and we can note that this rate of infection can vary. In bacterial conjugation it will typically be rather a low percentage that is replaced or supplemented; to reproduce the typical effects of sexual reproduction, as indicated in the previous section, this rate should be ~50%. But in principle we may want, for different effects, to choose any value between 0% and 100%. In practice, for normal usage and for comparability with a standard GA, the 50% rate is recommended. The simplest way of doing this, equivalent to Uniform

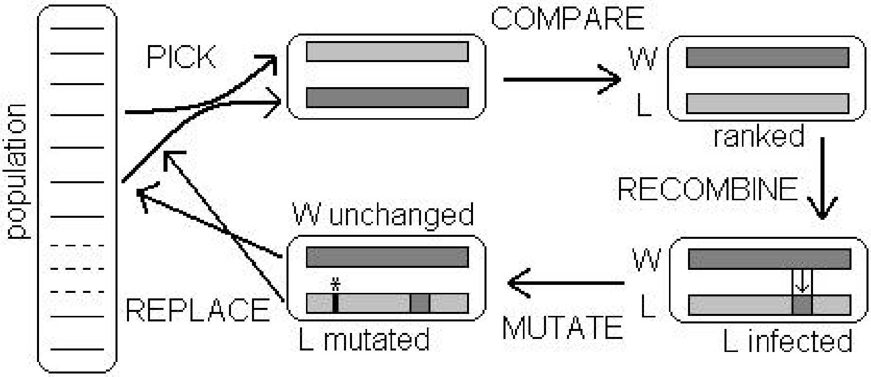


Fig. 1. The genotypes of the population are represented as a pool of strings. One single cycle of the Microbial GA is represented by the operations PICK (two at random), COMPARE (their fitnesses to determine Winner = W, Loser = L), RECOMBINE (where some proportion of Winner's genetic material 'infects' the Loser), and MUTATE (the revised version of Loser).

Recombination, is with 50% probability independently at each locus to copy from Winner to Loser, otherwise leaving the Loser unchanged at that locus.

From a programming perspective, this cycle is very easy to implement efficiently. For each such tournament cycle, the Winner genotype can remain unchanged within the genotype-array, and the Loser genotype can be modified (by 'infection' and mutation) *in situ*. We can note that this cycle gives a version of 'elitism' for free: since the current fittest member of the population will win any tournament that it participates in, it will thus remain unchanged in the population -- until eventually overtaken by some new individual even fitter. Further, it allows us to implement an effective version of geographical clustering for a trivial amount of extra code.

3.1 Trivial Geography

It is sometimes considered undesirable to have a panmictic population, since after many rounds of generations (or, in a Steady State GA, generation-equivalents) the population becomes rather genetically uniform. It is thus fairly common for evolutionary computation schemes to introduce some version of geographical distribution. If the population is considered to be distributed over some virtual space, typically two-dimensional, and any operations of parental choice, reproduction, placing of offspring with the associated culling are all done in a local context, then this allows more diversity to be maintained across sub-populations. Spector and Klein [3] note that a one-dimensional geography, where the population is considered to be on a (virtual) ring, can be as effective as the 2-D version, and demonstrate the effectiveness in some example domains. If we consider our array that contains the genotypes to be wrap-around, then we can implement this version by, for each tournament cycle: choose the first member A of the tournament at random from the whole population; then select the next member B at random from a deme, or sub-population that starts immediately after A in the array-order. The deme size D, <= P, is a parameter that decides just how local each tournament is.

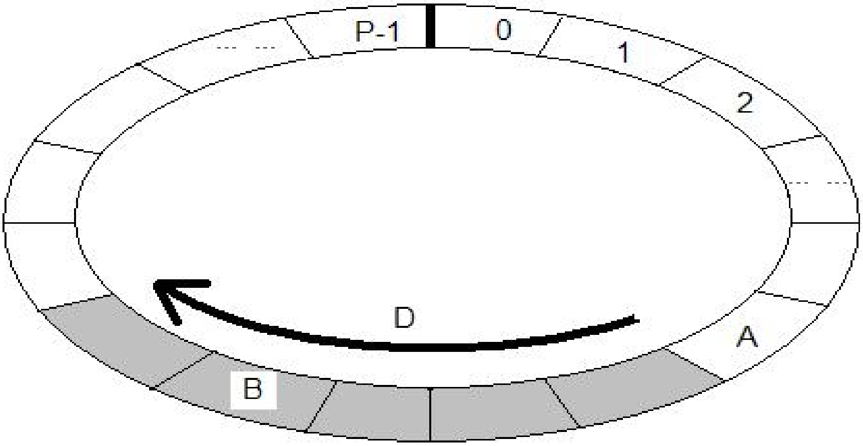


Fig. 2. The genotypes of the population are represented as geographically distributed on a ring, numbered from 0 to P-1. For a tournament, A is picked at random from the whole population; then B is picked at random from the deme (here of size D=5) immediately following A.

3.2 Program Code

The core part of the code for a Microbial GA is here given in pseudo-code based on C. We shall assume that there is a population size `P` of binary genotypes length `N`, stored in an array of the form `gene[P][N]`, that has been suitably initialized at random for the start of the GA; we have a function `eval(i)`, that returns some real value for the ith member of the population. This code represents a single tournament, that is repeated as many times as is considered necessary. We assume a pseudo-random number function `rnd()` that returns a real number in the range [0.0,1.0). `REC` is the recombination or 'infection' rate (suggested value 0.5), and `MUT` is the (per locus) mutation rate. `D` is the deme size.

This minimal GA routine incorporates Rank Selection, (variable rates of) Recombination and Mutation, (variable size) Demes, and Elitism, in just a few lines:-

```
void microbial_tournament(void) {
  int A,B,W,L,i;
  A=P*rnd();                        // Choose A randomly
  B=(A+1+D*rnd())%P;                // B from Deme, %P..
  if (eval(A)>eval(B)) {W=A; L=B;}  // ..for wrap-around
  else {W=B; L=A;}                  // W=Winner L=Loser
  for (i=0;i<N;i++) {               // walk down N genes
    if (rnd()<REC)                  // RECombn rate
      gene[L][i]=gene[W][i];        // Copy from Winner
    if (rnd()<MUT)                  // MUTation rate
      gene[L][i]^=1;                // Flip a bit
  }
}
```

4 Discussion

The original version (without demes) of this minimal Microbial GA has been widely used and taught at Sussex for over a decade [1], but only previously discussed in published form briefly in [2], in the context of Evolutionary Robotics. Another application in that field has been the Embodied Evolution of [4]. It is here presented in full for the first time with added Trivial Geography [3] providing demes.

We do not claim that it is more effective than the standard GA; indeed its underlying functionality is basically the same. It lends itself well to distributed, asynchronous applications. The minimalism makes it easy to teach and to implement. The re-interpretation in terms of bacterial conjugation opens up new perspectives and possibilities, including that of varying the rates of recombination or 'infection'.

References

1. Harvey, I.: The Microbial Genetic Algorithm (1996) (unpublished report)
2. Harvey, I.: Artificial Life, a Continuing SAGA. In: Gomi, T. (ed.) ER-EvoRob 2001. LNCS, vol. 2217, pp. 94–109. Springer, Heidelberg (2001)
3. Spector, L., Klein, J.: Trivial Geography in Genetic Programming. In: Yu, T., Riolo, R.L., Worzel, B. (eds.) Genetic Programming Theory and Practice III, pp. 109–124. Kluwer Academic Publishers, Boston (2005)
4. Watson, R.A., Ficci, S.G., Pollack, J.B.: Embodied Evolution: Distributing an evolutionary algorithm in a population of robots. Robotics Autonomous Systems 39(1), 1–18 (2002)

HybrID: A Hybridization of Indirect and Direct Encodings for Evolutionary Computation

Jeff Clune[1], Benjamin E. Beckmann[1], Robert T. Pennock[1,2], and Charles Ofria[1]

[1] Department of Computer Science and Engineering
[2] Department of Philosophy and Lyman Briggs College
Michigan State University, East Lansing, MI, USA
{jclune,beckma24,pennock5,ofria}@msu.edu

Abstract. Evolutionary algorithms typically use *direct encodings*, where each element of the phenotype is specified independently in the genotype. Because direct encodings have difficulty evolving modular and symmetric phenotypes, some researchers use indirect encodings, wherein one genomic element can influence multiple parts of a phenotype. We have previously shown that HyperNEAT, an indirect encoding, outperforms FT-NEAT, a direct-encoding control, on many problems, especially as the regularity of the problem increases. However, HyperNEAT is no panacea; it had difficulty accounting for irregularities in problems. In this paper, we propose a new algorithm, a Hybridized Indirect and Direct encoding (HybrID), which discovers the regularity of a problem with an indirect encoding and accounts for irregularities via a direct encoding. In three different problem domains, HybrID outperforms HyperNEAT in most situations, with performance improvements as large as 40%. Our work suggests that hybridizing indirect and direct encodings can be an effective way to improve the performance of evolutionary algorithms.

Keywords: Indirect (generative, developmental) encodings (representations), artificial neural networks, neuroevolution, evolutionary algorithms.

1 Introduction

Evolutionary algorithms, such as neuroevolution and genetic algorithms, typically use *direct encodings*, where each element of a phenotype is independently specified in its genotype. However, these direct encodings are limited in their ability to evolve complex, modular, and symmetric phenotypes because individual mutations cannot produce coordinated changes to multiple elements of a phenotype [1]. Such coordinated mutational effects can occur with *indirect encodings*, also called *developmental* or *generative* encodings, wherein a single element in a genotype can influence many parts of the phenotype [1, 2]. Indirect encodings have been shown to produce highly regular solutions to problems [1, 3-5], but their bias toward regularity makes it difficult for them to properly handle irregularities in problems [4].

In this paper, we propose a new algorithm that is a Hybridized Indirect and Direct encoding (HybrID), which combines the benefits of both encodings. Although we present one specific implementation of a HybrID, we apply the term to any combination

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 134–141, 2011.

of indirect and direct encodings. While we are not aware of any prior work that specifically combines direct and indirect encodings, researchers have previously altered representations during evolutionary search, primarily to change the precision of values being evolved by genetic algorithms [6]. Other researchers have employed non-evolutionary optimization techniques to fine-tune the details of evolved solutions [7]. However, such techniques do not leverage the benefits of indirect encodings.

This paper presents results from problems where an indirect encoding has already been shown to outperform a direct encoding [3, 4], and demonstrates that HybrID can further improve performance by as much as 40%. The major contribution of this paper is to introduce HybrID as a type of evolutionary algorithm that can both exploit a problem's regularities and account for its irregularities.

2 The Indirect and Direct Encodings and Their Hybridization

The HybrID in this paper first runs HyperNEAT [8], an indirect encoding for evolving artificial neural networks (ANNs), and then *switches* to FT-NEAT, its direct-encoding control [3, 8]. We next describe each encoding and their hybridization.

2.1 HyperNEAT, the Indirect Encoding

HyperNEAT is an indirect encoding for evolving ANNs that is inspired by the way natural organisms develop [8]. It evolves Compositional Pattern Producing Networks (CPPNs) [9], each of which is a genome that encodes an ANN phenotype (also called a *substrate*) [8]. Each CPPN is itself a directed graph, where each node is a mathematical function, such as sine or Gaussian. The nature of these functions can facilitate the evolution of properties such as symmetry (e.g., an absolute value or Gaussian function) and repetition (e.g., a sine function) [8, 9]. The signal on each link in the CPPN is multiplied by that link's weight, which can alter its effect.

A CPPN is queried once for each link in the ANN substrate to determine that link's weight. The inputs to the CPPN are the Cartesian coordinates of both the *source* (e.g., [x_1=2, y_1=3]) and *target* (e.g., [x_2=1, y_2=5]) nodes of a link and a constant bias value. The CPPN takes these five values as inputs and produces one or two output values, depending on the substrate topology. If there is no hidden layer in the substrate, the single output is the weight of the link between a source node on the input layer and a target node on the output layer. If there is a hidden layer, the first output value determines the weight of the link between the associated input (source) and hidden layer (target) nodes, and the second output value determines the weight of the link between the associated hidden (source) and output (target) layer nodes. All pairwise combinations of source and target nodes are iteratively passed as inputs to a CPPN to determine the weight of each substrate link.

HyperNEAT is capable of exploiting the geometry of a problem [8]. Because the link values between nodes in the final ANN substrate are a function of the geometric positions of those nodes, HyperNEAT can exploit such information when solving a problem [8, 10]. In the case of quadruped locomotion, this property helped HyperNEAT produce gaits with front-back, left-right, and four-way symmetries [3, 10].

The evolution of the population of CPPNs occurs according to the principles of the NeuroEvolution of Augmenting Topologies (NEAT) algorithm [11], which was originally designed to evolve ANNs. NEAT can be fruitfully applied to CPPNs because of their structural similarity to ANNs. For example, mutations can add a node, and thus a function, to a CPPN graph, or change its link weights. The NEAT algorithm is unique in three main ways [11]. Initially, it starts with small genomes that encode simple networks and slowly *complexifies* them via mutations that add nodes and links to the network, enabling the algorithm to evolve the network topology in addition to its weights. Secondly, NEAT has a fitness sharing mechanism that preserves diversity in the system and gives time for new innovations to be tuned by evolution before competing them against more adapted rivals. Finally, NEAT tracks historical information to perform intelligent crossover while avoiding the need for expensive topological analysis. A full explanation of NEAT can be found in [11].

2.2 FT-NEAT, a Direct-Encoding Control

To isolate the effects of changing only the representation, the direct encoding switched to after the HyperNEAT stage is *Fixed Topology NEAT* (FT-NEAT) [3]. FT-NEAT directly evolves each weight in the ANN independently and does not use complexification. All other elements from NEAT (e.g., its crossover and diversity preservation mechanisms) remain the same between HyperNEAT and FT-NEAT.

2.3 A HybrID of HyperNEAT and FT-NEAT

The HybrID presented in this paper runs HyperNEAT for a fixed number of generations and then the encoding is changed to FT-NEAT at a *switch point*. To switch, we transfer the ANN phenotypes of each individual in the HyperNEAT population to FT-NEAT genomes that are then further evolved with FT-NEAT. In the discussion section we describe alternate HybrID instantiations.

3 Results from Three Problem Domains

We compare HybrID to controls on three problems that have scalable regularity to assess HybrID's performance on versions of the same problem with varying amounts of problem regularity. The first two problems were chosen because they are easy to conceptualize, and the third is a more challenging, realistic problem.

We conducted 50 runs of each experimental treatment in this paper, and all data plotted is averaged across them. The parameter configurations for all experiments are similar to those in [3, 4, 8, 10], and can be viewed at http://devolab.msu.edu/ SupportDocs/Hybrid. The mutation rate per link was 0.08 for HyperNEAT and 0.0008 for FT-NEAT; preliminary experiments revealed these to be effective mutation rates for each encoding. FT-NEAT has a lower per-link mutation rate because its genome has many more mutational targets than HyperNEAT. Additional experiments showed no statistically significant increase in HyperNEAT's performance on all three problems when its mutation rate was dropped to 0.0008 at the switch point.

The Target Weights Problem: We begin our analysis with the simple test problem of evolving a specific target ANN. At the start of each run, a target weight is assigned to each link in an ANN, and during the run fitness values are set in proportion to each organism's similarity to that target ANN. As in previous work [4], we scaled the regularity of this problem by varying the percentage of links in the target ANN that were the same, randomly chosen value (Q), from 0% to 100%. All non-Q target weights were each independently assigned a random value. Target weights were in the range [-3, 3]. The ANN had 9 input and 9 output nodes, and was fully connected. HybrID switched from HyperNEAT to FT-NEAT at 100 generations, and the experiment lasted a total of 1000 generations. The population size was 1000.

As previously shown [4], HyperNEAT quickly discovered the regularity in the more regular versions of this problem, but had difficulty making exceptions to account for irregularities, even after hundreds of generations (Fig. 1). FT-NEAT, on the other hand, was slower, but eventually performed well, in part because the problem has no epistatic interactions and thus coordinated mutational effects are not required. HybrID combined the best attributes of both encodings: it quickly discovered the regularity of the problem and, after the encoding switch, was able to further optimize solutions by accounting for irregularities. While HybrID and FT-NEAT eventually evolved solutions of similar quality, early on HybrID did better on more regular problems and less well on less regular problems. HybrID significantly outperformed both HyperNEAT and FT-NEAT at generation 250 on the 70%, 80%, and 90% regular problems (p < .01, Wilcoxon rank-sum test). Earlier switch points further improved the speed at which HybrID made progress on this problem (data not shown).

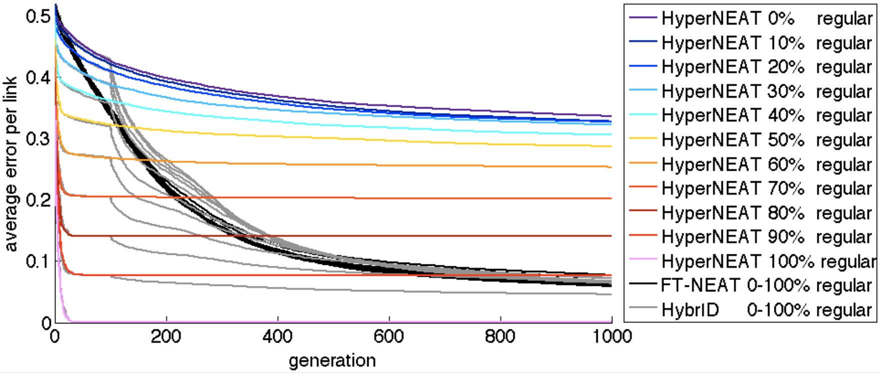


Fig. 1. A comparison of HyperNEAT, FT-NEAT and HybrID on a range of problem regularities for the target weights problem. For each regularity level, a HybrID line (gray) departs from the corresponding HyperNEAT line (colored) at the switch point (generation 100). The performance of FT-NEAT (black lines) was unaffected by the regularity of the problem, which is why the lines are overlaid and indistinguishable. HybrID outperforms HyperNEAT and FT-NEAT in early generations on versions of the problem that are mostly regular but have some irregularities.

The Bit Mirroring Problem: The next problem, called *bit mirroring*, [4] is more challenging and realistic because it has epistasis. In this problem, each input node is paired with a target output node, and the goal is to pass a value from each input node to its associated target output node. The ideal solution features a positive-weight link between each input-output pair, and a zero weight for every other link. The ANN substrate for this experiment had a 49-node input sheet and 49-node output sheet, each arranged in a 7x7 grid. Each input node was connected to all output nodes, totaling 2401 links. The most regular version of the problem paired each input node with the output node in the same row and column. We decreased the regularity, first by reducing the fraction of inputs that were constrained to be in the same *column* as their target, and then by further reducing the fraction of inputs that were constrained to be in the same *row* as their target. Because it has previously been shown that HyperNEAT outperforms FT-NEAT on all versions of this problem [4], here we only compare HybrID to HyperNEAT. A population of size 500 was evolved for 5000 generations and the switch point for HybrID was at generation 2500.

The results reveal that HybrID ties HyperNEAT on the most regular versions of this problem, and provides a significant fitness improvement over HyperNEAT on all versions of the problem that have a certain amount of irregularity (Fig. 2). HybrID's advantage over HyperNEAT was largest on problems of intermediate regularity.

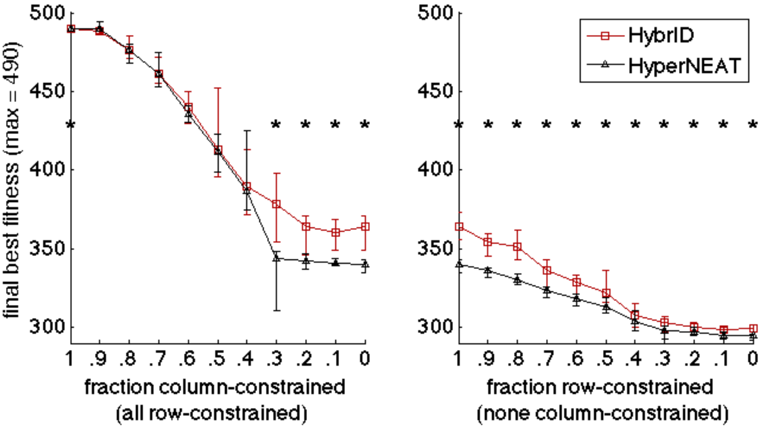


Fig. 2. HybrID vs. HyperNEAT performance on versions of the bit mirroring problem with regularity decreasing from left to right. Plotted are median values ± the 25th and 75th quartiles. Asterisks indicate p < . 05 (Wilcoxon rank-sum test).

The Simulated Quadruped Controller Problem: The previous two test problems were chosen because they are easy to conceptualize, their regularity can be scaled, and because the ideal solution is known *a priori*. However, it is also important to demonstrate that HybrID succeeds on more complicated, real-world problems, such as evolving controllers for legged robots. HyperNEAT has previously been shown to evolve fast, natural gaits for simulated legged robots [3]. The evolved gaits, however, were extremely coordinated, with all legs often moving in near perfect synchrony. We tested HybrID on this problem to determine if it would improve fitness by facilitating the fine-tuning of aspects of the controller, such as the movements of individual legs.

We repeated the experiment from [3] with HyperNEAT and HybrID, except that half the simulation time was allotted per evaluation due to computational limitations.

The ANN substrate consisted of three 5x4 Cartesian grids of nodes forming input, hidden, and output layers. Adjacent layers were completely connected, meaning that there were $(5\text{x}4)^2 \text{ x } 2 = 800$ links in each substrate. The inputs to the substrate were the current angles of each of the 12 joints of the robot (described below), a touch sensor that provides a 1 if the lower leg is touching the ground and a 0 if it is not, the pitch, roll, and yaw of the torso, and a modified sine wave (to facilitate the production of periodic behaviors). The sine wave was the following function of time (*t*) in milliseconds: $\sin(120 \text{ x } t) \text{ x } \pi$. This function produces numbers from –πto π, which was the range of the unconstrained joints. During preliminary tests, we experimentally found the constant 120 to produce fast, natural gaits, and determined that the touch, pitch, roll, yaw, and sine inputs all contributed to the ability to evolve fast gaits [3].

The ANN substrate outputs were the desired angles for each joint, which were fed into proportional controllers that applied forces to move the joints toward the desired angles. Robots were evaluated in the ODE physics simulator (www.ode.org). The rectangular torso of the robot was (in ODE units) 0.15 wide, 0.3 long, and .05 tall. Each of four legs was composed of three cylinders (length 0.075, radius 0.02) and three hinge joints. The first cylinder functioned as a hip bone. It was parallel to the proximal-distal axis of the torso and barely stuck out from it. The other two cylinders were the upper and lower leg. There were two hip joints and one knee joint. The first hip joint allowed the legs to swing forward and backward (anterior-posterior) and was constrained to 180° such that, at maximum extension, it was parallel with the torso. The second hip joint allowed a leg to swing in and out (proximal-distal). Together, the two hip joints approximated a universal joint. The knee joint swung forward and backward. The second hip and knee joints were unconstrained.

Each controller in a population of 150 was simulated for 3000 time steps (3 seconds). Experiments lasted 1000 generations with a switch point at generation 500. Trials were cut short if any part of the robot except its lower leg touched the ground, or if the number of direction changes in joints exceeded 960. The latter condition reflects the fact that servo motors cannot be vibrated incessantly without breaking. The fitness of controllers was the following function of *d*, the maximum distance traveled: 2^{d^2}. The exponential nature of the function magnified the selective advantage of small increases in the distance traveled. Because HyperNEAT greatly outperforms FT-NEAT on this problem [3], we compare HybrID to only HyperNEAT.

HybrID should excel when a problem is mostly regular but has some irregularities. Adding faulty joints to the quadruped provides such a problem. In addition to experiments with no faulty joints, we also conducted tests where the proportional controller of 1, 4, 8 or all 12 joints had an error such that if the ANN specified a desired target angle of A, the actual desired angle fed to the proportional controller was A + E, where E is an error value in degrees in the range [-2.5, 2.5]. The value E was set once at the beginning of each run for each faulty joint and was constant throughout the run. This is analogous to expected variation in joint function due to manufacturer error.

The results show that HybrID yielded an improvement over HyperNEAT on all versions of this problem (Fig. 3, $p < 0.001$, Wilcoxon rank-sum test). The improvement of HybrID over HyperNEAT was 6, 10, 30, 40, and 38 percent, respectively, for treatments with 0, 1, 4, 8, and 12 faulty joints. The performance boost from HybrID tended to increase with more faulty joints, and thus roughly correlated with the irregularity of the problem. Interestingly, HybrID with 1 faulty joint actually outperformed

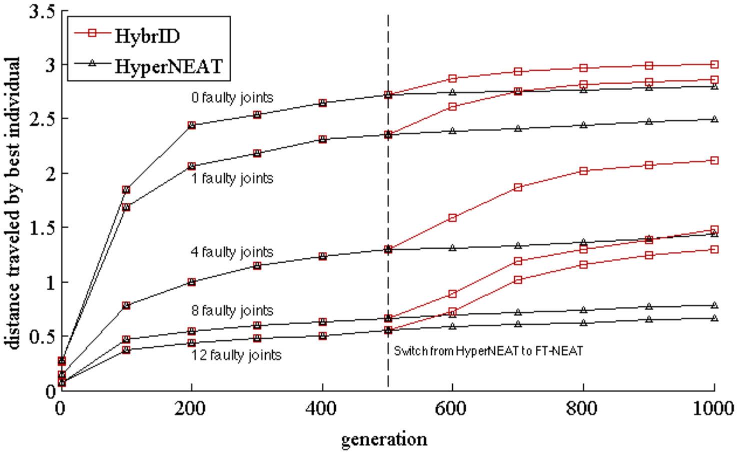


Fig. 3. HybrID outperforms HyperNEAT on all versions of the simulated quadruped controller problem. The increase generally correlates with the number of faulty joints.

HyperNEAT with 0 faulty joints ($p < 0.05$, Wilcoxon rank-sum test). The results suggest that HybrID can increase performance when it is allowed to fine tune the regularities produced by HyperNEAT.

4 Discussion and Conclusion

We presented only one of many possible HybrID implementations. The HybrID in this paper evolves first with an indirect encoding then switches to a direct encoding, and could be called a *switch-HybrID*. Another candidate HybrID implementation would have the indirect encoding produce a set of values (e.g., link weights) and the direct encoding evolve a set of offsets that modify the individual values. This *offset-Hybrid* would allow exceptions to be made while the indirect encoding is still evolving. HybrIDs can also be made with other indirect encodings, and in domains besides neuroevolution. Additionally, instead of occurring at a predefined time, the switch from indirect to direct encodings, or the addition of offsets, could occur automatically after fitness has stagnated. Future investigations are required to test the efficacy of different HybrID implementations.

This work also suggests that it is difficult for evolutionary encodings to simultaneously discover the regularity of problems and make exceptions to account for problem idiosyncrasies. At a minimum, HyperNEAT exhibits this deficiency in its present form. Theoretically, HyperNEAT should be able to make any exception required, but in practice it frequently does not. We predict that similar problems exist with other generative encodings, although additional research is required to test our prediction. It is an open challenge for the field to improve current generative encodings to enable the encoding of both regularities and exceptions to those regularities. If such developments are made to generative encodings, it might obviate the need for HybrID algorithms, but until that time HybrID remains an effective enhancement of generative encodings.

Many real world problems have regularities but also require exceptions to be made. It is important for evolutionary algorithms to both exploit such regularity in problems and account for their irregularities. We have shown that HybrID, a combination of indirect and direct encodings, accomplishes this goal by first discovering the regularity inherent in a problem and then accounting for its irregularities. We validated the algorithm on two simpler test problems and on a more challenging, real-world problem. HybrID frequently outperformed HyperNEAT, sometimes by as much as 40%. While further research is needed to see how HybrID works with other pairs of indirect and direct encodings, alternate HybrID implementations, and on additional problems, these preliminary results suggest that HybrID is an effective algorithm for evolving solutions to complex problems.

Acknowledgments. NSF Grants CCF-0750787, CNS-0751155, and CCF-0820220, U.S. Army Grant W911NF-08-1-0495, and a Quality Fund Grant from MSU. Thanks to helpful comments from Ken Stanley, Heather Goldsby and Dave Knoester.

References

[1] Hornby, G.S., Lipson, H., Pollack, J.B.: Generative representations for the automated design of modular physical robots. IEEE Transactions on Robotics and Automation 19(4), 703–719 (2003)

[2] Stanley, K.O., Miikkulainen, R.: A taxonomy for artificial embryogeny. Artificial Life 9(2), 93–130 (2003)

[3] Clune, J., Beckmann, B.E., Ofria, C., Pennock, R.T.: Evolving coordinated quadruped gaits with the hyperneat generative encoding. In: IEEE Congress on Evolutionary Computing (CEC), Trondheim, Norway, pp. 2674–2771 (2009)

[4] Clune, J., Ofria, C., Pennock, R.T.: How a generative encoding fares as problem-regularity decreases. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 358–367. Springer, Heidelberg (2008)

[5] Gruau, F.: Automatic definition of modular neural networks. Adaptive Behavior 3, 151–183 (1994)

[6] Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 3(2), 124–141 (1999)

[7] Grimbleby, J.: Automatic analogue circuit synthesis using genetic algorithms. IEE Proceedings - Circuits, Devices and Systems 147(6), 319–323 (2000)

[8] Stanley, K.O., D'Ambrosio, D.B., Gauci, J.: A hypercube-based indirect encoding for evolving large-scale neural networks. Artificial Life 15(2) (2009)

[9] Stanley, K.O.: Compositional pattern producing networks: A novel abstraction of development. Genetic Programming and Evolvable Machines 8(2), 131–162 (2007)

[10] Clune, J., Ofria, C., Pennock, R.T.: The sensitivity of hyperneat to different geometric representations of a problem. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Montreal, Canada, pp. 675–682 (2009)

[11] Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation 10(2), 99–127 (2002)

An Analysis of Lamarckian Learning in Changing Environments*

Dara Curran and Barry O'Sullivan

Cork Constraint Computation Centre
Department of Computer Science, University College Cork, Ireland
{d.curran,b.osullivan}@cs.ucc.ie

Abstract. It is widely recognised that many species adapt to complex and dynamic environments, but it is no longer accepted that an organism passes characteristics acquired during its lifetime to its offspring. However, in evolutionary computation such Lamarckian inheritance can be useful. Simulations of the benefits of Lamarckian inheritance have been reported in the literature. However, in this paper we present the first formal proof that Lamarckian inheritance can dominate more traditional individual (non-inheritable) learning. We present a parameterised model that can demonstrate conditions in which different inheritance types perform best, which we empirically validate.

1 Introduction

The study of natural systems has shown that many species are capable of adapting to real-world environments exhibiting dynamic and unpredictable situations. The underlying mechanics of such systems are relevant to the development of computer systems where novel circumstances may present themselves. Furthermore, such systems can inform the development of artificial systems capable of robust responses to such changes.

One form of learning which has recently been employed in computer simulations is *Lamarckian learning* [9, 11, 13]. Lamarckian learning allows individuals to store the information they have acquired during their lifetime in their genome, allowing this experience to be inherited directly by successive generations. While the existence of this form of learning in the natural world has been largely discredited [2], it has been the focus of evolutionary biology, evolutionary computation and machine learning research investigating its possible applications in artificial systems [11, 13, 8, 10, 1, 4–6]. An early model of Lamarckian learning [8] showed that it should be favoured in slowly changing environments, but this model did not distinguish between an individual's genotype and phenotype, as its purpose was not to compare Lamarckian learning with individual (non-inheritable) learning.

The comparison between Lamarckian learning and non-inheritable individual learning has been the focus of a number of simulations developed by a range of researchers [13, 11]. While these empirical models have provided some interesting results, there is as yet no formal proof describing the suitability of Lamarckian learning for changing environments. Our aim is to provide a formal analysis of Lamarckian learning using

* This work was supported by Science Foundation Ireland (Grant IN/05/I886).

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 142–149, 2011.

a simple mathematical model which is capable of describing the conditions in which Lamarckian learning is favourable. The model captures the experimental results obtained by other researchers and provides a sound framework from which to build more complex models describing more complex environmental changes. In addition, this paper presents a simple set of experiments which demonstrate that a population capable of selecting for or against Lamarckian learning will do so in accordance with the conditions predicted by our proof.

2 Related Work

Evolutionary Learning. A large number of natural organisms are born with innate abilities embodied in their genetic make-up. An accepted process for the development of these abilities is that of natural selection, as proposed by Darwin [3]. Individuals that are well-adapted to their environment are more likely to survive to reproduce and impart these abilities to the next generation through the transmission of genetic material. Over time, traits that are useful to a species become established in the population [3].

The process of natural selection can be seen as a learning mechanism applied to a species as a whole and is often dubbed *evolutionary learning*. While individuals within the species do not explicitly learn to adapt to their environment, the emergent properties of natural selection ensure that future generations are better suited to their environment than previous ones. In this model, the learning process is strictly confined to each organism's genetic material: the organism itself does not contribute to its survival through any learning or adaptation process [7].

Lamarckian Learning. There exist species in nature that are capable of learning, or adapting to environmental changes and novel situations at an individual level. Such learning, known as *life-time learning* is often coupled with evolutionary learning, further enhancing the population's fitness through its adaptability and resistance to change.

An evolutionary theory proposed by Jean-Baptiste Lamarck posits that adaptations which occur during an individual's lifetime can be directly passed on to that individual's offspring [9]. In other words, the claim is that lifetime learning directly influences evolution by altering an individual's genetic make-up during its life so that any such adaptations are genetically inherited by future generations. The theory has largely been discredited in light of the fact that such direct adaptations do not occur in nature. However, recent work has shown a number of mechanisms capable of neo-Lamarckian inheritance, particularly in the field of epigenetics [12].

3 An Analytical Model

The following is a simple mathematical model which enables us to compare evolutionary with Lamarckian learning, so one can determine when one is likely to dominate the other. While the model is very simple, we will see that the conclusions drawn from it are consistent with a much more sophisticated empirical study, which will be presented in the next section.

Let E_0 and E_1 denote two environments. Let T_0 and T_1 be the target solutions for environments E_0 and E_1 where $T_0 < T_1$; for the purposes of our analysis we can (arbitrarily) set $T_0 = 0$ and $T_1 = 1$. We denote the current target as T. We consider an environment oscillates between the two target values T_0 and T_1 at certain intervals. We begin with $T = T_1$ (arbitrarily) at generation 1. When the environment changes, $T = T_0$. Once the environment changes back, $T = T_1$ and so forth. For the time being, we will consider that environmental changes occur every generation. This is expanded further later in this section.

We let g be the genotype of an individual in a population, where g is a real number in the range $[T_0, T_1]$, and p be the phenotype of an individual in a population, where p is a real number in the range $[T_0, T_1]$. Each individual is born with genotype g and is capable of altering its genotype to create its phenotype p, measured at the end of an individual's lifetime. Adaptation is assumed to be always beneficial, moving an individual's genotype closer to the current target T. We denote by L the amount by which an individual's genotype is moved towards T to produce the individual's phenotype p. For simplicity, this scheme considers that individuals reproduce asexually. We will denote as g' the offspring genotype of the individual with genotype g.

We consider two types of adaptation mechanisms: *Baldwinian* and *Lamarckian*. Baldwinian adaptation allows populations to alter their genotypes to produce phenotypes closer to the current environmental target. Individuals are selected according to fitness and reproduce (asexually) to create new individuals. The mechanism of reproduction only considers the genotype of the individual, i.e. any lifetime adaptations stored in an individual's phenotype are not re-encoded into the genotype. Note that we do not intend Baldwinian adaptation in the strict sense of the Baldwin effect, but rather as a convenient label assigned to evolving populations capable of lifetime adaptation.

Lamarckian adaptation is equal to Baldwinian adaptation in all respects except that any adaptations acquired by an individual are re-encoded into the genotype and passed on to the next generation. For simplicity, this scheme assumes that both adaptation mechanisms evolve through some mutation operator. The mutation operator is assumed to provide an equal evolutionary force in both adaptation schemes. Because of this, the following does not consider the effect of the mutation operator, but rather concentrates on the relative impact of each adaptation scheme in isolation. In addition, we do not consider the effect of crossover as it is assumed to be mutually beneficial to either adaptation mechanism.

The offspring at every generation for each of the schemes can be defined as:

$$g'_{Baldwin} = g.$$

$$g'_{Lamarck} = \begin{cases} g + L & \text{if T = } T_1, \\ g - L & \text{if T = } T_0. \end{cases}$$

The phenotype of an individual using Baldwinian adaptation is defined as:

$$p_{Baldwin} = \begin{cases} g + L & \text{if T = } T_1, \\ g - L & \text{if T = } T_0. \end{cases}$$

The situation for an individual employing Lamarckian adaptation is slightly different. Table 1 shows the phenotypes of two individuals over time, one employing Baldwinian

Table 1. Phenotypes of individuals over time

Generation	1	2	3	4	5
Target	T_1	T_0	T_1	T_0	T_1
Baldwin	g+L	g-L	g+L	g-L	g+L
Lamarck	g+L	(g+L)-L	g+L	(g+L)-L	g+L

and the other Lamarckian learning. The Baldwinian individual's phenotype alternates between $g + L$ and $g - L$. However, since the Lamarckian individual re-encodes its phenotype into its genotype, its phenotype over time is not the same as the Baldwinian individual. Therefore, Lamarckian adapted phenotypes are defined as:

$$p_{Lamarck} = \begin{cases} g + L & \text{if T = } T_1, \\ g & \text{if T = } T_0. \end{cases}$$

Let d be the distance from an individual's phenotype to T (recall that g is initialised such that $g \leq T_1$ and $g \geq T_0$). The distance for an individual employing Baldwinian adaptation can be defined as:

$$d_{Baldwin} = \begin{cases} T - (g + L) & \text{if T = } T_1, \\ (g - L) - T & \text{if T = } T_0. \end{cases}$$

Note that this assumes that the learning mechanism stops once an individual matches the current target T, (i.e. that $(g + L) \leq T_1$ and $(g - L) \geq T_0$), and therefore, both $(T_1 - (g + L))$ and $((g - L) - T_0)$ will always be positive (recall that $T_0 < T_1$ and g is initialised such that it lies between $[T_0, T_1]$).

The distance for an individual employing Lamarckian adaptation can be defined as:

$$d_{Lamarck} = \begin{cases} T - (g + L) & \text{if T = } T_1, \\ g - T & \text{if T = } T_0. \end{cases}$$

The average distance for each individual, $\bar{d}$ can be calculated as:

$$\begin{aligned} \bar{d}_{Baldwin} &= \frac{(T_1 - (g + L)) + ((g - L) - T_0)}{2} \\ &= \frac{T_1 - T_0}{2} - L. \end{aligned}$$

For Lamarckian adaptation:

$$\begin{aligned} \bar{d}_{Lamarck} &= \frac{(T_1 - (g + L)) + (g - T_0)}{2} \\ &= \frac{T_1 - T_0}{2} - \frac{L}{2}. \end{aligned}$$

Therefore, Baldwinian adaptation has, on average, a smaller distance to the average target and consequently a higher average fitness.

Let's now consider how the gap between environmental changes affects the performance of each adaptation mechanism. Let n be the number of generations between E_0 and E_1. The average distance for the Baldwinian population is the same because adaptations are not re-encoded into the genotype:

$$\begin{aligned} dPop_{Baldwin} &= \frac{(T_1 - (x_i + L)) + ((x_i - L) - T_0)}{2} \\ &= \frac{T_1 - T_0}{2} - L. \end{aligned}$$

For Lamarckian adaptation, the average distance can be derived as:

$$\begin{aligned} dPop_{Lamarck} &= \frac{(T_1 - (x_i + nL)) + (x_i - T_0)}{2} \\ &= \frac{T_1 - T_0}{2} - \frac{nL}{2}. \end{aligned}$$

It follows that:

- When $n = 1$, $dPop_{Baldwin} < dPop_{Lamarck}$ (by $\frac{L}{2}$);
- When $n = 2$, $dPop_{Baldwin} = dPop_{Lamarck}$;
- When $n > 2$, $dPop_{Baldwin} > dPop_{Lamarck}$.

Therefore, we can say that, on average, Lamarckian adaptation is better suited to situations where the gap between environment changes is more than 2. Note that this is restricted to the situation where environments oscillate between extremes and the learning mechanism moves an individual towards a target by a constant amount.

4 Experiments

Given the above proof, it seems likely that Lamarckian learning should be selected for a population faced with an environment in which changes occur less frequently. To investigate this, we devised a simple experiment that allows individuals within an evolving population to prefer Lamarckian learning over evolutionary learning depending on the merits of either approach in a specific setting. We chose a simple problem domain, and illustrate the concept using a genetic algorithm. In one environment the target phenotype is one which contains all 1s and in the other, one which contains all 0s.

Each individual in the population possesses a genome comprised of 1001 bits, where 1000 bits encode a possible solution and the last bit encodes whether the individual employs Lamarckian learning or not. Each individual is allowed a learning round where a number of its unfit alleles are improved by setting either to 1 or 0, depending on the environment. If an individual employs Lamarckian learning, these changes are re-encoded into its genome to be passed on to the next generation. If an individual does not employ Lamarckian learning, all changes are lost to the next generation.

A number of experiments were undertaken, examining both the effect of increasing the number of generations between environmental changes and also the number of alleles altered during the learning round applied to both populations. The former can be regarded as controlling the frequency of environmental change, while the latter controls

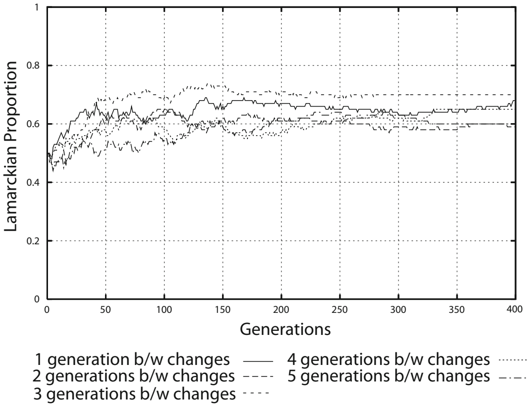


Fig. 1. 1 allele change per learning round

learning rate. We examined populations employing 1 and 50 allele changes in environments changes every 1, 2, 3, 4 and 5 generations. The results from these experiments (averaged from 20 independent runs) are illustrated in Figures 1 and 2.

Figure 1 shows populations allowed to change 1 unfit allele each generation. Although the results show a slight trend towards Lamarckian adaptation, the selection process does not truly discriminate between Lamarckian and Baldwinian learning regardless of the number of generations between changes. The modest amount of adaptation that occurs when only one allele is changed is not enough to bring out the characteristics and relative advantages or disadvantages of either learning mechanism. By contrast, the results obtained for 50 allele changes illustrated in Figure 2 show a clear divergence between Baldwinian and Lamarckian learning according to the number of generations between changes. Lamarckian adaptation is selected *against* in environments where there are 3 or less generations between changes and selected *for* in environments with larger gaps between changes. Clearly, increasing the number of allele changes allows the selection process to distinguish between Baldwinian and Lamarckian learning.

In summary, these results show that Lamarckian learning is not favoured by selection in environments where changes occur very frequently. In addition, the results show that as the gaps between environmental changes increases, the probability of Lamarckian learning being selected increases. Finally, in the situation where changes occur every 3 generations, the probability is split (slightly disfavouring Lamarckian learning). This is because the relative advantage of Lamarckian learning in this environment is not sufficient to fully take over the population. What is crucial is that individuals employing Lamarckian learning continue to exist in much larger numbers than in environments with smaller gaps between environmental changes. These results correlate with both our formal analysis of Lamarckian adaptation and with previous empirical research on Lamarckian learning.

A brief examination of the fitness values of the Baldwinian and Lamarckian populations for each environment further illustrates the selection pressure exerted by the environmental conditions on the Lamarckian evolution gene. An experiment was conducted with two populations, one employing Baldwinian learning and the other Lamarckian

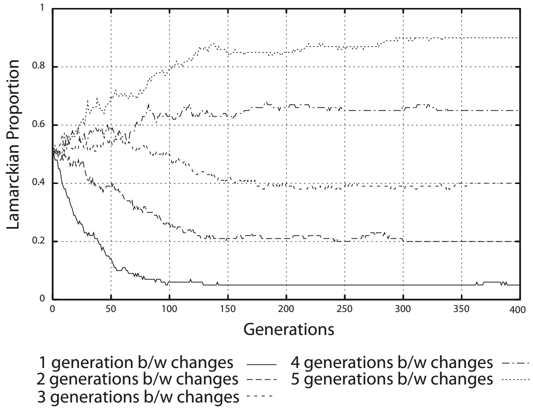


Fig. 2. 50 allele changes per learning round

learning. Both populations were allowed 50 allele changes during the lifetime learning stages. The average fitness values ($\bar{F}$) over the entire experiment were computed for each population as follows:

$$\bar{F}_{Baldwin} = \frac{1}{G}\sum_{g=0}^{G}\bar{f}_{Baldwin}(g)$$

$$\bar{F}_{Lamarck} = \frac{1}{G}\sum_{g=0}^{G}\bar{f}_{Lamarck}(g)$$

where G is the number of generations and f_g is the average fitness of the populations at generation g. The difference between Lamarckian and Baldwinian populations (D) was calculated as:

$$D = \bar{F}_{Lamarck} - \bar{F}_{Baldwin}.$$

Table 2 shows the average fitness for both Baldwinian and Lamarckian populations as well as the difference measure D for each environmental setting. Negative values indicate fitter Baldwinian populations while positive values indicate fitter Lamarckian populations. It is clear from these results that Lamarckian learning produces fitter populations in environmental conditions where the gap between environmental changes is larger than 2, as predicted by the analysis outlined in Section 3.

Table 2. Average fitnesses and differences for Baldwinian and Lamarckian populations

Change Interval	Baldwin	Lamarck	Difference
1	584.1	577.7	-6.4
2	575.5	570.5	-5.0
3	568.2	575.4	7.2
4	565.6	577.4	11.8
5	564.7	581.2	16.5

5 Conclusions

This paper presented a formal analysis of Lamarckian learning showing the conditions in which Lamarckian learning can be a better choice over evolutionary learning. We presented the first formal analytical proof that can be used to show that Lamarckian inheritance can dominate more traditional individual (non-inheritable) learning schemes. The findings of this model were validated through a set of simple experiments showing that the evolutionary pressure to select for or against Lamarckian learning closely follows the conditions predicted by the proof. Future work will examine the suitability of Lamarckian learning in more complex environments.

References

1. Cortez, P., Rocha, M., Neves, J.: A lamarckian approach for neural network training. Neural Process. Lett. 15(2), 105–116 (2002)
2. Crick, F.H.C.: Central dogma of molecular biology. Nature 227, 561–563 (1970)
3. Darwin, C.: The Origin of Species: By Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life. Bantam Classics (1859)
4. Giraud-Carrier, C.: Unifying learning with evolution through baldwinian evolution and lamarckism: A case study. In: Proceedings of the Symposium on Computational Intelligence and Learning (CoIL 2000), pp. 36–41. MIT GmbH (June 2000)
5. Grefenstette, J.J.: Lamarckian learning in multi-agent environments. In: Belew, R., Booker, L. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA, pp. 303–310. Morgan Kaufmann, San Francisco (1991)
6. Hayes, B.: Experimental lamarckism. American Scientist 87(6), 494–498 (1999)
7. Hinton, G.E., Nowlan, S.J.: How learning guides evolution. Complex Systems 1, 495–502 (1987)
8. Jablonka, E., Oborny, B., Molnar, I., Kisdi, E., Hofbauer, J., Czaran, T.: The adaptive advantage of phenotypic memory in changing environments. Philos. Trans. R Soc. Lond. B Biol. Sci. 29(350), 133–141 (1995)
9. Lamarck, J.B.: Philosophie zoologique ou exposition des considrations relatives l'histoire naturelle des animaux. UCP (reprinted 1984), Paris (1809)
10. Lamma, E., Moniz Pereira, L., Riguzzi, F.: Belief revision by lamarckian evolution. In: Boers, E.J.W., Gottlieb, J., Lanzi, P.L., Smith, R.E., Cagnoni, S., Hart, E., Raidl, G.R., Tijink, H. (eds.) EvoIASP 2001, EvoWorkshops 2001, EvoFlight 2001, EvoSTIM 2001, EvoCOP 2001, and EvoLearn 2001. LNCS, vol. 2037, pp. 404–413. Springer, Heidelberg (2001)
11. Paenke, I., Sendhoff, B., Rowe, J., Fernando, C.T.: On the adaptive disadvantage of lamarckianism in rapidly changing environments. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 355–364. Springer, Heidelberg (2007)
12. Richards, E.J.: Inherited epigenetic variation - revisiting soft inheritance. Nat. Rev. Genet. (2006)
13. Sasaki, T., Tokoro, M.: Comparison between lamarckian and darwinian evolution on a model using neural networks and genetic algorithms. Knowl. Inf. Syst. 2(2), 201–222 (2000)

Linguistic Selection of Language Strategies

A Case Study for Colour

Joris Bleys[1] and Luc Steels[1,2]

[1] Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Brussels, Belgium
jorisb@arti.vub.ac.be
[2] Sony Computer Science Laboratory, Paris, France
steels@arti.vub.ac.be

Abstract. Language evolution takes place at two levels: the level of language strategies, which are ways in which a particular subarea of meaning and function is structured and expressed, and the level of concrete linguistic choices for the meanings, words, or grammatical constructions that instantiate a particular language strategy. It is now reasonably well understood how a shared language strategy enables a population of agents to self-organise a shared language system. But the origins and evolution of strategies has so far been explored less. This paper proposes that linguistic selection, i.e. selection driven by communicative success and cognitive effort, is relevant and shows a concrete case study for the domain of colour on how different language strategies may cooperate and compete for dominance in a population.

1 Language Strategies and Language Systems

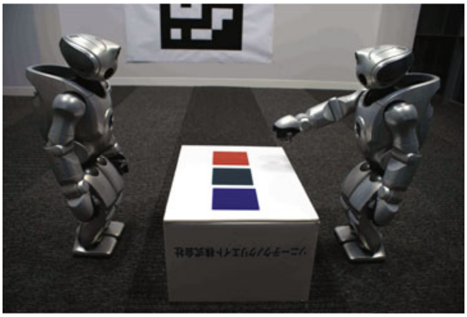


Fig. 1. Robotic experiments with the Colour Naming Game. The speaker draws attention to a chip by naming its colour and the hearer points to the chip which has been named. Agents give feedback after this interaction in order to share and align their colour categories and vocabularies.

Human languages are complex adaptive systems that are shaped and reshaped by their users, even in the course of a single dialogue [1]. They undergo change in order to remain adaptive to the expressive needs of the community, while maximising communicative success and minimising cognitive effort [2]. The past decade, considerable progress has been made to model the architecture and behaviour of ‘linguistic’ agents such that symbolic communication systems with properties similar to human languages may arise through language games [3]. In this paper we will use the

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 150–157, 2011.

Colour Naming Game, where the speaker uses a colour to draw the attention of the hearer to an object in the world [4]. Two agents drawn randomly from a population are shown a set of Munsell colour chips. The speaker chooses one chip as the topic, categorises the colour of this topic, and then searches in his own private lexicon how this category is named. The hearer gets the name, looks it up in his own lexicon, and then identifies and points to the chip in the context which fits best with the category named. If this is the chip that the speaker had in mind, the game is a success. Otherwise the speaker points to his original choice and the hearer can learn the name and the category expressed by it.

Colour Naming is an interesting domain because it has been studied intensely by anthropologists, neuroscientists, and psychologists, and so there is a significant body of empirical data available, including data about the evolution of colour terms and colour categories [5]. These empirical studies of colour naming in humans have shown three facts:

(1) There are different strategies by which speakers use colour to draw attention to objects in the world. One common strategy, which has been studied the most, is to use a limited set of basic colour prototypes utilising the full colour space, meaning the two hue opponent channels and brightness (for example: black, white, red, green, blue, yellow, pink, purple, brown, orange) [6]. We further call this the *Basic Colour Strategy.* Another strategy is to use only brightness, with words like “dark”, “shiny”, or “light”. We will call this the *Brightness Colour Strategy.* There are still other possibilities: to combine two basic colour prototypes (as in “bluish green” or “reddish orange”), to suggest colours by naming an object that typically has the colour (as in “lila” or “almond”), to combine the latter with basic colours as in “grass green”, “milk white” or “sky blue”, etc.

(2) There is considerable variation in the way in which a particular strategy is instantiated in a language and how it determines how languages change over time. For example, “red” is the name of a prototypical colour in English, roughly in the 625-740 nanometer range of the colour spectrum, but it used to be called “read” in Old English. The same colour prototype is called “rojo” in Spanish, “aka” in Japanese, “červený” in Czech, or “merah” in Indonesian. The basic colour prototypes used in different languages vary as well and there is also evolution, usually towards more and more refined colour prototypes [6]. For example, English speakers make a rather clear distinction between green and blue, but in Chinese and Japanese there is a single colour category which covers both areas, named “ao” in Japanese or “qīng” in Chinese. It is used for the (green) traffic light (“ao shingo”) or the colour of unripe bananas, but also for a blue sky (“aozora”). The Berinmo, a Papua New Guinea indigineous culture, has a word “wor” which covers some of the green region, a word “nol” which covers much of green, blue and blue/purple, a word “wap” which covers almost all the lightest colours, and a word “kel” which covers almost all dark colours [7].

(3) Interestingly, a kind of switch has often happened or is happening in predominantly brightness-oriented colour languages towards predominantly full-colour oriented languages which use both brightness and hue [8], showing that

there is not only evolution in how a strategy is instantiated (in other words which words and categories are used) but also which strategies a language community employs. Today's hues like "yellow", "brown", or "blue" were all expressing brightness-based distinctions in Old English before they became used as part of the *Basic Colour Strategy* in the late Middle English period (1350-1500) [9], showing that the same linguistic elements (e.g. the same words) may be used by different strategies leading to a kind of competition and mutual influence across strategies.

Given that we see variation and evolution at these two levels, we must conclude that individual language users master both *language strategies*, which are procedures for building, expanding, and adapting form-meaning mappings in order to achieve a particular communicative goal, and *language systems*, which are the concrete instantiations with respect to meanings (ontology), words (lexicon) or grammatical constructions (grammar) given a particular strategy. The *communal language strategies*, i.e. the strategies shared by all or most members of a population, and the *communal language system*, being the shared choices in a particular community, emerge out of the collective activity of all individuals and is not explicitly accessible nor represented.

The goal of this paper is to understand and model the competition, selection and evolution of language strategies using colour as a concrete case study, specifically the interaction between brightness-based and full-colour-based strategies as attested in the evolution of English and many other languages.

2 Language Strategies

The first step in the investigation is to operationalise the language strategies themselves. We have done this here for two strategies: the *Basic Colour Strategy* and the *Brightness Colour Strategy*. We have chosen the CIE L*u*v* space because the distance between two colours in this space accurately represents the psychological distances between these colours perceived by human subjects [10]. The L* dimension represents brightness (ranging from black to white), the u* dimension represents the red-green opponent channel and the v* dimension the yellow-blue channel. A two-dimensional projection of the Munsell chips is shown in Fig. 2. Colour categories are represented by a single point in this colour space, usually called its focal prototype [11], and categorisation can be modeled with a standard one-nearest neighbour classification algorithm. The prototypes nearest to each chip in the context are computed and categorisation is successful if there is a unique distinctive prototype found for the topic. This prototype is named and if this lead to a successful game, it is shifted with a very small factor towards the stimulus topic. When there is

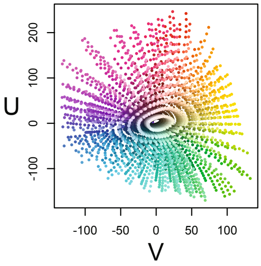
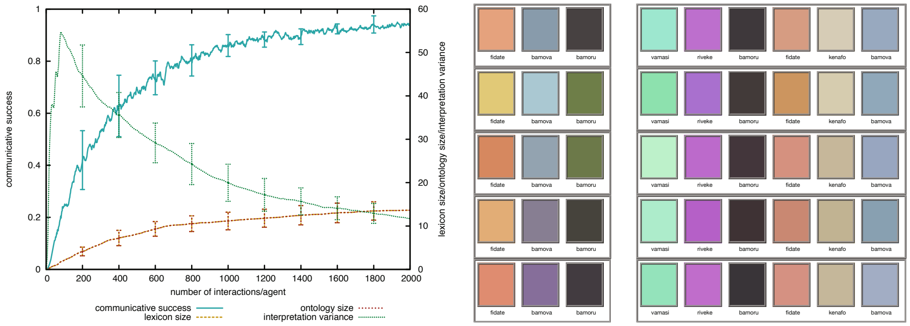


Fig. 2. A two-dimensional projection of the Munsell chips on the hue plane. A subset of these chips is used as the stimuli in a language game.

another stimulus with the same prototype as the topic, a new prototype is introduced using the topic as seed. This happens at a very low rate, to ensure that new categories and the names to express them become sufficiently shared in the population before another new category is invented. When the hearer encounters a name he has never heard before, he adds a new association between this name and a newly created colour category based on the current topic.

The *Basic Colour Strategy* uses all three dimensions of the colour space (both the brightness and the hue dimensions). Figure 3(a) shows that this strategy enables a population of agents to self-organize a colour lexicon from scratch. Figures 3(b) and 3(c) show the evolution of a colour lexicon in a typical experiment.



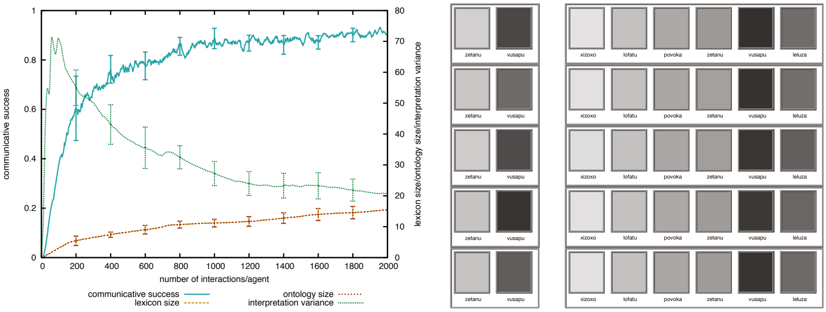
(a) (b) (c)

Fig. 3. The *Basic Colour Strategy* allows a population of agents (in this case 10) to self-organise a colour lexicon from scratch (a). The graph shows how steady high communicative success is reached with a lexicon of about 15 colour words. The evolution of a typical lexicon in a smaller population (5 agents), is shown after 400 (b) and 1200 (c) games per agent. Each row represents the lexicon of one agent.

The *Brightness Prototype Strategy* is similar to the previous strategy, but instead of taking all three dimensions into account, only the L* dimension of both the stimuli in the context and the prototypes of the colour categories are compared. While learning, the prototype of the used colour category is shifted on the L* axis towards the L* value of the topic. During invention, only the L* value of the topic is considered relevant. Figure 4(a) shows that this strategy is also adequate to allow a population of agents to self-organize and coordinate a colour lexicon from scratch. The resulting colour lexicon now consists of different shades of gray (see Figs. 4(b) and 4(c) for the evolution of a typical lexicon).

3 Strategy Selection

We can now turn to the main topic of this paper: how can there be selection and cooperation between different language strategies, and how can there be evolution, in the sense that one strategy overtakes another. Different strategies may



(a) (b) (c)

Fig. 4. The *Brightness Colour Strategy* also allows a population of agents (in this case 10) to evolve an adequate colour lexicon (a). The evolution of a typical lexicon in a smaller population (5 agents), is shown after 400 (b) and 1600 (c) games per agent. Each row represents the lexicon of one agent.

cooperate in the sense that one strategy may be better in certain circumstances than in others. For example, a brightness colour vocabulary is obviously more appropriate when talking about black-and-white photographs. But strategies also compete because a population will obviously be more successful if each language user adopts the same *default strategy* for dealing with the same sort of communicative problem. There is also competition for words and meanings among the different strategies. A hearer cannot know whether a particular word (e.g. “yellow”) is to be interpreted using one strategy (brightness-based) or another one (full-colour-based), particularly because both strategies could work in similar circumstances. For example, yellow colour chips are often the most bright ones and hence both strategies would work.

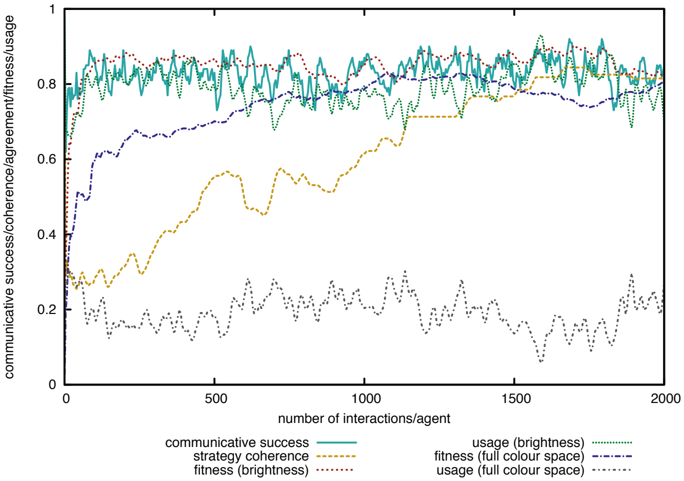
We have operationalised the linguistic selection of strategies in the following way. Each strategy is reified, in the sense that it is an object represented as such in the memory of the agents. A strategy has a score which reflects its ‘fitness’. This fitness is based on the communicative success of the words and meanings that were built or used with this strategy. The meanings, words or grammatical constructions are tagged with the strategy that has been used to invent or learn them. For example, if a word “dark” is acquired with the brightness-based strategy, it is tagged with that strategy. The same word may be tagged with different strategies because a learning agent does not know which strategy has been used by the speaker with a particular word and hence may have to make different hypotheses. In speaking, agents handle a communicative problem with the solution stored in their language system that had most success in the past and this solution implies a particular strategy. When the problem cannot be handled, the speaker has to expand his set of meanings and his lexicon and he prefers the default strategy, i.e. the strategy that had most success in the past. It is only when this strategy does not work that other alternative strategies are tried out in decreasing order of fitness. In listening, the hearer first applies his own stored solution to interpret the utterance, which again implies the use of the language strategy associated with this solution. When the hearer is confronted with an unknown word or with a situation in which his interpretation of the word does not work for the present context (because apparently the speaker used another strategy for the word), he uses first his own default strategy to figure out the meaning of the unknown word, and, if that does not work, he tries out alternative strategies, again in the order of decreasing fitness.

Due to space limitations, we can only show the outcome of one of our experiments to study the rich and complex dynamics that result from these behaviours. In this experiment, the set of prototypes is kept fixed, namely equal to the focal colours underlying the Spanish colour system [12]. However we left it open which strategy agents should use. For example, the word "morado" (purple) can both be interpreted in the full-colour space and in the brightness space. Two situations arise. In one situation, a single strategy becomes clearly dominant in the population. This could either be the brightness or the full colour strategy, depending on small fluctuations in the early stages (Fig. 5(a)). But we have also observed situations where one strategy becomes dominant first (for example, the brightness strategy) to be overtaken later by another strategy (i.c. the full-colour strategy), as seen in the history of English (Fig. 5(b)). The two strategies continue to co-exist in this case. Brightness is still used in circumstances where this gives a higher chance of communicative success, for example when colour chips are close in hue but distinct in brightness or when there is a word which has most of its success in the brightness dimension.

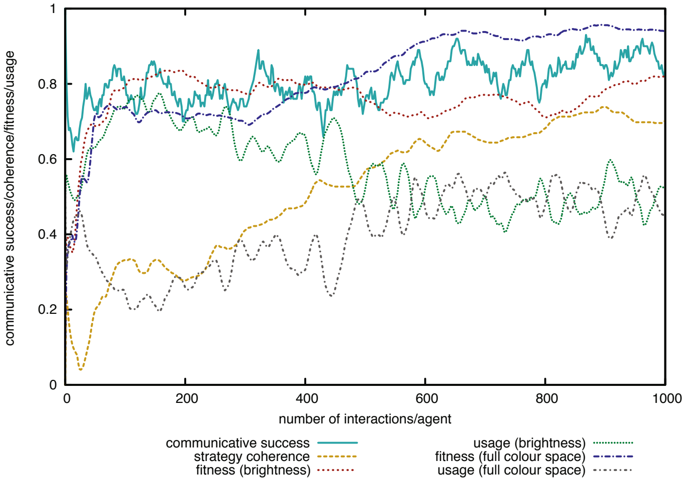
4 Conclusion

Language strategies can only compete with each other through the use of the language systems that they enable their users to build. And language systems can only be tested through the production and comprehension of concrete utterances. So we get selection at two levels: (1) The application of a language strategy by a population generates possible variation (possible categories, possible words) and those variants that lead to higher communicative success undergo positive selection and are hence re-used in future communications. We have shown that this selectionist process can be orchestrated by coupling communicative success to language adaptation (Figs. 3(a) and 4(a)). (2) The recruitment of cognitive functions generates possible language strategies and those strategies that lead to the construction of language systems with higher communicative success undergo positive selection, and are thus used even more in the future. We have shown here that this selectionist process can be orchestrated by having the agents keep track of which strategy they used for the building and interpretation of a particular word as well as the long term communicative success of each strategy (its fitness). We have shown that this dynamics allows a population to settle on a dominant default strategy although there is not necessarily a winner-take-all situation (Figs. 5(a) and 5(b)).

The same sort of competition and selection between different strategies has been observed in many areas of lexical and grammatical evolution. For example, there is currently an evolution going on in Spanish clitics ("le", "la", "lo")



(a)



(b)

Fig. 5. In (a) the *Brightness Colour Strategy* becomes entirely dominant and is used significantly more, whereas in (b) one strategy overtakes the dominant one which is reflected in the respective use of each strategy

whereby the etymological system of Standard Spanish, which uses clitics to express different cases (nominative, dative, accusative), is shifting to a referential system in which case differentiation is lost, but with existing forms recruited for expressing gender and number distinctions [13]. The two-level selectionist dynamics discussed here is relevant to understand more broadly how an individual may select the language strategies used in his or her language community and how language strategies may evolve in the historical evolution of a language.

Acknowledgements

This research is funded through a fellowship of the Institute of the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen), with additional funding from the EU FP7 ALEAR project and the Sony Computer Science Laboratory in Paris.

References

1. Garrod, S., Doherty, G.: Conversation, co-ordination and convention: An empirical investigation of how groups establish linguistic conventions. Cognition 53, 181–251 (1994)
2. Steels, L.: Language as a complex adaptive system. In: Schoenauer, M. (ed.) Proceedings of PPSN VI. LNCS, pp. 17–26. Springer, Berlin (2000)
3. Steels, L.: A self-organizing spatial vocabulary. Artificial Life 2(3), 319–332 (1995)
4. Steels, L., Belpaeme, T.: Coordinating perceptually grounded categories through language: a case study for colour. Behavioral and Brain Sciences 28(4), 469–489 (2005)
5. Hardin, C., Maffi, L.: Color categories in thought and language. Cambridge University Press, Cambridge (1997)
6. Berlin, B., Kay, P.: Basic color terms: their universality and evolution. University of California Press, Berkeley (1969)
7. Roberson, D., Davies, I., Davidoff, J.: Color categories are not universal: replications and new evidence from a stone-age culture. Journal of Experimental Psychology: General 129(3), 369–398 (2000)
8. Maclaury, R.: From brightness to hue: an explanatory model of color-category evolution. Current Antropology 33(2), 137–187 (1992)
9. Casson, R.: Color shift: evolution of English color terms from brightness to hue. In: Hardin, C., Maffi, L. (eds.) Color Categories in Thought and Language, pp. 224–239. Cambridge University Press, Cambridge (1997)
10. Fairchild, D.: Color appearance models. Addison Wesley Longman, Reading (1997)
11. Rosch, E.: Cognitive representations of semantic categories. Journal of Experimental Psychology 104(3), 192–233 (1975)
12. Lillo, J., Moreira, H., Vitini, I., Martín, J.: Locating basic Spanish colour categories in CIE L*u*v* space: identification, lightness segregation and correspondence with English equivalents. Psicológica 28, 21–54 (2007)
13. Fernández-Ordóñez, I.: Leísmo, laísmo, loísmo: estado de la cuestión. In: Bosque, I., Demonte, V. (eds.) Gramática descriptiva de la lengua española, vol. 1, pp. 1319–1390. Real Academia Española / Espasa Calpe, Madrid (1999)

Robots That Say 'No'

Frank Förster, Chrystopher L. Nehaniv, and Joe Saunders

Adaptive Systems Research Group, School of Computer Science
University of Hertfordshire, College Lane,
Hatfield, AL10 9AB. United Kingdom

Abstract. This paper reports on foundational considerations for experiments into the acquisition of human-like use and understanding of negation in linguistic utterances via a developmental robotics approach. For this purpose different taxonomies of negation in early child language are analysed in order to show the large variety of communicative functions that these different types of negation have. Requirements for robotic systems that aim at acquiring these utterances in a linguistically unconstrained human-robot dialog are derived from this analysis.

1 Introduction

This paper presents an analysis of negation in early child language which offers an alternative to the prevailing paradigm of propositional representation with one that considers language as a means to manipulate the world. Apart from being necessary in order to ground negative utterances this shift in the viewpoint seems more in accordance with evolutionary perspectives on language [9]. The discussion below contributes to ongoing research into achieving human-like language acquisition in robots via developmental processes [16,10,5].

The motivation for the development of existing frameworks is at least partially to tackle the symbol grounding problem by linking the symbolic representations of the system to sensorimotor data [8,15]. Some of the embodied frameworks such as [19] enable artificial agents to invent their own vocabulary and simple forms of grammar like word order for the purpose of communicating with each other in a language constructed by and understandable to the robotic participants of the dialog [18]. Other frameworks, e.g. [6] or [14], enable robots to learn and understand names for objects, actions, or spatial relations, simple propositions, or commands taken from natural human language. From the perspective of pragmatics all of these frameworks enable robots to engage in one or two types of speech acts: commenting on the state of the world, an assertive speech act, and following orders, a reaction to directive speech acts. As is known from speech act theory [1,17] human language has many more functions than only stating facts about the world or receiving and giving orders. Observations of the earliest language use of children show clearly that the *functional* scope of early human language transcends these types of speech acts [4]. Already in the pre-grammatical phase of early utterances children express linguistically attraction

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 158–166, 2011.

or aversion towards objects, actions and persons by requesting or refusing. The latter constitutes only an alternative means for conveying communicative intentions which in the pre-linguistic phase were conveyed via gestures or prosodically marked non-lexical utterances that accomplish the same function. Negatives play a major role in so called “langage de volonté” (language of will) [7] and many types of negation are prototypical examples for utterances that cannot be grounded without being linked to the volitional or affective state of an agent. Pea [13] uses the apt expression of “motor-affective sensorimotor intelligence” to characterize the capability that children must have in order to refuse things in a linguistic manner. This highlights the difference compared to pure sensorimotor grounding of object words like “cup” or “ball” or action words like “grasp” or “throw”. Prohibitive speech acts, acts of refusal, or acts of motivation-dependent denial are examples for negatives that are at least partially distinguished through their affective nature. Other negative speech acts include commenting on disappearance of objects, cessation of events, or expressing unfulfilled expectations which could be grounded with the existing frameworks. It can neither be assumed that the last mentioned types constitute the overwhelming majority of early negation nor can these be expected to be the dominant types of negation in a human-robot dialog with unconstrained language use on part of the human.

We examine the preconditions of this very early form of human speech that must be met by robotic agents in order to ground negatives and engage in negative speech acts in the context of such a dialog. Less affective utterances that comment on disappearance or express unfulfilled expectations, sometimes grouped together under the term of non-existence, have the advantage of being a natural counterpart to positive comments on and descriptions of the world and thus highlight the difference between recent frameworks and the new approach proposed here, which aims at extending these frameworks towards the grounding of negation including affect-heavy forms like rejection or prohibition.

The presented outline is meant to be a theoretical basis for research which is dedicated to enable robots to obtain early pre-grammatical human language. The account tries to stay close to early human language development while at the same time exposing the crucial properties of negation which have to be met by robotic systems in order to be able to engage in dialogues with humans. Section 2 gives a short overview of negative speech acts observed in early child language in the one-word stage and their characteristics. Different taxonomies are presented and compared with specific regard to properties that are crucial for grounding and representation of negatives in robotic language acquisition. These acts will be considered from the point of view of pragmatics, speech act theory, semiotics, and accounts of early language acquisition [4,12]. Section 3 outlines implications of the linguistic analysis for robotic frameworks and describes cornerstones of a computational model designed to enable humanoid robots to acquire early human language via dialog with a human partner. Section 4 concludes with a short summary.

2 Emergence of Negation in Human Language Development

Compared to words that name objects, actions or even abstract categories, negation is semantically hard to classify and is deeply interwoven with lexical meaning in general [13]. There are several strategies to categorize negation semantically with varying granularity (see also [13] for a discussion). To the knowledge of the authors there is despite many centuries of research on many aspects of negation no commonly accepted taxonomy with regard to its semantics. This circumstance indicates the difficulty of this enterprise. We adopt in the following the taxonomy proposed by Pea [13] as it is motivated by similarities in the situational context and the child's behavior which leads to an intuitive clustering. Moreover this taxonomy exhibits the nice feature of emphasizing different properties for the particular negation types that lend themselves readily to a translation into different requirements for the computational model delineated in section 3 below. Note that this taxonomy does not necessarily reflect the child's own distinction (see [13]). Other taxonomies can be found in [2] and [3] and will be compared to the adopted one. Table 1 summarizes the most frequent types of early negation and their characteristics in the period up to 25 months according to Pea [13]. The negation types in the table roughly emerge in the order listed, whereby the three middle types can also be exchanged with regard to order of emergence. The reason for this variability is simply that different parents from potentially different social classes speak very differently to their children which has a strong impact on their linguistic development. Nonetheless the tendency from a rather reactive behavior towards a behavior that requires an increasingly complex memory is observable.

Negation types not listed in table 1. Less frequent and therefore omitted types are make-believe, agreement to negative statements, motivation-dependent, and perspective-dependent denial (see [13]).

Rejection. Action-based negations that serve to reject objects, persons, activities or events in the immediate environment fall into this category. Often expressed as a simple "No" this type of negation is repeatedly reported to be the first type that emerges and has gestural and non-gestural predecessors long before the emergence of the first word. Rejections might be the most affective type of negation as they cannot be interpreted without reference to aversion.

Self-Prohibition. Utterances of negation in the context of objects or actions that have been forbidden previously by the teacher. One can observe these utterances in a scenario, where the child approaches a forbidden object, hesitates to touch it while looking at the teacher, approaches it again and so on. Self-prohibition is more complex than rejection in the sense that it requires a internal representation of the preceding external prohibition.

Disappearance. Typically expressed as "gone" or "all-gone" in English, disappearance negatives signal the disappearance of something in the immediate

Table 1. Most frequent types of early negation. Taxonomy and characteristics are compiled from Pea [13] except for characteristics tagged with * which were added by authors. (affective) means that corresponding negation types are not intrinsically affective but accompanied or caused by affective states. 'Adjacent' describes whether or not an utterance is produced in response to another one.

Negation type	**Topic**	**Characteristics**
Rejection	objects, persons, events, activities in the present	affective, adjacent and nonadjacent, action-based, no need for internal representation
Self-Prohibition	objects, persons, events, activities	nonadjacent, (affective)*, referent in present but assumes prohibition in past with regard to the same referent on part of the carer
Disappearance	objects, persons, events, activities	adjacent and nonadjacent, non-affective*, referent in the immediate past, typically in context of where-questions but also with declaratives, need for internal representation
Unfulfilled Expectation	objects, persons, events, activities	nonadjacent, (affective)*, referent in past or present, comment on constraint on activity or absence other than immediately prior disappearance or cessation, need for internal representation, first action-uses then existential and locational uses
Truth-functional Denial	facts of the situation which the proposition that is to be denied refers to	adjacent, non-affective*, response to a proposition, need for abstract internal representation

past. Like self-prohibition this type requires an internal representation, in this case of objects that disappear, but on a shorter time-scale. Short-term memory might be enough to support this type.

Unfulfilled Expectation. Like disappearance, unfulfilled expectations might be expressed as "gone" or "all-gone". They are uttered in contexts where objects are absent from their expected or habitual location without having been present in the immediate past. They also occur when an activity is unsuccessful in contrast to previous success, e.g. caused by broken toys. This type requires representations of objects, actions or events on longer time-scales than disappearance.

Truth-functional Denial. This type of negation is the most abstract in the taxonomy and the last to emerge. Generally it is a response to a proposition not held to be true by the child. To employ this kind of negation a child must be able to conduct logical judgments and use at least some truth-conditional semantics of language. The negated proposition is independent of the child's attitude towards it, distinguishing this from motivation- and perspective-dependent denial. The facts that are referred to might be in the present, past or future.

Taxonomy Proposed by Choi. Choi divides cross-linguistic negation up to age 40 months into nine categories, which roughly emerge in three developmental phases [3]:

Phase 1: (nonexistence), prohibition, rejection, (failure)
Phase 2: denial, (inability, epistemic negation)
Phase 3: normative negation, inferential negation

The brackets indicate a variation in the time of emergence for the different children observed by Choi. For some children they emerged during the indicated phase for others they emerged one phase later. With *normative negation* Choi describes negatives that occur when the state of affairs differs from the agents habitual expectations. Negations of this type are evoked through a deviation from normative expectation (e.g. persons go *in* and not *on* a car) or the unorthodox use of a tool. *Inferential negation* is related to denial, but in contrast the agent assumes that the conversation partner holds the statement which is to be denied to be true rather than having actually heard the latter expressing it. *Nonexistence* is expressed when the agent expects an entity to be present which is not or the entity disappears. Nonexistence according to Choi therefore seems to subsume Pea's categories of disappearance and unfulfilled expectation. *Prohibition* is used by the agent to negate action on part of the interaction partner. Pea does not list this category explicitly but notes that it is tightly linked to rejection. *Failure* is the reaction to a specific event that does not occur as expected. Like nonexistence it therefore seems to map to Pea's category of unfulfilled expectation. *Denial* probably subsumes all the differentiated types of denial of Pea. On the other hand Choi only quotes one example which falls into Pea's category of truth-functional denial. *Inability* describes an agent's negation of its physical ability to accomplish a task. It probably maps best to Pea's unfulfilled expectation category as this also subsumes constraints on activities. *Epistemic negation* describes negative responses to requests for information like “I don't know”. This type does not seem to be captured by Pea's taxonomy maybe because he did not observe it while monitoring the children of his study.

Taxonomy Proposed by Bloom. Bloom is cited by both of the other authors and seems to have provided one of the first accounts of the development of negation with regard to semantics [2]. At the same time her partition is the least elaborate one of the presented taxonomies. She only distinguishes the three types nonexistence, rejection and denial. Another striking difference is the fact that she reports nonexistence as the first negation type to emerge. Pea argues that this might be the case because Bloom focuses on negative meanings in sentences instead of their emergence during the single-word period.

Nonexistence: The referent is not manifest in the context, where there is an expectation of its existence, and is correspondingly negated in the linguistic expression.
Rejection: The referent actually exists or is imminent within the contextual space of the speech event and is rejected or opposed by the child.
Denial: The negative utterance asserts that an actual (or supposed) predication is not the case. The negated referent is not actually manifest in the context as it is in *rejection*, but it is manifest symbolically in a previous utterance.

Bloom's category of *denial* is evidently the same as Pea's category of truth-functional denial. *Rejection* is defined in the same way as given by Pea. *Nonexistence* maps to Pea's unfulfilled expectation. Blooms focus on sentential negation renders it incomparable with regard to developmental emergence and we therefore will not take it into account in what follows.

3 Prerequisites for Grumpy Robots

From the linguistic analysis above we infer that there are three crucial properties that distinguish the different types of negation. First they are distinguished through their relatedness to affect or volition. E.g. rejection cannot be grounded meaningfully without referring to the affective state of the agent. We therefore propose to replace sensorimotor grounding with sensorimotor-affective grounding to take this circumstance into account. An easy way to accomplish this might be to simply introduce two-dimensional values for affect denoting the valence (positive/negative) and the degree of affect. These values principally could be associated to instances of acquired concepts, linked to ongoing 'needs' of the robot, treated in the same way as other sensorimotor values and stored together with the other sensor readings for each frame in two additional dimensions. These values constituting the willingness to cooperate/accept or the opposite thereof could in the beginning be chosen arbitrarily indicating a tendency to accept or reject certain actions and objects. For the sole purpose of grounding the reason why an affective state is the way it is seems not of importance. In order to signal the state of affect to the interaction partner in language acquisition games this state should be mirrored by facial or body gestures of the humanoid. This is necessary in order to provoke negative utterances by the interaction partner in the case of non-cooperation/non-acceptance. This could be utterances like "No? You don't like this ball?". In later stages the arbitrary choice of affect could be replaced by more sensible measures like the outputs of a planning module depending on the decision if a certain action is useful or harmful to the agent to achieve its goals, to satisfy its internal drives, or to maintain its state [11].

The second distinction we can observe is the increasing complexity with regard to the required memory: from a purely reactive behaviour in the case of rejection to the need for long-term memory for unfulfilled expectations or normative negation. Interesting issues in this context are the need for an internalization of a previously external physical prohibition in the case of self-prohibition and

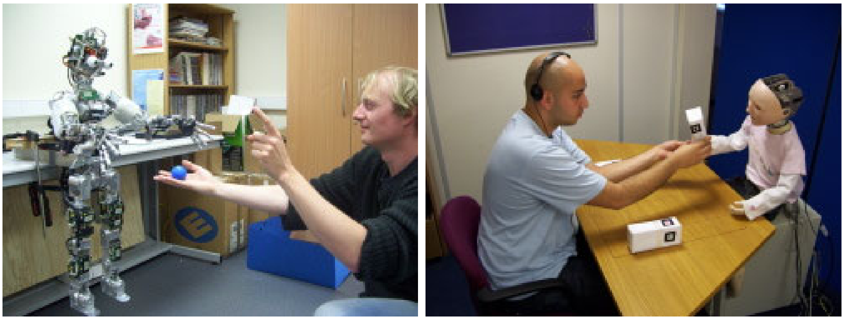


Fig. 1. Humanoids iCub (left) and Kaspar (right) used in language acquisition studies. These studies include work on learning vocabulary, holophrases, negation and grammar in a manner where language is grounded in active manipulation of objects, the social environment, and in sensorimotor experience in unconstrained interaction with human teachers. The work is targeted at the acquisition of human language-like capabilities.

the need for an internal representation of habitual locations and habitual functionality in the case of unfulfilled expectations. It is unclear whether these two memory-related requirements should be treated in a differentiated manner or whether it would be advantageous to distinguish them on a higher level but map them to the same underlying memory-structure. Experiments with different memory-models in which both negation types are acquired simultaneously should shed light onto this issue.

The third distinction for the types is given by their property of being adjacent or nonadjacent (see table 1). In order to engage in adjacent negation the agent must know when it is addressed. The only purely adjacent type is truth-functional negation, the latest negation type expected to emerge. In this case turn-taking is an issue and the analysis of prosody or an artificial replacement thereof as the agent is supposed to react to being spoken to and therefore must have the means to find out when it is addressed.

The large functional variability of the different negation types suggests that from a developmental point of view early negation might be rather considered as a family of related but different types of speech acts than being a variation of a single phenomenon. Seen from this perspective an implementation in the spirit of item-based constructions seems natural [20]. The latter support a treatment of these types as if they were entirely independent in the beginning and leave it up to machine learning algorithms to detect the similarity. Eventually a higher-level schema constructed through these mechanisms might emerge that could be labeled negation by an external observer.

4 Summary

We provided an analysis of early negation types with regards to their functional use which highlighted crucial differences with regards to their treatment in the context of robotic language acquisition in general and furthermore in a setting of human-robot dialogues. We propose to introduce a representation of affect or volition into the system in order to enable the acquisition of particular negation types like rejection. We also propose to introduce a differentiated memory architecture in order to support other types of negation like disappearance or unfulfilled expectations. The fact that all but one negation type are expressed also in a nonadjacent manner suggests that the integration of means to detect questions, e.g. through prosody extraction, is of importance but seems to be not of equal priority. Nonetheless turn-taking and the detection of questions is important in order to maintain the progress of the dialog which drives the grounded acquisition process itself. Initially it could be replaced by easier means of dialog control until e.g. robust methods for social cue and prosody detection are found.

Acknowledgement. The work described in this paper was conducted within the EU Integrated Project ITALK ("Integration and Transfer of Action and Language in Robots") funded by the European Commission under contract number FP-7-214668.

References

1. Austin, J.L.: How to Do Things with Words. Harvard University Press, Cambridge (1975)
2. Bloom, L.: Language Development: Form and Function in Emerging Grammars. M.I.T. Press, Cambridge (1970)
3. Choi, S.: The semantic development of negation: a cross-linguistic longitudinal study. Journal of Child Language 15(3), 517–531 (1988)
4. Clark, E.V.: First Language Acquisition. Cambridge University Press, Cambridge (2009)
5. Dominey, P.F., Metta, G., Nori, F., Natale, L.: Anticipation and initiative in human-humanoid interaction. In: 8th IEEE-RAS Intl. Conf. Humanoid Robots, pp. 693–699 (2008)
6. Dominey, P.F., Boucher, J.D.: Learning to talk about events from narrated video in a construction grammar framework. Artificial Intelligence 167, 31–61 (2005)
7. Guillaume, P.: Les débuts de la phrase dans le langage de l'enfant. Journal de Psychologie Normale et Pathologique 24, 1–25 (1927)
8. Harnad, S.: The symbol grounding problem. Physica D 42, 335–346 (1990)
9. Maynard Smith, J., Szathmáry, E.: The Major Transitions in Evolution. Oxford University Press, Oxford (2002)
10. Nagai, Y., Rohlfing, K.J.: Computational analysis of motionese toward scaffolding robot action learning. IEEE Trans. Autonomous Mental Development 1(1) (2009)
11. Nehaniv, C.L., Dautenhahn, K., Loomes, M.J.: Constructive Biology and Approaches to Temporal Grounding in Post-Reactive Robotics. In: McKee, G.T., Schenker, P. (eds.) Sensor Fusion and Decentralized Control in Robotics Systems II. Proc. of SPIE, vol. 3839, pp. 156–167 (1999)
12. Owens, R.E.: Language Development: An Introduction. Pearson Education, London (2007)
13. Pea, R.: The Development Of Negation In Early Child Language. In: Olson, D.R. (ed.) The Social Foundations of Language & Thought: Essays in Honor of Jerome S. Bruner, pp. 156–186. W.W. Norton, New York (1980)

14. Roy, D.: Semiotic schemas: A framework for grounding language in action and perception. Artificial Intelligence 167(1-2), 170–205 (2005)
15. Roy, D., Reiter, E.: Connecting language to the world. Artificial Intelligence 167, 1–12 (2005)
16. Saunders, J., Lyon, C., Förster, F., Nehaniv, C.L., Dautenhahn, K.: A Constructivist Approach to Robot Language Learning via Simulated Babbling and Holophrase Extraction. In: Proc. 2nd Intl. IEEE Symposium on Artificial Life, pp. 13–20 (2009)
17. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)
18. Steels, L.: The origins of syntax in visually grounded robotic agents. Artificial Intelligence 103(1-2), 133–156 (1998)
19. Steels, L.: Evolving grounded communication for robots. Trends in Cognitive Science 7(7), 308–312 (2003)
20. Tomasello, M.: Constructing a Language. Harvard University Press, Cambridge (2003)

A Genetic Programming Approach to an Appropriation Common Pool Game

Alan Cunningham and Colm O'Riordan

National University of Ireland, Galway, Ireland
alan.cunning@gmail.com, colm.oriordan@nuigalway.ie
http://www.it.nuigalway.ie/cirg/

Abstract. We investigate the performance of agents co-evolved using genetic programming techniques to play an appropriation common pool game. This game is used to study behaviours of users participating in scenarios with shared resources or interests eg. fisheries. We compare the outcomes achieved by the evolved strategies to that of human players as reported by [6]. Results show that genetic programming techniques are suitable for generating strategies in a repeated investment problem. We find that by using co-evolutionary methods, populations of strategies will quickly converge to nash equilibrium predicted by game theoretic analysis, but also lose many adaptive behaviours. Further, by evolving against a set of naive strategies, we show the creation of diverse and adaptive behaviours that play similarly to humans as described in previous experiments.

1 Introduction

Common pool problems are ones that deal with multiple agents and shared resources. These games model many scenarios in real world economic and social scenarios where many individuals are sharing resources e.g. fishing grounds. We are interested in a specific type of common pool problem – the appropriation game. The main characteristic of this game is that the level of yield from the pool is dependent on all the agents acting in the pool. The problem becomes how the agents must appropriate their investments to achieve an optimum return. Agents must cooperate to some degree in order to achieve a maximally beneficial outcome for the individual and the group.

In this game we are studying a co-evolved genetic programming method for creating investment strategies in a common pool resource (CPR). We wish to compare the performance of the strategies generated and the group level behaviour of the agents with that of the human players performance in a similar game [6]. This game provides a difficult problem for the agents and a difficult search space for the genetic programming process. In the agent environments specified, the maximum return from investments is gained by using the CPR, however over-investment ruins the output of the CPR for the group.

For these experiments a fixed number of agents play an iterated appropriation game. Each agent is provided with an endowment of tokens at the beginning of

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 167–174, 2011.

each round which must be invested in two markets. The first market offers a fixed return on the tokens invested, while the second, the CPR, provides a level of return based on the amount of total investment by all the agents in that market. The GP generated tree chooses the investment strategy for each agent every round and the fitness of the solution is a measure of the success of the agent.

The performance of these evolved trees are compared to the human players in order to discover: (1) if the evolved solutions are close to the aggregate group level Nash equilibrium as predicted by non-cooperative game theory, (2) if the individual play by the agents approaches Nash equilibrium by selective pressure and (3) what decision making processes are adopted by the agents is while playing the game against various opponents.

The main contributions of this paper are as follows: (1) a genetic programming based study of this CPR game. GP can provide an evolutionary perspective on this problem and by using decision trees it can overcome the constraints in other evolutionary methods[7], (2) the use of co-evolution and discussion of the findings, (3) the study of strategies employed by agents and (4) a comparison of the evolutionary results with human players.

The paper is divided in several sections. The next section is a discussion of the background and related research, which introduces briefly the concepts seen in the rest of the report. Section three introduces the CPR appropriation game and our GP approach. Section four introduces the experiments that we have conducted and section five contains the results. The final section discusses the uses for the techniques in the paper as well as possible future work.

2 Related Research

There is a great deal of literature available on specific field based CPRs such as the fishery [4] and for a review of the literature see [3]. We focus on studies that are more comparable to the computer agent simulations we employ. In [6] CPR problems are discussed from the point of view of lab experiments with human players and a study of common-pool scenarios in the field. Their main focus from the experiments is the appropriation game with a view to discover how changes to the rules of the games alters the rate of cooperation. They find that both the ability to communicate and sanctioning in games increases the players' yield.

There have been a number of evolutionary approaches to different economic games in previous research. It has been shown that multi population co-evolution with genetic algorithms performs well when for searching for equilibria in simple standard games used in organisational theory[7]. All the games explored were one shot, simultaneous move games owing to a representational constraint in GAs. The authors suggest that other techniques, including GP, would provide the ability for conditional choice and thus exploration of more complex and repeated games.

The emergence of cooperation has been demonstrated in a spatial CPR game where the spatial element is represented by agents on a circle[5]. They show that in the CPR game a cooperative strategy can survive, even when the majority of agents is defecting. As the strategies of players were based on neighbors it allowed some agents to maintain cooperation even when globally defection was the generally adopted strategy.

A model using the Swarm multi-agent simulation environment was built[1] to capture how adaptive agents perform in the baseline CPR game as defined in [6]. The agents are randomly allotted a subset of 16 preprogrammed strategies for which the performance results of each alternate strategy is measured as the average return that the agent would have received in each round had it utilized it. It is found that the agents perform similarly to the humans in the lab experiments. These findings can be used for comparison with other approaches to the same game.

3 Game Definition

Researchers[6] developed a series of laboratory experiments utilising human subjects and investigate the correlation between the behaviours of the human players and the behaviours predicted by non-cooperative game theory.

The baseline experiment comprised the following parameters: eight human participants made finitely repeated investment decisions regarding an amount of tokens with which they were endowed at the beginning of each round. The tokens are then invested in either Market 1, offering a fixed return, or Market 2, the common pool offering a return based on the level of investment, or some combination of both. Participants know the number of other players, their own endowment, their own past actions, the aggregate past actions of others, the payoff per unit for output produced in both markets, the allocation rule for sharing Market 2 output, and the finite nature of the game's repetitions. Participants also know the mapping from investment decisions into net payoffs.

Table 1 shows the parameters for the two baseline experiments undertaken which are also used for the evolutionary experiments. The number of tokens predicted by Nash equilibrium is 8 for each agent, or 64 for the group when such comparisons are made, for both experiments.

Averaged across several experiments the results were as follows: the average net yield as a percentage of the maximum yield was in the 10 token game 37% and in the 25 token game -3%. The main conclusion from this baseline experiment was that even as users reach the equilibrium point, net yield decays toward 0 and rebounds as subjects alter their investment strategies. In low endowment settings, aggregate behaviour results tend toward Nash equilibrium. In the high endowment setting aggregate behaviour in early rounds is far from Nash euilibrium but does approach it in later rounds. At the individual decision level, however, behaviour is inconsistent with the Nash prediction. In [1] they report almost identical performance from their agents, in terms of efficiency, when compared to the human players.

Table 1. Human Experiments

	Type of Endowment	
	Low (10 tokens)	High (25 tokens)
Number of Subjects	8	8
Individual token endowment	10	25
Production function (x_i is the investments by player i)	$23(\sum x_i) - .25(\sum x_i)^2$	$23(\sum x_i) - .25(\sum x_i)^2$
Market 2 return/unit of output	$.01	$.01
Market 1 return/unit of output	$.05	$.05
Earnings/subject at group maximum	$.91	$1.65
Earnings/subject at Nash equilibrium	$.66	$1.40
Earnings/subject at zero rent	$.50	$1.25

3.1 Using GP to Co-evolve Agents in the Two Market Game:

We use the GP process to create a tree for each agent representing its investment strategy for the CPR. As all tokens must be invested each round, the remainder are put into the fixed market. The GP creates trees from the set of functions and terminals contained in Table 1 which are divided into different *nodesets* for the STGP (for determining creation and crossover points for the trees). The functions are listed in Table 2 denoted with an $f(n)$ where n is the number of arguments the function takes.

The description of the GP nodesets is as follows: The environment nodeset contains functions and terminals that represent the agent's perception of the environment. Where the node says that it is believed, for this game that belief is with 100% certainty. The function BELIEVEDPROFITFROMM1 returns the amount the agent would receive based on the level of input as a given argument. BELIEVEDPROFITFROMM2 acts the same but also requires the total group investment as an argument.

The constant nodeset contains terminals in the form of constants. It also contains some standard mathematical functions which take two arguments to be operated on (multiply, divide). It also contains the *If* function which takes 3 arguments, the first from the decision set, the second is the true branch and the third is the false branch.

The decision set has functions which take environmental or constant nodes and compare them (greater, less, equal). We have also included *And* and *Or* which take other decisions. This allows for complex if then statements to be created.

We use a co-evolution method for evaluating the GP trees. We iterate through the population 20 times selecting an individual member and choosing a random seven others to create a group of eight agents – with each tree representing the investment strategy of each agent. As the opponent strategies are chosen at random each member of the population usually plays a far higher number than this minimum 20. The group then plays the game as the humans would with one difference – a fixed 20 rounds instead of at least 20 but the agents do not know the exact finite nature of the game.

For each round, the tree is evaluated returning a value of tokens which the agent will invest into the CPR. The remainder of the tokens are invested into the fixed market. The fitness of the individual is the cumulative profit of the agent over the 20 investment rounds with a small penalty for length of the tree. We also penalise invalid trees by assigning a very poor fitness score. An invalid tree is one that returns a value of tokens to invest in the CPR greater than the round endowment or less than zero. When a strategy like this occurs during evalutaion, it is replaced by a one investing a random percentage of the tokens endowments each round. The GP parameters that we use, based on experimentation and recommendation, are as follows: population size of 250, 100 generations, crossover probability of 90%, a mutation rate of 10%, a creation probability of 2% and finally elitism is employed.

Table 2. Node Sets for GP

Environment	Constants	Decisions
BELIEVEDROUND	Multiply $f(2)$	Or $f(2)$
BELIEVEDTOTALGROUPTOTKENS	Mod $f(2)$	Greater $f(2)$
BELIEVEDAMOUNTOFAGENTS	Plus $f(2)$	Less $f(2)$
BELIEVEDROUNDENDOWMENT	Divide $f(2)$	Equal $f(2)$
PROFITLASTROUND	If $f(3)$	And $f(2)$
PROFITM1LASTROUND	$\{0, 1, 2, \ldots\}$	
PROFITM2LASTROUND		
TOTALGROUPINVESTMENTM2LASTROUND		
INVESTEDM2		
CUMULATIVEPROFIT		
AVERAGEINVESTEDM2		
AVERAGECUMULATIVEPROFIT		
AVERAGETOTALGROUPINVESTMENTM2		
AVERAGEPROFITEACHROUND		
AVERAGEPROFITM1EACHROUND		
AVERAGEPROFITM2EACHROUND		
BELIEVEDPROFITFROMM1 f(1)		
BELIEVEDPROFITFROMM2 f(2)		

4 Experiments

We evolve 8 separate populations of solutions. We then select the tree with the best fitness at generation 100 to be a representative of that evolutionary run. We play each of the 8 selected strategies in a game against each other. As a benchmark they also play against a group of pre coded fixed strategies. We use the game the evolved agents play together as the comparison with the human players' game. Averaged over 3 iterations of this process we use the aggregate play as well as individual investments in each round as the basis of our comparison. We repeat the experiments for both 10 and 25 token endowments per agent per round.

We use a set of fixed naive strategies to assess the similarity in performance between the individual representative evolved strategies. The cumulative profit of each strategy should show the degree of variance between them and also how well each strategy performs. We, for an analysis, also save the best strategy each generation for the evolutionary runs. We analyse the population of 100 “best” strategies creating an overall profile of the evolutionary process in this problem domain.

We play each of the strategies against each other. We record the cumulative profits and rank the solutions accordingly to determine the if there is an increase in performance as the evolution time increases. We also analyse the playing of rounds to see if there is a general pattern that agents adopt.

5 Results

5.1 GP Co-evolved Agents

Using the tree with the best fitness at the end of 100 generations of the evolutionary run as a representative, we use eight of these candidates to play the game. We find that, regardless of the token amount, the chosen agents play close or equal to the Nash equilibrium amount of tokens.

Recall from Table 1 that the individual investment Nash equilibrium is 8 tokens and for the group is 64 tokens. In the three 10 token scenarios the agents play fixed strategies. The average group token investment for the games is 65.68. This is the same performance as the human players had at an aggregate level. At an individual level however we see a different trend. The evolutionary pressure is on the individual to perform at the Nash equilibrium and this is the result we see from our evolved agents; this deviates from what the human players played. The human performance show a changeable strategy that tends towards a group aggregate Nash in later rounds but at the individual level it is quite varied. The evolved agents play a typical non varying strategy of an average 8.21 tokens.

In the 25 token game, taking a candidate agent from eight different evolutionary runs, we see a perfect Nash play in the three chosen instances. During the game between the evolved agents, each agent plays an unchanging strategy of 8 tokens per round.

We can see from Fig. 1 that the evolutionary process converges to the Nash amount of tokens quite quickly. The standard deviation of the population's investment decreases for the run. The results are averaged over eight separate evolutions of the problem. This is evolution pattern is seen with both token endowment levels.

In order to further evaluate the co-evolved strategies we play them against a set of naive strategies that play 0%, 20%, 40%, 60%, 80% or 100% of the token endowments. These opponents have a uniform and unchanging strategy throughout. We perform this experiment to test if our evolved strategies contain the ability to exploit when there is low pool investment and to counteract over-investment.

While playing the naive fixed strategies the evolved 25 token agents played a mix of fixed and mild exploitative strategies. The fixed played 8 tokens or 8 in the first round and then 9 tokens in subsequent rounds. Some exploitative strategies gradually increased the amount of tokens from 8 to 10 when playing with the low investment naive strategies. Once the high investment naive strategies were in play all of the evolved agents did not reduce their usage of the pool and as such suffered net losses.

In the 10 token game the agents are not as badly exploited as they are in the 25 token game. This is due to the fact that the naive agents are bound to the maximum of 10 token investments so we do not see negative returns. We do observe an inability to change strategy as the pool is over exploited with agents playing roughly fixed strategies that invest either 8 or 9 tokens.

As the co-evolved population of agents converge to play the Nash amount of tokens, where during their evaluation a larger proportion of opponents will be playing similar strategies, the agents do not evolve very dynamic strategies that counteract being exploited. This results in very fixed strategies, as seen above, where typically the evolved agents do not take into account the actions of others.

5.2 Evolving against Naive Strategies

To explore the possibility of creating a good strategy we use the naive strategies in the evaluation of our agent trees. Each agent plays 6 different games with the naive strategies as described above. This is the first time that the evolved strategies different outcomes between the 10 and 25 tokens games. As the 10 token game does not get exploited when all tokens are invested by the 8 players the strategies learned are consistently close to or at a Nash level of investment.

For the 25 token games, the strategies evolved vary. Some play a uniform investment strategy while others play adaptive strategies. When the new evolved strategies play each other the performance is similar to human play. The pool utilization refines itself close to Nash at the aggregate level but from an individual level it is wildly varying. The best performing strategy is one that does not invest in the pool at all while the two next best are adaptive strategies. The worst performing strategies are ones that invest the total amount of tokens constantly. These findings are shown in Fig. 2. The average efficiency of the investments as a percentage of the maximum return is 53.5% which is much higher than that earned by human players.

6 Discussion and Future Work

We have shown that genetic programming techniques are suitable for generating strategies in a repeated investment problem. By using co-evolutionary methods we have shown that populations of strategies will quickly converge to Nash equilibrium, as predicted by game theoretic analysis, but also lose many adaptive behaviours. We have shown that genetic programming techniques are suitable for generating strategies in a repeated investment problem. By evolving against a set

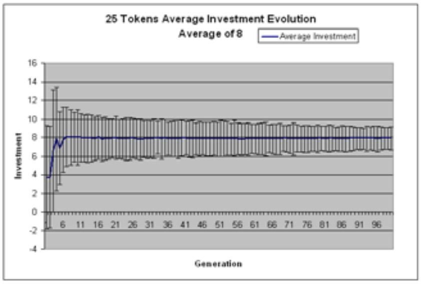


Fig. 1. Average Investment for 25 tokens

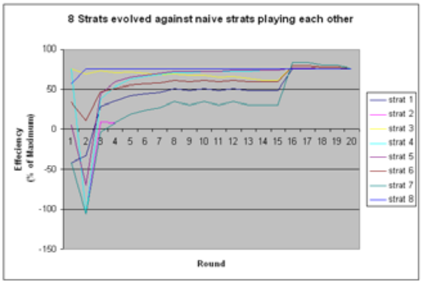


Fig. 2. Evolved Strats playing each other

of naive strategies we have shown the creation of diverse and adaptive behaviours that play similarly to humans. Erratic behaviours emerge when the goals of the opponents differ from the evolving agent; this supplies an evolutionary pressure towards more adaptive strategies.

We plan to explore and compare single and multiple population co-evolution for this problem. In conjunction with this we wish to examine the fitness to take into account varying proportions of the group score in order to ascertain if there is a tipping point in the agents' performance towards the optimum for the group. Using multiple populations would allow for specific roles to emerge to emerge within the population if the group composition is fixed.

References

1. Deadman, P.J.: Modelling individual behaviour and group performance in an intelligent agent-based simulation of the tragedy of the commons. J. Environ. Manage 56, 159–172 (1999)
2. Ferber, J.: Multi-Agent System: An Introduction to Distributed Artificial Intelligence. Addison Wesley Longman, Harlow (1999)
3. Furubotn, E.G., Pejovich, S.: Property Rights and Economic Theory: A Survey of Recent Literature. Journal of Economic Literature 10(4), 1137–1162 (1972)
4. Gordon, S.: The Economic Theory of a Common-Property Resource: The Fishery. J. Polit. Econ. 62, 124–142 (1954)
5. Noailly, J., Withagen, C.A., van den Bergh, J.C.J.M.: Spatial Evolution of Social Norms in a Common-Pool Resource Game. Environmental and Resource Economics 36(1), 113–141 (2007)
6. Ostrom, E., Gardner, R., Walker, J.: Rules, Games and Common-Pool Problems. Michigan, Ann Arbor (1994)
7. Price, T.C.: Using co-evolutionary programming to simulate strategic behaviour in markets. Journal of Evolutionary Economics 7(3), 219–254 (2007)

Using Pareto Front for a Consensus Building, Human Based, Genetic Algorithm

Pietro Speroni di Fenizio[1] and Chris Anderson[2]

[1] Engenheria Informatica, Coimbra University,
Coimbra, Portugal
speroni@dei.uc.pt
[2] Turnfront
chrisanderson@turnfront.com

Abstract. We present a decision making procedure, for a problem where no solution is known a priori. The decision making procedure is a human powered genetic algorithm that uses human beings to produce variations and evaluation of the partial solution proposed. Following [1] we then pick the pareto front of the proposed partial solutions proposed, eliminating the dominated ones. We then feed back the partial results to the human beings, asking them to find a alternative proposals, that integrate and synthesize the solutions in the pareto front. The algorithm is right now being implemented, and some preliminary results are being presented. Some possible variations on the algorithm, and some limits of it, are also discussed.

1 Introduction

Decision making is a common challenge in any community, independently of its size. Every form of government is essentially a way to solve this problem, but it is also solved (or suffered) in groups that are too small to have any form of official decision making structure. Often the way to find an agreement is through a democratic voting system. In this way the options are listed, and everybody votes what is their favourite option. Theory of voting is, generally, considered part of game theory, and it has a long history. For a good review we suggest [2]. Mostly the difference between the various voting systems are limited to (1) how many options can a member of the community endorse; (2) if the option chosen are ranked or not, and (3) how are the voting of the various member integrated to acknowledge what would be the winning proposal. The general structure is then: (a) the possible options are spelled out (often by some member of the community); (b) everybody votes; (c) through an algorithm the votes are counted; (d) the winning proposal is interpreted as the representative of the community global desire.

Of course unless the decision was taking through an unanimity, not everybody will have favoured the winning proposal, and thus some people of the community will be forced to accept a decision they do not favour. If this happens often, with the same people forcing over the same minority some decisions, it is common to talk about a tyranny of the majority. It is also well known in voting theory

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 175–182, 2011.

(and practice) that requiring an unanimity before letting a decision be accepted tend to freeze a community, when its size grows too much. How much is "too much", is different from community to community, but often between 10 and 20 members are enough to freeze a community who is trying to achieve unanimity. Is there really no other option?

2 The Unspoken Assumptions

There are a number of assumptions that are at the base of all those voting systems. We shall try to list them, and propose an alternative decisions system not based on them. Firstly there are some assumptions being made about the size of the space of the possible actions that can be undertaken by a community. It is generally assumed that those possibilities are few (first assumption), are clear (second assumption), and can be easily recognized (third assumption) and acknowledged (fourth assumption). By assuming that there are few possibilities, we also assume that it is possible to list them all (fifth assumption).

If we break free from those assumption, we can instead suppose that the space of possibility is vast, needs to be explored, and no single human being can see all the possible solutions. We can even assume that at the starting point no group of people can see all the possible solutions. Now the problem have moved from finding an algorithm that decides which of the few possibilities to implement, based on how many people prefer which of them, to finding a solution that fits as many persons as possible.

To say that generally the problem is simply posed as listing the possibilities, and the voting on them, ignores the important aspect of mediation. Some, generally few, individuals, when posed with different positions will try to mediate among the various possibilities, trying to find a solution that fits more people. Not only those people are rare, but also communities need to have one of those people in a particular position of power, to be able to benefit of its ability. So we could say that in a community, where everybody can vote, there are generally only few individuals that set up the possible proposals on which everybody else will vote, and even less that are actively working to find a mediation between the various groups. Those mediators are the ones which are effectively looking for possible solutions that can be endorsed by a wider base.

What we want to suggest here is a system where every member of a community has the opportunity to present proposals, mediate between existing proposals, endorse others proposals, and finally where this happens in a cyclic way in such a way that eventually an optimal solution is eventually reached.

3 The Algorithm

The algorithm that we are proposing in this paper, is a human based genetic algorithm [3], that explores the space of possible solution to the question posed. Looking for a solution that can satisfy the maximum number of people; potentially satisfying everybody.

The algorithm starts with a question being posed. It can be posed by one of the user, or it can be posed by an external person. Then every user will be allowed to write a possible solution (called proposal) to the question. At this stage the proposals are secret, and no user is allowed to see the proposals written by the others participants. When everybody has written their proposal, the writing phase end, and the algorithm moves to the endorsing phase. Now everybody is allowed to read each other proposal, and endorse all the proposals they agree with. It is important in this stage that each user endorse *all* of the proposals that he agrees on. No limit should be set on the number of proposals that a participant can endorse. In the worse case a participant will only endorse their own proposal. If a participant is instead satisfied with all the proposals he can endorse them all. Once everybody has endorsed all the proposals they are happy with, a selection process happen.

We define a proposal A as dominating a proposal B, if the set of participants that endorse proposal A strictly contains the set of participants that endorse proposal B. Of course if A dominates B, and B dominates C, then A dominates C.

To select the winner proposals we eliminate all the proposal that are being dominated by any other proposal. What remains is a Pareto Front of the proposals. Note that each participant will be present in at least one proposal. As such the Pareto Front can be said to represent every person that has participated, so far. As the selection ends, we say that also a generation (or a turn) has passed.

If the selection process produced a single proposal, this must necessarily be endorsed by everybody. We then decide that the question has been answered, and an acceptable unanimous solution has been found. If the selection process did not produce a single proposal, the process continues with a new generation.

Now at this new generation the participants are presented with the question that was posed (the same as before), and the pareto front of the proposals that won the past round. All the other proposals from the past generation have been eliminated, and will be ignored. The participants are now invited to write new proposals, taking the previous Pareto Front as an inspiration. They should try to find possible synthesis among them. Although this is the invitation that they are presented each participant is allowed to write anything they want. They can introduce new solutions, re-propose past solutions that did not make it to the Pareto Front, rephrase past proposals. If a participant feels that he has nothing to contribute, and that his view is fully represented by the proposals that made it to the Pareto Front, he is also allowed not to contribute at all at this stage. Once the participants (who wanted to write) have written their proposals, again we move to the endorsing phase, and then to the selection stage, and so on.

The process continues through writing, and endorsing phase, until the system has either converged to a single unanimous answer, or the system does not seem to produce any more variations, and generations after generations the Pareto Front is always the same.

The system is very simple, it is a genetic algorithm that uses human beings to produce the 'genetic' variation. It also uses human beings to evaluate the generated partial solutions. And then then limits itself to the pareto front of the proposals.

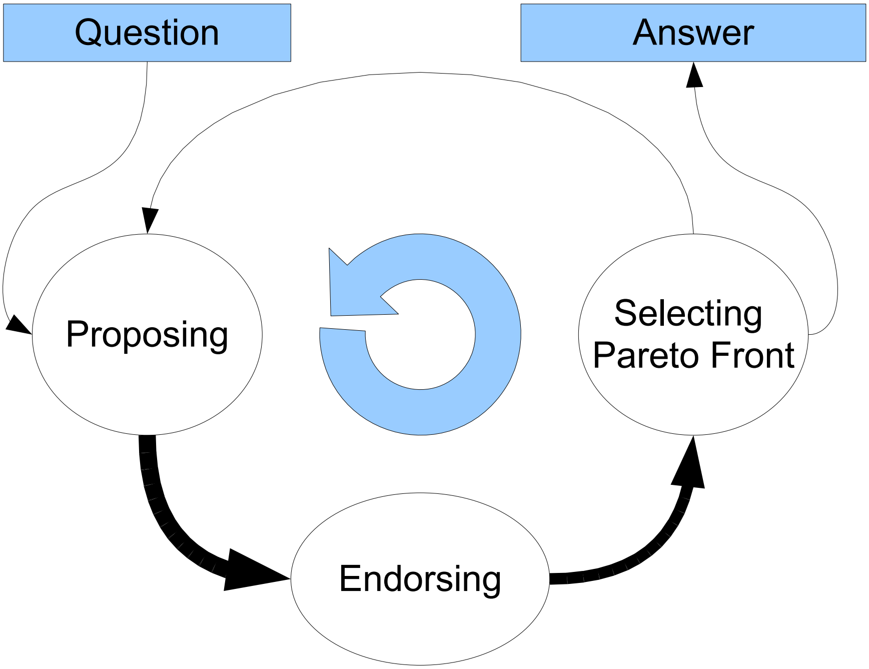


Fig. 1. A schematic representation of the genetic algorithm

We have already discussed in the introduction, how we moved from framing the problem as finding the best solution among a given set, to find the best solution, in an open context. Once we have framed the problem as a search problem, using a genetic algorithm is a natural choice.

What we are looking for is a solution which is expressed as a text describing a set of actions. If we were trying to produce such algorithm automatically we would incur in three different problems. Each unsolved so far. First of all there is no automatic way to translate a text in a set of actions. Solving this problem would be equivalent to solve the general problem of automatically finding the semantic meaning of a text. Although a lot of research has been done so far, the results generally apply only for very specific cases. The second problem is to select and filter the actions that are meaningful and possible, from the ones that do not simply make any sense. And this is a second unsolved problem, which would instead require a computer program to have an understanding of the physical world, and its laws. And finally the genetic algorithm would have to be able to evaluate which solutions are better than others. None of those problems have been solved so far.

Similarly to [4] we decided to let human beings produce the possible solutions, and again human beings evaluate them. Evaluating the proposals, selecting which to pick for the next generation is in itself not a straightforward task. There are here three requirements that need to be satisfied: (1) we want the algorithm to converge in few solutions, that eventually (through various generations) give rise to a single answer; (2) we want the set of solutions to represent the whole panorama of possibilities, and (3) we want every participant to feel that the solutions they endorse are part of those panorama.

In the literature there are different threads of research that studied this problem. If we look at this as a voting problem, there is the whole voting theory that has been developed. On the other hand, if we look at this as a genetic algorithm problem, there is also a huge literature on the subject, too.

As a voting problem the assumption has always been that each voters had an equally valid point of view, and so it was natural and fair to just sum them. But basically each voter represents a different dimension, on which each of the proposals would be evaluated. If we sum the votes, we are evaluating a bunch of points in a unidimensional space. And we just need to decide where to stop. How many proposals should be allowed to the next generation. At this point the proposals are also playing a transitive game. If proposal A, wins over proposal B (i.e. A has more votes than B), which wins over proposal C, then A wins over C. But we are effectively throwing away a lot of information. In particular we are ignoring who has voted for what proposal. So the result is suboptimal, in the sense that many participants are not really considered in the result. If instead we consider this information, the problem moves from a transitive game to an intransitive game; now a proposal A can be better than a proposal B (according to Joe), which can be better than the proposal C (according to Suzanne), which can again be better than the proposal A (according to James). In this case no proposal is then inherently better, and we are faced with what is called an intransitive cycle. In the field of genetic algorithm Bucci and Pollack [1] have shown that it is possible to escape those intransitive cycles by using Pareto Fronts. Not only this moves back from the potentially dangerous place of intransitive cycles, back to a transitive games; now a proposal A dominates a proposal B if and only if A is bigger or equal in all dimensions to B (no participant prefers B to A), and there is at least a dimension in which A is strictly bigger than B (there is one person that prefers A to B). So if A dominates B, and B dominates C then A will dominate C. This effectively permits us to drop all the proposals that are being dominated by another one. What remains is what is called a Pareto Front. While we are doing this we are not losing the representation of any participant: if proposal A dominates B, and we drop B, all the participant that have endorsed B will have endorsed also A. So by ignoring B we are not ignoring any participant input. They are all present in A. Now assuming that each person have endorsed at least one proposal (which is a safe assumption, since at the very least they would have endorsed the proposal written by them), then each person will be present in the Pareto Front.

This solution also solved the problem of how many proposals to take-on to the next generation. We need to take all and only the proposals of the Pareto Front. If we took less, the solution would not be inclusive, and we would run the risk of representing all the participants. If we took more, we would be keeping a solution which is unnecessarily redundant.

4 Limits, Problems and Variations

While the basic algorithm is quite simple, there are a number of possible variations that should be tested. Some of those variations are presented here.

Anonymity of the proposals. Should proposals be anonymous, or should the people who are endorsing the proposals be allowed to know who have written a particular proposal. If the proposals are anonymous the participants are forced to read the whole proposal, before endorsing it. Will they be able to understand it? This depends on the topic, and on the group. The philosophy behind this choice requires each proposal to stand on its own too legs, and not be endorsed thanks to the popularity of the person writing it. On the other side, if the proposals are non anonymous, people who are interested in the topic, but lack the technical knowledge to understand the subtle elements of it, could still participate in the vote. In system where the key element is a comparison between set of voters, and not the actual counting of the endorsers, the anonymous choice seems to be a natural one.

Anonymity of the endorsements. After the endorsing phase, the successful Pareto Front of the proposals are fed back to the participants, asking them to write new proposals. When this is done the name of the participants that have endorse each proposal can also be made public, or not. By making those information public it permits to the participants to understand which are the major proposals, also it works as a light social control system, to avoid participants abusing of the system in an antisocial way. In this system each user has a strong power. If a participant writes a proposal, and then endorses only his own proposal, he can be sure that his proposal will be present in the pareto front. This behavior is possible, and even socially acceptable when a person is honestly in disagreement with all the proposals that are being presented. But it can be abused by using it as a way to protest. By letting everybody see who has voted for what, this kind of antisocial behavior is exposed, and generally ceases. On the other hand an anonymous system would permit everybody to endorse what they truly believe in. So in this case both of the possibilities make sense, and both should be tested.

Who is allowed to write, who is allowed to endorse? In our description we assumed that everybody who was allowed to write a proposal was allowed to endorse them. This does not necessarily need to be so. If the participants who are allowed to propose are a subset of the participants which are allowed to endorse we have a situation who is similar to a modern democracy, where few people define the options for everybody else. We are not particularly interested in this situation, as it has already been tested enough in modern democracies. If instead the participants who are allowed to suggest proposals are a supra-set than the set of the endorsers, then we have a situation where a community are discussing an issue, and external people are allowed to insert new ideas. This situation has rarely been tested. Another, different, possibility is a situation where no one who has proposed something is allowed to participate in the endorsing phase. If this is done by splitting the group into two subgroups at the beginning, and keeping every proposal anonymous. This last option might produce interesting results.

Changing the Participants During the Process. So far we have assumed that the same participants that have written a proposal one generation, will also write it on the next generation. And somehow this would probably be an optimal situation, because, since the people participating are also the ones evaluating the results, this defines a static fitness landscape on which the genetic algorithm can climb. Unfortunately this is not always possible. Since this decision system requires multiple voting, periods, and in general a protracted interaction between the users and the system, it is possible (and even common) that the participants in a generation (or even between phases, inside a generation) might change. If the community is big enough this is not necessarily a problem, provided there are enough participants to represent the various possible ideas, the system can keep on finding an answer. If the community is small, changing the participants half way seems to produce the most unreliable results. As it would be running a genetic algorithm where the fitness function changes from one generation to the next. Unfortunately most of the test that we could do far of this method suffered of this problem.

Real Questions versus False Questions. Although the work is still preliminary, we already noticed an interesting pattern. We tried the algorithm several times, on various questions. Every time the question was a real question, among real participants, which were going to have their life changed by the result, the algorithm seemed to work better. It would act in a more predictable way, it would converge more rapidly. When more than one result was in the Pareto Front, the participants would try harder to synthesize an acceptable compromise. When instead the question was irrelevant, the answer were random, the endorsing was random, and the algorithm did not seem to converge easily (if it would converge at all). All this seem to suggest that the algorithm is indeed exploring a space of possibilities. And when the question is a real question, there is a definite fitness space to be explored. With peaks, valleys, and neutral ridges. When instead the question is irrelevant (to the participants), the algorithm is unable to find any real synthesis because no real synthesis is there to be found. This suggests that future work should be done on participants that are really involved with the results of the procedure.

5 Partial Results and Conclusions

At the moment we only did preliminaries studies on the subject. We tried it out among six participants with pen and paper. We then implemented the algorithm on a website (http://pareto.ironfire.org), and invited some testers to try it out. All the results are promising, but not consistent enough to make any statistical case. We will thus only rely them as an anticipation of some future work. On the pen and paper example, the question posed was: "We are going for one month together, in vacation, this summer. What shall we do?". This test only lasted two generations. On the first generation the answer were: "go camping"; "help my grandmother with her garden" (from participant 'D.'); "join a construction site and build a house"; "go biking in east Europe"; "go to thailand"; "go to Canada". After the evaluation process the Pareto Front only included two proposals left: "go to Canada" and "help my grandmother with her garden". Then the participants were invited to write new proposals. Five out of six proposals suggested to "go to Canada with D.'s Grandmother, and ...". The sixth proposed to pay a gardener for D.'s Grandmother, and then go to Canada. We notice here an interesting result. The proposals seem to get more complex as the generation passes. As if the algorithm started by exploring the space of possibilities, in a more general way, and then become more precise in successive generations. Everybody is effectively trying to mediate between the elements. It was also interesting that each person tried to reinsert what they really cared for, in the next proposal. For example the participant that first suggested to "go biking in east Europe", on the second generation suggested to "go to Canada with D.'s grandmother, and go biking, after leaving D's grandmother in a camping site."

A decision making algorithm was presented to permit to a community to investigate and discover the most widely endorsed proposal that answers a given question. A number of possible variations were discussed and some partial results were presented. Future work include testing with bigger communities, for longer time, as well as testing the effect of the possible variations.

Acknowledgment

We thank for the interesting discussions: Giovani Spagnolo, Manuel Barkau, Christopher Ritter and the people who participated in our tests.

References

1. Bucci, A., Pollack, J.B.: A mathematical framework for the study of coevolution. In: Foundations of Genetic Algorithms, vol. 7, pp. 221–235. Morgan Kaufmann, San Francisco (2003)
2. Taylor, A.D., Pacelli, A.M.: Mathematics and politics: Strategy, voting, power and proof. Springer, pp. 1–377 (2008)
3. Kosorukoff, A.: Human based genetic algorithm. In: 2001 IEEE International Conference on Systems (January 2001)
4. Defaweux, A., Grosche, T., Karapatsiou, M., Moraglio, A., Shenfield, A.: Automated concept evolution. Technical Report Vrije Universiteit Brussel, Belgium (2003)

The Universal Constructor in the DigiHive Environment

Rafał Sienkiewicz and Wojciech Jędruch

Gdansk University of Technology, Gdansk, Poland
{Rafal.Sienkiewicz,Wojciech.Jedruch}@eti.pg.gda.pl

Abstract. The paper describes an universal constructor model realized in artificial environment called DigiHive. The environment is a two dimensional space, containing stacks of hexagonal tiles being able to moving, colliding, and making bonds between them. On the higher level of organization a structure of tiles specifies some function whose execution affects other tiles in its neighborhood. After short description of the DigiHive the paper describes design of an universal constructor and discusses possibilities of simulating self-replicating strategies in the environment.

1 Introduction

The individual and agent based modeling are natural tools for modeling of basic biological phenomena. Using this approach an universal or problem oriented modeling environments (artificial worlds) can be constructed. The DigiHive environment [1,2] (see also precursor of the environment [3,4]) is an artificial world aimed for modeling of various systems which a complex global behavior emerge as a result of many simultaneous, distributed in space, local and simple interactions between its constituent entities. It is especially convenient for modeling of basic properties of self organizing, self modifying and self replicating systems. laws The short description of the DigiHive system is given in the first part of the paper.

The various strategies of self replication usually involves existence of an universal constructor, a device which can build any entity basing on its description and using surrounding building materials. The design of the universal constructor in the DigiHive environment is described in the second part of the paper. The design process must still take into account the fact that all entities in the environment are in continuous movement. The design constitues the base for testing a series of modeling of various self reproducing strategies: their efficiency, speed, and ability to evolve.

2 The DigiHive Environment

The environment is a two dimensional space containing objects which move, collide and change their structure. The constituent objects of the environment, called *particles*, are represented by hexagonal tiles. The particles are of 256 types

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 183–190, 2011.

and are characterized by velocity, position, and internal energy. Each type is related with a set of constant attributes: mass, bond energy (needed to disrupt bond between particles of given types), activation energy (needed to initiate any change of bonds).

Particles can bond together forming a *complex* of particles. The permanence of the complex depends on its constituent particles bond energies. Particles can bond vertically on the directions up and down forming stacks. The particles on the bottom of the stack can bond horizontally. An example of stack of particles and a complex formed by horizontal bonds are shown in Fig. 1.

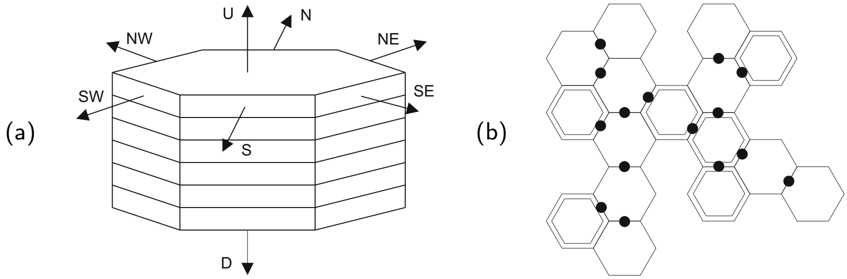


Fig. 1. Examples of complexes: (a) horizontal view of single stack of particles with directions shown, and (b) vertical view of complex formed by horizontal bonds, where hexagons drawn by single lines represent single particles, and by double lines represent stacks of particles and black dots mark horizontal bonds between particles

There are two types of collisions of particles and complexes: elastic and inelastic one, where the type is chosen randomly according to preset probability. During elastic collision, the resulting velocities are calculated according to the rules of classical mechanics resolving the collision of two disks (circles surrounding hexagons) – conservation of momentum and of kinetic energy is observed.

After inelastic collision the velocities of both particles are equalized – conservation of momentum is observed but the resulting decrease of kinetic energy of particles is compensated by emission of a *photon*. Photons move and collide with particles (but not with other photons). The collisions result in creating or removing bond between hit particle and other particle close to the hit one.

When the probability of inelastic collision is set to 0 the above described properties allow to perform simulation of classical mechanics processes – equivalence of a simple molecular dynamic. When the probability is set to non zero value, some photons will appear as a result of collisions and then the reactions triggered by photons will lead to creation of various complexes of particles.

2.1 Functions

Besides the reactions resulting from the collision of particles with photons, there also exists additional class of interactions in which complexes of particles are capable to recognize and manipulate particular structures of particles in the space around them. The description of the function performed by a complex is contained in the types and locations of particles in the complex. The structure of the complex is interpreted as a program written in a specially defined declarative language. Syntax of the program encoded in a complex of particles is similar to the Prolog language, using the following predicates only: program, search, action, structure, exists, bind, unbind, move, not. Predicates: program, search, action and structure helps maintain the structure of the program. The other predicates are responsible for selective recognition of the particular structure of particles (exists) and for manipulation of them (bind – create bonds, unbind – remove bonds and move – move particles).

```
program():–
              search(), action().
search():–
              structure(0).
structure(0):–
              exists([0,0,0,0,0,0,×,×], mark V1),
              exists([1,1,1,1,1,1,1,1] bound to V1 on N, mark V2),
              exists([0,0,0,0,0,0,0,0], mark V5),
              not(structure(1)),
              not(structure(2)).
structure(1):–
              exists([1,1,1,1,0,0,0,0] bound to V2 on NW, mark V3),
              exists([1,1,1,1,0,0,0,0] bound to V3 on SW, mark V4),
              not(structure(3)).
structure(3):–
              exists([0,0,0,0,1,1,1,1] bound to V4 on S).
structure(2):–
              exists([1,0,1,0,1,0,1,0]).
action():–
              bind(V2 to V5 on SW).
```

Fig. 2. Example of a program recognizing the structure shown in Fig. 3

An example of a program is presented in Fig. 2. The program recognizes the structure shown in Fig. 3, and then binds the particle 11111111 to the unbound particle 00000000.

The predicate program consists of exactly two predicates search and action. First one calls the searching predicates, while the second one calls the predicates responsible for performing some actions in the environment.

The predicate search calls the predicate structure. The predicate structure consists of sequence of exists predicates and/or other structure predicates, always followed by the negation not. It provides the ability of recognizing the particular

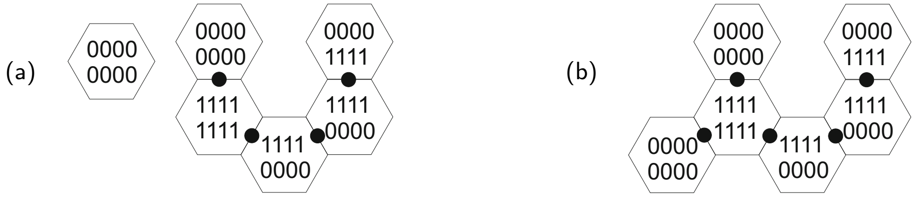


Fig. 3. Single particle and a complex of particles recognized by the program listed in Fig. 2 (a), and the structure after action of the program (b)

structure in case of some other structure does not exists. In the example, the particle of type 10101010 plays role of the reaction inhibitor (its existence would prevents program from being executed).

With the predicate exists it is possible to check various conditions, e.g.: existence of some particular particle type (exists([0,0,0,0,1,1,1,1] ...)), check if the particle is bound to some other particle on given direction (exists(... bound to V2 on N ...)) etc. It is also possible to mark the particle which fulfills the predicate condition with one of 15 labels (variables: V1 to V15), e.g. the exists([1,1,1,1,0,0,0,0] bound to V2 on NW, mark V3) means: find the particle of type 11110000 which is bound to the particle marked as V2 on direction NW, and store the result in variable V3 (mark the found particle as V3).

In fact, the only result of search predicate is some state of the variables V1 to V15 in which there are stored particles fulfilling the conditions grouped in all structure predicates. While the searching is performed via Prolog like backtracking algorithm, the order of execution inside this predicate doesn't matter (besides performance impact).

The predicates grouped in the action predicate are sequence of predicates that affect the environment. Contrary to the previous one, the order of execution is important. As a matter of facts this part of the program acts as an imperative (not declarative) sequence of commands which operates on data provided by declarative search part.

The important feature of the particle complex language is the property that small changes in code of a program (i.e. in particles in which the program is encoded) usually lead to small changes in its execution effects. Such a property of the language is crucial while simulation of self organizing phenomena. The simulation that demonstrates this property was presented in [2].

The program is encoded by complex of particles. Each predicate structure is represented by the single stack of particles. Such a stack encodes a list of predicates exists. Stack which encodes structure(0) also encodes predicates bind, unbind and/or move. Adjoining stacks encode negative form of predicates structure. More details can be found at project website [1].

The introduced property of the environment that the complexes of particles perform some functions on other particles and a moment later are an object of manipulation by another complex opens the wide possibilities of modeling variety of self modifications systems and regulation chains.

3 The Universal Constructor

The universal constructor is a concept introduced by von Neumann in his famous work on self-replicating cellular automata [5] (see also [6] for most recent ideas).

The universal constructor in DigiHive environment is a structure A (complex of particles) being able to constructs other structure X based on its description $d(X)$. It is admissible to constructs the X structure via description $d(X')$ of an intermediate structure $X' \neq X$, being able to transform itself into the X.

The universal constructor is the consistent structure (set of programs) being able to fulfill the following tasks:

1. search for valid information structure (information string) – $d(X)$. The information string encodes description of some structure $d(X)$. The description may be viewed as another program written in simply universal constructor language with the following commands: `PUT` (add specified particle to the stack), `SPLIT` (start construction of a new stack connected horizontally to the existing one), `NEW` (start construction of a new stack and not connect it to the existing one), and `END` (end of the information string). The information string is a stack of particles of specified types as described in the Fig. 4a. As an example the following program:

   ```
   PUT(01010101) PUT(01010101) END
   ```

   which describes stack of two particles of type `01010101` can be encoded by the stack:

   ```
   11111111
   01010101
   00000001
   01010101
   00000001
   ```

2. connect itself to the found information string and start constructing the structure X. The structure X consists of horizontally joined stacks of particles. There is always exactly one stack of particles being build at the moment, called active stack $X^\star$,
3. sequentially process the joined information string:
   (a) if current particle in the information string encodes command `PUT` – find the particle of *specified* type which is on the top of stack of building material. The building material is contained in stacks of particles (named M) marked by the particle at the bottom (material header) of the type `0000××10` (see Fig.4b). The newly found particle is removed from the top of the stack M, and put on the top of the stack $X^\star$ (Fig. 5a).
   (b) if current particle in the information string encodes command `SPLIT` – split the stack $X^\star$ into two stacks: remove particle from top, move the trimmed stack in the *specified* direction, and create horizontal bond between $X^\star$ and removed particle. The particle becomes the bottom of active stack $X^\star$. This action is presented in the figure 5b,

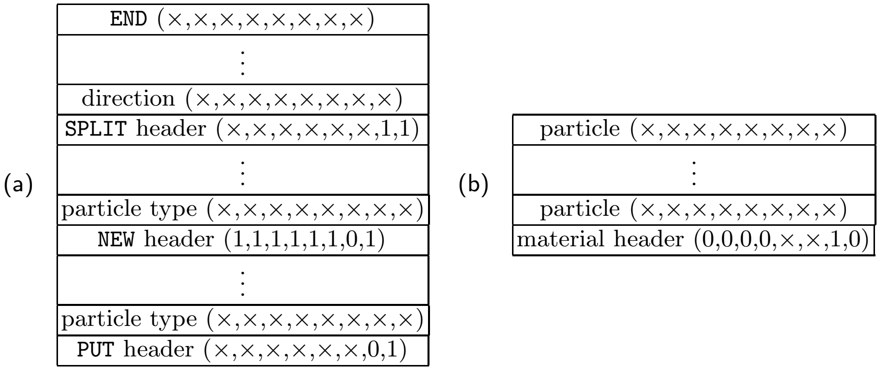

(a)
END (×,×,×,×,×,×,×,×)
⋮
direction (×,×,×,×,×,×,×,×)
SPLIT header (×,×,×,×,×,×,1,1)
⋮
particle type (×,×,×,×,×,×,×,×)
NEW header (1,1,1,1,1,1,0,1)
⋮
particle type (×,×,×,×,×,×,×,×)
PUT header (×,×,×,×,×,×,0,1)
(b)
particle (×,×,×,×,×,×,×,×)
⋮
particle (×,×,×,×,×,×,×,×)
material header (0,0,0,0,×,×,1,0)

Fig. 4. Encoding stack (a) and building material (b)

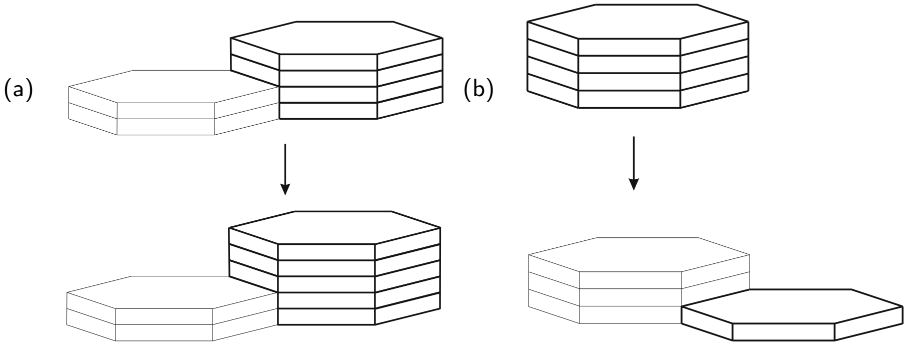


Fig. 5. Illustration of actions of the universal constructor during processing of the information string: (a) action caused by the command PUT. The particle of specified type is put on the top of active stack $X^\star$ (draw using thicker lines), (b) action caused by the command SPLIT. The particle is removed from top of $X^\star$, then the bond is created on specified direction, the particle becomes a new active stack $X^\star$.

(c) if current particle in the information string encodes command NEW – disconnect the structure X and immediately start the construction of a new structure with *specified* particle as the beginning of $X^\star$ (single information string can then encode various structures, e.g. both $d(B)$ and $d(A)$ – see also additional note on the use of the NEW command at the end of this paragraph),

(d) if current particle in the information string encodes command END – release the information string and the constructed structure.

The universal constructor was implemented as a set of cooperating 10 programs being able to fulfill tasks described above, enhanced with 5 helper stacks of particles. Helper stacks are used mainly for performing synchronization between working programs, they also mark some characteristic part of the universal constructor e.g. a place where the new structure is build, a place where the universal constructor join the encoding string etc.

In order to illustrate the universal constructor abilities it was provided with the information string describing flat, rhombus-shaped complex. As the result, the programmed structure was successfully build. The simulation screenshots are presented in the fig. 6.

Note, that after the complex is finished the constructor is immediately ready to connect to the same information string again and start a new translation. Such behavior would cause that constructor will produce all the time the copies of the same structure. This undesirable side effect can be resolved be encoding a simple program which can disable the information string – i.e. prevent it from being recognized by the constructor. The program can be encoded together with the main structure in one information string. During translation the program can be separated from the main structure by the NEW command. It is also possible to encode another program which can enable again the information string (e.g. at the beginning of the string in order to guarantee the existence of at least one program after the reaction). Note, that the efficiency of universal constructor reaction can be then adjusted by the concentration of both types of programs.

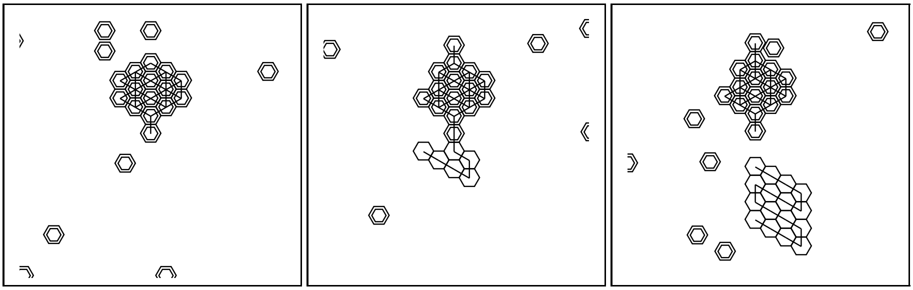


Fig. 6. Construction process after 0 (a), 26 (b) and 88 (c) simulation cycles

4 Strategies of Self-reproduction

During designing of the universal constructor, several technical problems has been encountered, solving which would need introduction of some subtle programming tricks leading to in some way to superfluous structure of the constructor.

For example in case when the constructor A works with its description $d(A)$, i.e. when constructor builds its own copy the problem is to prevent the activity of partially constructed structure. This structure should not manifest any activity before it is completely finished, in other case the simulation can become unpredictable (e.g. the unfinished constructor may start producing the copy of itself etc.). On the other hand the constructor that builds the structure should not recognize this structure as a part of itself. The another problem is a geometry of the constructed structure. Sequential adding of subsequent stacks in various direction may lead to the situation that a built structure will try to occupy a place already occupied by the constructor itself. Also sequential adding of stacks produces complexes only in the form of one dimensional strings of stacks.

Taking into consideration the tradeoff between the above problems and the simplicity and efficiency of solutions it has been assumed that the constructor may not be fully universal – some structures can not be built straightforwardly.

These limitations can be fully compensated using a wide possibilities offered by the actions of functions encoded in complexes. The problem of constructing any structure can be resolved in one of the following ways:

1. By constructing some intermediate structure I' being able to transform itself into the desired one: I, in finite number of time cycles:

$$A + d(I') \rightarrow A + d(I') + I' \rightarrow A + d(I') + \mathbf{I}$$

2. By constructing set of programs $I_1, I_2, \ldots, I_n$ being able to build the structure I in finite number of time cycles:

$$A + \sum_{i=1}^{n} d(I_i) \rightarrow A + \sum_{i=1}^{n} d(I_i) + \sum_{i=1}^{n} I_i \rightarrow A + \sum_{i=1}^{n} d(I_i) + \sum_{i=1}^{n} I_i + \mathbf{I}$$

5 Conclusions and Further Research

The universal constructor implemented in the DigiHive environment has been designed and implemented. The limitations of the design has been discussed and the ways of compensation has been shown.

The aim of the further research is to implement and compare the effectiveness of various possible self-reproduction strategies and their ability to evolve in time.

References

1. Sienkiewicz, R.: The DigHive website, `http://www.swarm.eti.pg.gda.pl/`
2. Sienkiewicz, R., Jedruch, W.: Artificial environment for simulation of emergent behaviour. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 386–393. Springer, Heidelberg (2007)
3. Jędruch, W., Barski, M.: Experiments with a universe for molecular modelling of biological processes. Biosystems 24(2), 99–117 (1990)
4. Jedruch, W., Sampson, J.R.: A universe for molecular modeling of self-replication. Biosystems 20(4), 329–340 (1987)
5. von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign (1966)
6. Hutton, T.J.: Evolvable self-reproducing cells in a two-dimensional artificial chemistry. Artificial Life 13(1), 11–30 (2007)

A Local Behavior Identification Algorithm for Generative Network Automata Configurations

Burak Özdemir and Hürevren Kılıç

Department of Computer Engineering,
Atılım University İncek, Gölbaşı, 06836,
Ankara, Turkey
bozdemir@gmail.com, hurevren@atilim.edu.tr
http://ceng.atilim.edu.tr

Abstract. Relation between the part and the whole is investigated in the context of complex discrete dynamical systems. For that purpose, an algorithm for local behavior identification from global data described as Generative Network Automata model configurations is developed. It is shown that one can devise a procedure to simulate finite GNA configurations via Automata Networks having static rule-space setting. In practice, the algorithm provides an automated approach to model construction and it can suitably be used in GNA based system modeling effort.

Keywords: Generative Network Automata, Automata Networks, Inverse Problem, Identification, Discrete Dynamical Systems.

1 Introduction

In the context of complex dynamical systems, one can talk about two general approaches to the study of behavior: The first one is to start with global information describing state and topology changes of the system and identify the local state transition/topology transformation functions that generate them (i.e. the *inverse* problem). The second approach is to start with some state transition/topology transformation functions at hand and observe the resulting behavior of the constructed system through local function execution (i.e. the *forward* problem) [1][2][3].

In this study, we consider the first problem in the context of Generative Network Automata (GNA) known to be a varying-topology extended modeling framework for complex dynamical systems [4]. GNA framework provides us a well-defined environment for examining potential relations and mechanisms between the part and the whole. Different from Automata Networks (AN) modeling in which locally interconnected large set of cells evolve at discrete time steps through mutual cell interactions [5], GNA modeling framework supports time varying connectivity among automata components that results in *autonomous* transformations of complex dynamical networks. To the best of authors knowledge, the first appearance of Automata Networks in literature is due to the works

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 191–199, 2011.

of John Von Neumann [6] in order to model various phenomena in physics and biology. When we consider discrete systems, one may encode any entity in bits. Here, the coded thing can be any value representing some property of the system. And, if the coded entity somehow changes its value over time, this makes the system suitable for being modeled by for example Cellular Automata [7] or in general by AN. In fact, the encoding can also be applied to GNA intercomponent connectivity information. So, one can devise a procedure to identify (then simulate) finite GNA configuration sequences via AN having static rule-space descriptions. This paper describes how the idea can be realized for some restricted type of discrete dynamical systems. The existence of proposed algorithm also proves (by construction) the generatability of seemingly *autonomous* behavior via *designed* one. Simply, the output of the algorithm is an instance of AN constituted by the identified *i*) set of encoded and coupled state/connectivity transition rules and *ii*) static AN topology (represented by what we call overlay level neighborhood graph) describing the structure and flow of state/connectivity information between AN components. In practice, the algorithm provides an automated approach to model construction and it can suitably be used in GNA based automated system modeling effort.

In section 2, we give some of the necessary definitions and assumptions made throughout the paper. The proposed algorithm is described in section 3. Section 4 is the conclusion part that includes discussions/speculations about applicability of the approach to different discrete dynamical network domains and possible future works. Throughout the text, we used the terms component and node interchangeably.

2 Definitions and Assumptions

GNA is a dynamical network described by a time-varying directed graph with its labeled nodes. Each node of the graph represents one system component and takes one of the possible state values defined by a state value set S . The edges are supposed to be the indicators of referential relationships between system components that cause state transitions and topology transformations in time. Below, we give definitions about GNA configuration and its temporal dynamics[4].

Definition 1: *A configuration of a GNA at time* t is a triplet $\langle V_t, C_t, L_t \rangle$ where

V_t : a finite set of nodes at time t describing dimension of the system,
C_t : a mapping $V_t \rightarrow S$ defining the global state of the system at time t,
L_t : a mapping $V_t \rightarrow V_t^*$ defining global topology of the system at time t and this mapping is defined by outgoing edges of a specific node to its destination nodes.

Definition 2: Let G be the set of all GNAs. *Temporal dynamics of a GNA* is defined by a triplet $\langle E, R, I \rangle$ where

E : a mechanism that produces an extraction $g_e \in G$ from a given GNA instance $g \in G$.

R : a mechanism that replaces the extracted subGNA g_e a with a new GNA instance $g_r \in G$ and produces correspondence $V_t^{g_e} \rightarrow V_t^{g_r}$ from nodes of g_e to nodes of g_r.

I : an initial configuration of GNA.

The extraction and replacement mechanisms can be deterministic or stochastic. The above extraction mechanism E, replacement mechanism R and initial configuration I are sufficient to define GNA based dynamic models and reflect a perspective (defined by $\langle E, R, I \rangle$ structure) to explain complex dynamic behavior of systems under study. Clearly, based on the amount of information provided by given global behavior data, there may be many GNA based models to be identified by an identification algorithm. This fact has also been pointed out in the context of reverse engineering of Automata Networks [7]. So, one need to define his/her assumptions about the characteristics of the system model under automated identification/construction for better precision. The assumptions/restrictions can be treated as apriori knowledge about the system under study that may vary from biological organisms, ecological communities and human societies to computer communication networks. Our assumptions include the following:

- Local behavior of system components are supposed to be *deterministic*. As a consequence, whole system behavior is supposed to be deterministic which is reflected by global behavior data under study.
- Components have *minimal memory*. In their update, components consider only one step previous state/connectivity of other system components (this may also include themselves).
- System evolves *synchronously* in terms of its component states and network connectivity updates.
- Global behavior of the system is supposed to be *cyclic*. In other words, the data reflecting last configuration is supposed to be followed by the first one, for completeness.
- We divide system into two parts: *observable* part and *reservoir* part (see Figure 1). The whole system is supposed to be *isolated* and its components are *distinguishable*. However, the observable part of the system from which global behavior data input is extracted is *non-isolated* while its components are still *distinguishable*.
- Components are supposed to appear at least once in the observable part of the system during their lifetime.
- All system components are assumed to be known apriori in the form of a finite ordered set H defining inter-component neighborhood relation based on ordering.
- Component behaviors are supposed to be explainable by state/connectivity information fed from *nearest neighbor* components defined by H.

- At any time a component can be in one of three states represented by 0, 1 or 2 symbols (i.e. $S = \{0, 1, 2\}$). A component residing in the observable part of the system may take state value either 0 or 1. On the other hand, the state value of a component in reservoir part can only be 2. This value represents unknown states of separate/disconnected components.

From Figure 1, we can conclude that set of system components $H = \{p, q, r, s, t, u, v, w, x, y, z\}$; finite set of components at time t, $V_t = \{p, r, t, v, w, y, z\}$; component states $C_t\,(r) = C_t\,(t) = C_t\,(v) = C_t\,(z) = 0$ and $C_t\,(p) = C_t\,(w) = C_t\,(y) = 1$; destination component sets for each component $L_t\,(p) = \{z\}$, $L_t\,(r) = \{v\}$, $L_t\,(t) = \{p\}$, $L_t\,(v) = \{p, w\}$, $L_t\,(w) = \emptyset$, $L_t\,(y) = \{r, t, w\}$, $L_t\,(z) = \{t, y\}$. The state of any component at reservoir part is unknown and symbolized by 2.

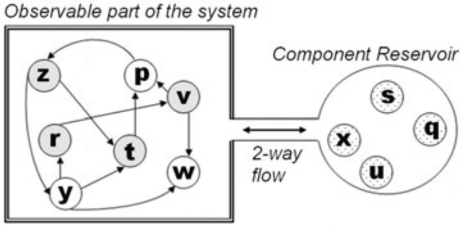


Fig. 1. An example system snapshot

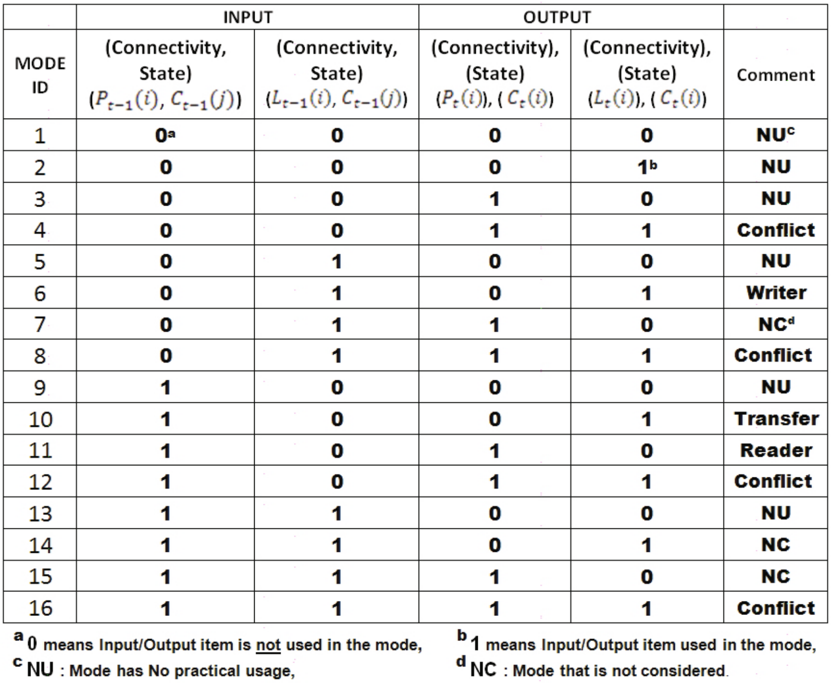
3 The Algorithm

In this section, we define an identification algorithm for the restricted type of systems defined by the assumptions given in section 2. The inputs of the algorithm are a sequence of GNA configurations reflecting the restricted system's global behavior; a finite ordered set H of components from which V_t sets are constructed and state/connectivity interaction mode considered in the identification process. A fundamental question that we have to answer is: "How should component states and connectivity affect each other so that given autonomously varying topology and state change sequence can be generated?". For that purpose, the algorithm considers three alternative state/connectivity interaction modes which characterize single component behavior that drive general GNA system evolution: *reader* mode, *writer* mode and *transfer* mode. For a given system, all components are supposed to behave in the same mode. In reader mode, component i is first supposed to be able to identify its incoming edges and their corresponding source nodes $P_{t-1}(i)$ (this set may also include the component itself) together with their states $C_{t-1}(j)$ where $j \in P_{t-1}(i)$ then decide on new incoming components $P_t(i)$ and the component i's new state $C_t(i)$.

Similarly in writer mode, component i first determines destination components $L_{t-1}(i)$ through outgoing edges and their corresponding states $C_{t-1}(j)$ where $j \in L_{t-1}(i)$ then the state and connectivity information is used to generate new outgoing components set $L_t(i)$ and new state of component i, $C_t(i)$.

In transfer mode, component i is supposed to generate new outgoing edges (i.e. destination components $L_t(i)$) and its state $C_t(i)$ based on its incoming components $P_{t-1}(i)$ and their states $C_{t-1}(j)$ where $j \in P_{t-1}(i)$ of the corresponding source components in previous time step. In this mode, the effect of individual component i over the whole system behavior is realized through transformation of incoming connectivity and corresponding state information to outgoing connectivity and the component's state.

Table 1. Mode Table - List of [(Connectivity,State)] → [(Connectivity),(State)] based component behavior modes for minimal memory restricted systems under study



	INPUT		OUTPUT		
MODE ID	(Connectivity, State) $(P_{t-1}(i), C_{t-1}(j))$	(Connectivity, State) $(L_{t-1}(i), C_{t-1}(j))$	(Connectivity), (State) $(P_t(i)), (C_t(i))$	(Connectivity), (State) $(L_t(i)), (C_t(i))$	Comment
1	**0**[a]	**0**	**0**	**0**	**NU**[c]
2	**0**	**0**	**0**	**1**[b]	**NU**
3	**0**	**0**	**1**	**0**	**NU**
4	**0**	**0**	**1**	**1**	**Conflict**
5	**0**	**1**	**0**	**0**	**NU**
6	**0**	**1**	**0**	**1**	**Writer**
7	**0**	**1**	**1**	**0**	**NC**[d]
8	**0**	**1**	**1**	**1**	**Conflict**
9	**1**	**0**	**0**	**0**	**NU**
10	**1**	**0**	**0**	**1**	**Transfer**
11	**1**	**0**	**1**	**0**	**Reader**
12	**1**	**0**	**1**	**1**	**Conflict**
13	**1**	**1**	**0**	**0**	**NU**
14	**1**	**1**	**0**	**1**	**NC**
15	**1**	**1**	**1**	**0**	**NC**
16	**1**	**1**	**1**	**1**	**Conflict**

[a] **0 means Input/Output item is not used in the mode,** [b] **1 means Input/Output item used in the mode,**
[c] **NU : Mode has No practical usage,** [d] **NC : Mode that is not considered.**

In computations, P_t is assumed to be a mapping $V_t \rightarrow V_t^*$ defining global topology of the system at time t by incoming edges of node i from its source nodes. We can relate P_t and L_t in such a way that for all $j \in P_t(i)$ we require $i \in L_t(j)$ and for all $j \in L_t(i)$ we require $i \in P_t(j)$. Clearly, one can obtain all P_t values from all L_t values. None of the proposed modes cause conflicts in terms of their decided state and connectivity values. One can consider alternative component behavior modes that can explain the restricted system's global behavior in terms of (connectivity, state) input tuples generating (connectivity) and (state) outputs (see Table 1).

In the Mode table, a component behaving in modes 1 to 4 does not get any input from the others. In other words, system components are disconnected. They have no meaning in our GNA context. Similarly, modes 1, 5, 9 and 13 do not produce any output and they do not define a dynamical system so have no practical importance in our study, Component behavior modes 4, 8, 12 and 16 update both incoming and outgoing component's connectivity, simultaneously that may result in conflicts and assumed that components live in peace. Therefore, we did not consider these modes during the identification process. However, for these modes one can establish some negotiation protocols and embedded component strategies as new constraints over the system. And, the identification algorithm can be extended to cover them also. Finally, modes 7, 14 and 15 are not considered but they can easily be added to the implementation. In the implementation of the algorithm, information about existence of incoming or outgoing connectivity between components (defined by sets $P_t(i)$ and $L_t(i)$ is encoded as 0 and 1 for simplicity. Depending on given configuration sequence, deterministic behavior of an individual component may not be explainable only by its own history. We call it *inconsistency*. They are resolved by neighborhood expansion of component in concern. So, the inconsistencies are resolved by the establishment of what we call *overlay* level component dependency settings that leads to the identification of an overlay level network topology. Initially, each component is assumed to depend only on itself. Therefore, the only overlay level neighbors of components are themselves, initially. Any neighborhood expansion may continue at most until all system components are added into the current overlay level neighborhood of the component under construction. Note that any inconsistency is guaranteed to be resolved at worst when the identified overlay network become fully-connected. This is because the given global behavior of the system has been assumed to be deterministic. The following example covers an inconsistency situation resolved through component overlay neighborhood expansion realized over finite ordered set H of system components which is given as input.In below, we give the proposed identification algorithm *IDENTIFY_GNA()*.

Inputs:

- a deterministic sequence of $k+1$ GNA configurations $\langle V_i, C_i, L_i \rangle$ labeled from S_0 to S_k where $0 \leq i < k$ and $S_0 = S_k$
- a finite ordered set H of system components such that $V_i \subseteq H$ for all $0 \leq i < k$,
- state/connectivity interaction mode read/write/transfer.

Outputs:

- an identified set of encoded and coupled state/connectivity transition rules T_i for each system component $i \in H$
- overlay level neighborhood graph F describing the flow of state/connectivity information between system components.

Algorithm IDENTIFY_GNA():

```
For each component $i$ {
 Set rule set $T_i$ to $\emptyset$;
 Set component $i$s overlay level neighbor set $F_i$ to {i};
}
For each component $i$ {
 If (mode = reader or mode = transfer)
    Construct the sets $P_t(i)$ by using $L_t(j)$
    of other components in given sequence;
}
For each component $i$ {
 $t$=0;
 While $t$ < $k$ do  {
    Switch(mode)   {
      case reader   : Construct next rule $u$
                      by using $P_t(i)$, $P_{t+1}(i)$, $C_t(j)$ and current $F_i$;
      case writer   : Construct next rule $u$
                      by using $L_t(i)$, $L_{t+1}(i)$, $C_t(j)$ and current $F_i$;
      case transfer : Construct next rule $u$
                      by using $P_t(i)$, $L_{t+1}(i)$, $C_t(j)$ and current $F_i$;
    }
    If ($u$ is consistent with all the rules in $T_i$ {
       If ($u \notin T_i$) Set $T_i$ = $T_i$ $\cup$ {$u$};
       t=t+1;
    }
    Else {
       Set rule set $T_i$ to $\emptyset$;
       Find next overlay level nearest neighbor component $n$ for $i$
       by using ordered set $H$;
       Set $F_i$ = $F_i$ $\cup$ {$n$};
       $t$=0;
       Start over while loop;
    }
 }
Return all sets $T_i$ and graph $F$;
```

Note that the algorithm does not directly produce the $\langle E, R, I\rangle$ triplet describing temporal dynamics of GNA but instead transition rules that describe component level local behavior together with overlay level neighborhood topology are the outputs. In fact, the outputs describe nothing but an Automata Network. Left-hand-sides of the identified AN transition rules describe the parts to be extracted from GNA due to a single component subGNA and right-hand-sides define the parts that replace the extracted left-hand-sides on the same single component subGNA scale. Collective extraction and replacement effort of the components results in sequence generation. The AN transition rules are sufficient to generate given global system behavior because it can be explained by collective behavior of individual system components co-evolving through state and connectivity based overlay level complex neighborhood interactions. Depending on the nature of the sequence, the identified AN may fit to instances of cellular automata, random Boolean networks, graph grammar, pure network growth models or any other hybrid GNA model. Finally, the identified rulespace of AN model might be partially defined since the information provided by GNA sequence is limited to some finite number of configurations.

4 Conclusions

Identification of the relation between local and global, subsystem and system or simply the part and the whole is a known fundamental question. An algorithm to identify the behavior of complex discrete dynamical systems from their whole behavior description to the parts behavior identification is developed. During identification of state/connectivity based transition rules of the systems in concern, apriori system knowledge is treated as assumptions/restrictions. An abstraction what we call overlay level nearest neighbor component interaction is introduced. Different component behavior modes (namely reader, writer and transfer) are defined to explain the whole systems behavior by simple local component set behavior. Clearly, the more information (i.e. state/topology change history) provided about the system the more rules can be identified.

Finally, we showed the existence of a GNA identification procedure for at least systems with some defined assumptions and restrictions. In fact, automated identifiability of systems (up to some degree) of different scales ranging from nonliving systems (e.g. communication networks), cellular/species level biological/ecological systems to human systems through devising computer procedures/algorithms is still questionable. This is because of known limitations of computing based on formal description of algorithms built over the Church-Turing thesis. The existence of alternative/unconventional hypercomputing approaches and their products may (hopefully/hopeholy but inshAllah) result in contribution to peace and charity. However, it seems that the success is technically upper bounded by human-being himself/herself who is complicated enough to be identified/understood. The proof is the existence of the authors of this paper. As a consequence, any identification/decision made about it only based on the letter sequences of *this* paper may not be sufficient for healthy model construction, not only for an intelligent machinery, but also for an expert of the domain. The proof, in this case, is the paper itself.

References

1. Garzon, M.: Models of Massive Parallelism Analysis of Cellular Automata and Neural Networks. Texts in Theoretical Computer Science. Springer, Heidelberg (1995)
2. Wolfram, S.: Universality and Complexity in Cellular Automata. Physica D 10, 1–35 (1984)
3. Langton, C.: Studying Artificial Life with Cellular Automata. Physica D 22, 120–149 (1986)

4. Sayama, H.: Generative Network Automata: A Generalized Framework for Modeling Complex Dynamical Systems with Autonomously Varying Topologies. In: Proc. of the 2007 IEEE Symposium on Artificial Life (CI-Alife 2007), pp. 214–221 (2007)
5. Goles, E., Martinez, S.: Neural and Automata Networks: Dynamical Behavior and Applications. Kluwer Academic Publishers, The Netherlands (1990)
6. Von Neumann, J.: Theory of Self-Reproducing Automata, edited by A.W. Burks. University of Illinois, Urbana (1966)
7. Kılıç, H.: A Methodology for Reverse Engineering Automata Networks, Ph.D. Dissertation, Middle East Technical University, Department of Computer Engineering, 135 pages, Ankara, Turkey (1998), http://www.atilim.edu.tr/~hurevren/PHDTHESIS_HUR.rar

Solving a Heterogeneous Fleet Vehicle Routing Problem with Time Windows through the Asynchronous Situated Coevolution Algorithm

Abraham Prieto, Francisco Bellas, Pilar Caamaño, and Richard J. Duro

Integrated Group for Engineering Research,
Universidade da Coruña, Spain
{abprieto,fran,pcsobrino,richard}@udc.es

Abstract. In this work we present the practical application of the Asynchronous Situated Coevolution (ASiCo) algorithm to a special type of vehicle routing problem, the heterogeneous fleet vehicle routing problem with time windows (HVRPTW). It consists in simultaneously determining the composition and the routing of a fleet of heterogeneous vehicles in order to serve a set of time-constrained delivery demands. The ASiCo algorithm performs a situated coevolution process inspired on those typical of the Artificial Life field that has been improved with a strategy to guide the evolution towards a design objective. This strategy is based on the principled evaluation function selection for evolving coordinated multirobot systems developed by Agogino and Tumer. ASiCo has been designed to solve dynamic, distributed and combinatorial optimization problems in a completely decentralized way, resulting in an alternative approach to be applied to several engineering optimization domains where current algorithms perform unsatisfactorily.

Keywords: Open-ended Evolution, Optimization, Vehicle Routing Problem.

1 Motivation

Typical engineering optimization problems belonging to the Operational Research field such as dynamic programming, assignment, scheduling, supply chain flow, network optimization or decision analysis, have been widely used, due to their complexity, as testbeds for new optimization techniques [1]. In fact, solving the real applications behind these problems is still a challenge, mostly due to the fact that the simplifications assumed in the computational models are usually substantial. Specifically, providing a practical solution for real time operation implies dealing with three main features: the huge number of interactions between the elements that make up the system, the locality of the available information and the dynamic properties of the environments. In this work, we study the application of a new evolutionary approach based on the algorithms used in Artificial Life simulations called ASiCo (Asynchronous Situated Coevolution) that includes these three features as a design requirement.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 200–207, 2011.

The idea of applying Artificial Life simulations in fields different from biology is not new [2]. In the particular case of engineering problems, these techniques have not been studied in depth but, nevertheless, relevant work centered on function optimization that has been applied to real problems may be found. Examples of this have been reported by Satoh [3] or Yang [4] among others. These authors propose an emergent colonization algorithm for the optimization of non-convex functions. The algorithm was later applied in [5] to design tasks. Another example that hybridizes different approaches is the one found in [6] where the authors propose an algorithm that combines an Artificial Life simulation with a Tabu search algorithm for the optimal design of an engine mount.

Within a different field, that of autonomous robotics, we must mention the work by Watson et al. on Embodied Evolution [7]. Basically, the authors seek a robot population that evolves in a completely autonomous manner, without external intervention to obtain a single controller that makes the robots fulfill their objectives. Recently, we have expanded this idea in order to consider the possibility of obtaining robot populations that cooperate towards an objective. In fact, this was the origin of the ASiCo algorithm as proposed by our group [8][9] and used in collaboration with Schut et al. [10] for the operation of a set of robots performing a surveillance task. The results show the appropriateness of the approach for real time operation over a fixed number of robots or agents.

Here we go one step further and evaluate the possibility of obtaining differentiated controllers (different species) for a heterogeneous population of agents. In fact, the objective is to test this strategy in a problem that would require dimensioning the agent set, selecting the appropriate features for each agent (size, capabilities), and constructing the individual agent controllers simultaneously. For formal reasons we have decided to do this on a well described and established problem within operational research, that is, the heterogeneous fleet vehicle routing problem with time windows (HVRPTW).

2 Asynchronous Situated Coevolution

The ASiCo algorithm is inspired on Artificial Life simulations in terms of the use of decentralized and asynchronous open-ended evolution. Unlike other bio-inspired approaches such as genetic algorithms in which selection and evaluation of the individuals is carried out in a centralized manner at regular processing intervals based on an objective function, this type of evolution is situated. This means that all of the interactions among individuals of the population are local and depend on spatial and temporal coincidence of the individuals, implying an intrinsic decentralization. Consequently, reproduction, creation of new individuals or their elimination is driven by events that may occur in the environment in a decentralized way.

This type of evolution has usually been employed for analysis purposes, this is, to study how a system evolves in an open-ended manner, and not really with an engineering objective in mind, and thus, there is no clear procedure to relate the global objective to be achieved with the local objectives of the agents that participate in the process. To this end, we take inspiration from the studies of utility functions and their distribution among individuals in order to structure the energy dynamics of the environment to guide evolution to the objectives sought. Specifically, we have used the principled evaluation function selection procedure for evolving coordinated multirobot systems developed by Agogino and Tumer [11], which establishes a formal procedure to obtain the individual utility function from the global function. With this procedure, ASiCo open-ended evolution becomes an evolutionary optimization algorithm that provides a distributed solution by means of the *whole population* and not only by the best individual as in typical evolutionary algorithms.

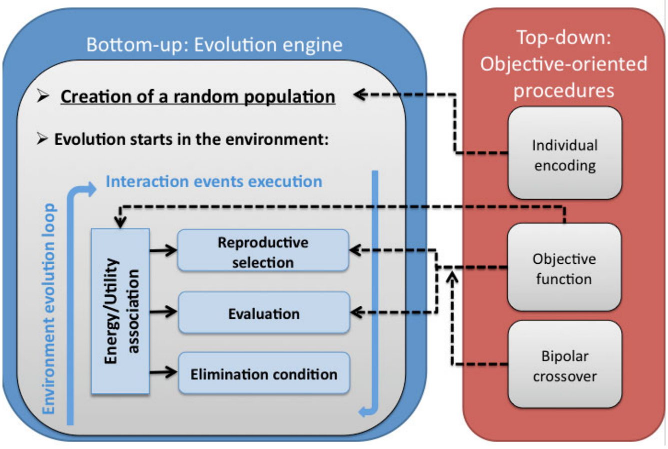

Bottom-up: Evolution engine
Creation of a random population
Evolution starts in the environment:
Interaction events execution
Environment evolution loop
Energy/Utility association
Reproductive selection
Evaluation
Elimination condition
Top-down: Objective-oriented procedures
Individual encoding
Objective function
Bipolar crossover

Fig. 1. Schematic representation of ASiCo structure

Fig. 1 displays a schematic representation of the algorithm's structure divided into two different parts, and the relations between these procedures and the processes carried out during evolution. On one hand (right block) we have the procedures that guide the evolution towards an objective. *Individual encoding* defines the solutions that can be generated, the *objective function* is established using the utility functions commented above, and, finally *bipolar crossover*, explained in detail below, allows for the evolution of a heterogeneous population. On the other hand (left block) we have the *evolution engine*, which is based on the interactions among elements in the environment. After the *creation of a random population*, the execution of the interaction events occurs in a continuous loop modifying the state of the elements. In some conditions and based on energetic criteria for spatiotemporally coinciding individuals, the procedures that represent the evolution of the population (*selection*, *evaluation* and *elimination*) occur. The *energy/utility association* represents the energetic rules in the environment and affects these three procedures. Finally, the objective function defines the energetic criteria, the selection and the elimination.

Thus, ASiCo is an interaction driven algorithm. Interactions are a set of rules that make the state of the elements and individuals in the environment change in time due to particular events. This process is independent from the evolution of the population. Two elements are very relevant within ASiCo. On one hand we have the flow of energy, which represents the different rules that regulate energy variations and transmissions between the individuals and the environment and vice versa. On the other, reproductive selection is the set of rules that regulates the reproduction process. This selection process must be defined for each problem and is based on spatial interactions together with some energetic criteria. Specifically, it is usually performed by means of a tournament operator, which, in a typical evolutionary algorithm, randomly selects a number of individuals from the population for the reproduction. This centralized behavior is not possible here, so the tournament has been modified to be *asynchronous and decentralized* and, consequently, based on local interactions between the individuals.

The reproduction process uses a *bipolar* crossover operator that we have developed to preserve *heterogeneous populations* in the evolution process. To achieve it, when two parent individuals are selected to create an offspring, one of them is randomly labeled as the *base individual*, and the resulting offspring will be a variation of this base individual. The crossover is performed gene by gene applying a Gaussian function that is centered on the genes of the base individual and with a deviation function that depends on the difference between the parents' genes. The idea is that, in the case of having two very different parent individuals (that could be considered as belonging to different species), this operator tends to create an offspring that is more similar to one of them, with a very small probability of being a mixture as this would eventually lead to a homogeneous population.

3 The Shipping Freight Problem

In order to evaluate ASiCo´s capability of generating heterogeneous variable sized populations in real time, we have solved a heterogeneous fleet vehicle routing problem with time windows (HVRPTW), as a testbed for real applications. It consists in simultaneously determining the composition and the routing controllers for a fleet of heterogeneous vehicles in order to serve a given set of customers with probabilistic delivery demands that have time constraints.

Several algorithms may be found that were designed specifically to solve HVRPTW. The most successful results have been achieved using classical heuristic methods adapted to the problem such as [12], or metaheuristics [13][14]. In this work, we apply a more general algorithm, which has not been designed for this particular problem. In addition, as we will explain later, ASiCo allows to simultaneously obtain the fleet composition and the optimal routing with a continuous range of possible vehicles, that is, we do not have to use fixed sets of vehicles to make up the fleet. Finally, ASiCo may be used in real time and adapts to changing circumstances in the environment or agents.

3.1 Experimental Setup

The particular HVRPTW we have studied is a *shipping freight* problem, that is, the fleet will be made up of cargo *ships* that must supply the *demands* that appear in some points (representing naval ports). For the sake of realism, two more elements have been added: *supply* points, where the cargo must be delivered, and *maintenance* points, that the ships must visit periodically. These 4 elements make up the simulation environment and they are represented in Fig. 2 where a screenshot of this application example in the ASiCo simulator [8] is shown.

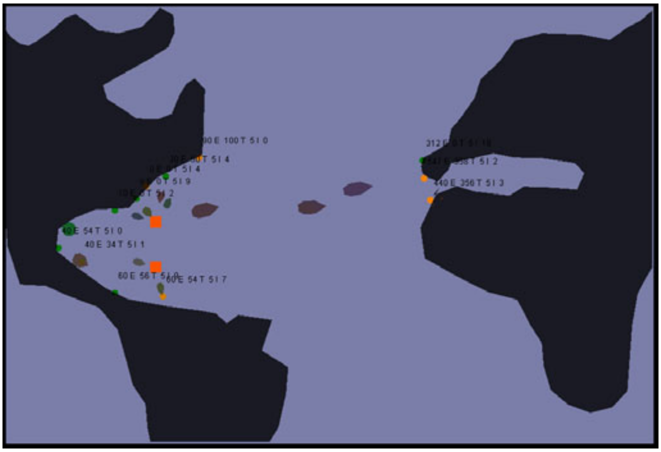


Fig. 2. Screenshot of the shipping simulation environment. Demand points are represented by small circles, supply points by squares and maintenance points by large circles. The remaining elements are ships of different size.

The objective of the ASiCO algorithm is *to obtain a fleet that minimizes the unsatisfied demand level*. Specifically, it must provide the composition of the fleet (number and size of ships) and the control system for each ship to reach this objective. To achieve it, the population of the algorithm is made up of individuals representing the ships that have the following 5 sensors:

1. *Reward*: provides the estimated level of resources the ship will have after performing a delivery.
2. *Minimal resources*: provides the minimum level of resources that is reached during the itinerary of a given route.
3. *Distance*: to the destination along a given route.
4. *Autonomy*: provides the autonomy the ship would have in the nearest maintenance point after the delivery following a given route.
5. *Load*: provides the load the ship would have after the delivery.

The action the ships can execute is simply the selection of the route they will follow to a demand point, either directly, through a supply point, through a maintenance point or through both, a supply and a maintenance point.

The control system of the ships acts as an evaluator that must select the route the ship must follow using information from the sensor values. This selection is performed through a function that provides an evaluation parameter (E), which depends on the load (C_L), autonomy (C_A) and resources (C_R) coefficients that represent the relevance of these input values in the selection of the route. In addition, E depends on the *reward rate* (R_{rw}) of a given route, obtained from the estimated time to reach a demand point, the resources consumed and the estimated reward. Thus, the valuation of each possible route can be obtained using:

$$E = R_{rw} C_L C_A C_r$$

Consequently, the genotype of each individual is made up of the following parameters: the capacity (Q) that indicates the maximum load a ship can transport, the autonomy (A) that indicates the time between maintenance operations, the velocity (V) and the 3 coefficients C_L, C_A and C_R.

The energy input in this environment occurs at the supply points. Every time a ship performs a delivery, it receives a reward value associated to the merchandise deposited there. Energy output takes place through resource consumption that occurs during each ship's life represented by two economic factors, the fuel consumption and the maintenance costs. The depreciation of the ship´s value has also been included in the consumption rate. It depends on the velocity, the autonomy and the fuel consumption according to [15]. Consequently we take as the global utility the sum of the private utilities of all the ships in terms of their energy, or in economic terms, their available budget.

Finally, the selection process in this problem is highly decentralized and asynchronous because the ships do not coincide spatially. Consequently, we have developed the following selection procedure: when a ship reaches a supply point with an energy level over a given threshold, it leaves a copy of its genotype. When the number of deposited genotypes is higher than a previously established tournament size, the bipolar crossover occurs using the two best individuals according to their individual utility.

3.2 Results

Using this setup we have created a simulation environment where a set of 20 demand points, located relatively near one supply and one maintenance points, were considered. Starting from an initial population of 50 ships, Fig. 3 (left) shows the changes in the fleet under these conditions. Here we have represented fleet size, fleet resources, fleet consumption and the whole demand level as time passes. It can be seen that, once fleet size becomes stable at around 40 homogeneous ships (4000 simulation steps), the demand level starts to decrease reaching a very low level. Fig. 3 (right) shows the variation in time of the average resource level and the average consumption level for the fleet. It tends to use homogeneous ships with a smaller consumption level and with a more adjusted profit margin, as expected.

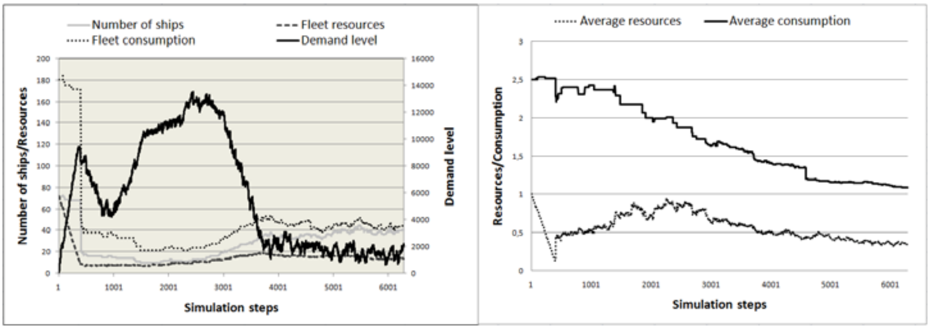


Fig. 3. Evolution results for the shipping freight problem

Due to the distribution of demand and supply points in this first experiment, we did not obtain a heterogeneous fleet as the optimal solution. To achieve it, we have created a second experiment with 17 demand points placed near one supply and one maintenance point (distance of less than 40 spatial units) and 6 demand points that are placed far from the supply and maintenance points (more than 500 spatial units). Fig. 4 (top) shows the results obtained after 9950 simulation steps, when the population is stable. The three graphs represent the relations between the three parameters that define the fleet composition: capacity, velocity and number of ships. As shown in the graphs, ASiCo produces a fleet made up of 22 ships, 12 with low capacity and velocity and 10 of higher capacities and velocities, constituting a self-organized heterogeneous solution. This distribution is consequent with that of the demand and supply points, because we obtain a fleet of 12 slow and small ships to cover the short haul routes and 10 faster and larger ships to deliver in long haul routes.

Finally, in Fig. 4 (bottom) we show the parameters that define the fleet at step 21400 if we simulate a variation in the fuel costs in simulation step 10000. As displayed in the figure, the fleet changes, as expected: it is made up of 13 ships, all of them are slow (right graph) and 11 of them have a low capacity (center graph). Obviously, they do cover the demand and the fleet is profitable despite the fuel price increase. With this result we show the ability of ASiCO to adapt the solutions to dynamic environments.

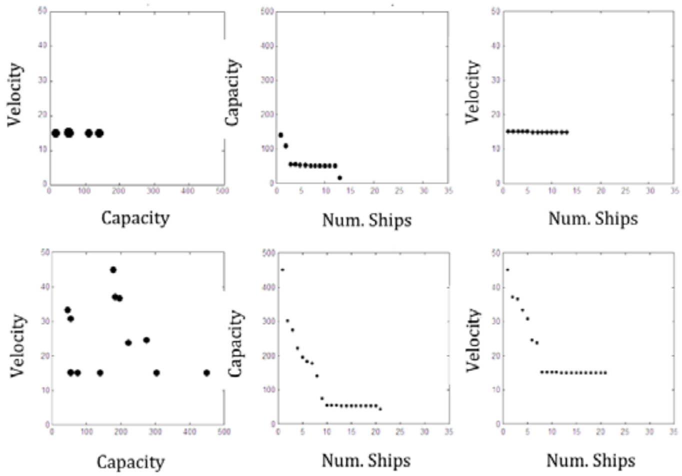


Fig. 4. Distribution of ships in the fleet in simulation step 9950 (top) and 21400 (bottom) after a variation in the fuel cost in step 10000

4 Conclusions

A heterogeneous fleet vehicle routing problems with time windows (HFVRPTW) is solved using the Asynchronous Situated Coevolution (ASiCo) algorithm. This problem is a very relevant testbed in the operational research field due to its broad range of practical applications. In this work, we show how an open-ended evolutionary simulation improved with a procedure to guide evolution towards a design objective provides successful results to this problem. The solution obtained is a self-organized fleet of heterogeneous ships that distribute their features adapted to the spatial distribution of the demand and supply points according to market requirements and restricted to time windows.

Acknowledgments. This work was partially funded by the MEC of Spain through projects DEP2006-56158-C03-02 and DPI2006-15346-C03-01.

References

1. Doerner, K.F., Gendreau, M., Greistorfer, P., Gutjahr, W., Hartl, R.F., Reimann, M.: Metaheuristics, Progress in Complex Systems Optimization. Operations Research/Computer Science Interfaces Series, vol. 39. Springer, Heidelberg (2007)
2. Langton, C.: Artificial Life: an Overview. MIT Press, Cambridge (1997)
3. Satoh, T., Uchibori, A., Tanaka, K.: Artificial life system for optimization of nonconvex functions, vol. 4, pp. 2390–2393 (1999)
4. Yang, B.S., Lee, Y.H.: Artificial life algorithm for function optimization. In: Proceedings of 2000 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (2000)
5. Yang, B., Lee, Y., Choi, B., Kim, H.: Optimum design of short journal bearings by artificial life algorithm. Tribology International 34(7), 427–435 (2001)
6. Ahn, Y.K., Song, J.D., Yang, B.: Optimal design of engine mount using an artificial life algorithm. Journal of Sound and Vibration 261(2), 309–328 (2003)
7. Watson, R.A., Ficici, S.G., Pollack, J.B.: Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots. Robotics and Autonomous Systems 39(1), 1–18 (2002)
8. Prieto, A., Duro, R.J., Bellas, F.: Obtaining Optimization Algorithms through an Evolutionary Complex System. In: Abstracts booklet ECCS 2007, pp. 132–132. European Complex Systems Society (2007)
9. Prieto, A.: Estudio de la coevolución asíncrona situada para la resolución de problemas dinámicos descentralizados en ingeniería, Ph. D. Thesis, Universidade da Coruña (2009)
10. Schut, M.C., Haasdijk, E., Prieto, A.: Is Situated Evolution an Alternative for Classical Evolution? To appear in Proceedings CEC 2009 (2009)
11. Agogino, A., Tumer, K.: Efficient evaluation functions for evolving coordination. Evolutionary Computation 16(2), 257–288 (2008)
12. Imrana, A., Salhi, S., Wassana, N.: A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. European Journal of Operational Research 197(2,1), 509–518 (2009)
13. Paraskevopoulos, D., Repoussis, P., Tarantilis, C., Ioannou, G., Prastacos, G.: A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. Journal of Heuristics 14(5), 425–455 (2008)
14. Lima, C., Goldbarg, M., Goldbarg, E.: A Memetic Algorithm for the Heterogeneous Fleet Vehicle Routing Problem. Electronic Notes in Discrete Math. 18, 171–176 (2004)
15. Stopford, M.: Is the drive for ever bigger containerships irresistible? In: Lloyds List Shipping Forecasting Conference (2002)

Modeling Living Systems

Peter Andras

School of Computing Science, Newcastle University,
Claremont Tower, Claremont Road, Newcastle upon Tyne,
NE1 7RU, United Kingdom
peter.andras@ncl.ac.uk

Abstract. A fundamental issue of evolution of life is the emergence and maintenance of self-referential autocatalytic systems (e.g. living cells). In this paper the problem is analyzed from a computational perspective. It is proposed that such systems have to be infinite autocatalytic systems, which can be considered equivalent to Turing machines. The implication of this is that searching for finite autocatalytic systems is likely to not be successful, and any such finite system would be maintainable only in a highly stable environment. The infiniteness of autocatalytic systems also implies that top-down search for the simplest living system is likely to stop at relatively complex cells that are still able to provide a realization of infinite autocatalytic systems.

Keywords: autocatalysis, computation, formal model, self-reference, infinite model.

1 Introduction

The questions of what makes living systems alive and how to create live systems are among the fundamental questions of biology [1-10]. A common feature of several theories of living systems is the requirement of self-referencing in the system [7-9], i.e. what happens in the system depends on what happened within the system before and the system somehow re-creates itself through such self-referential processes. What makes the self-referential recreation of the system problematic is that the references need to go back infinitely and the recreated system needs to be equivalent of the current one, without shrinking or explosive growth [7]. In terms of foundations of life these theories led to the postulation of autocatalytic chemical interaction systems that are assumed to provide the ground level building blocks for living systems [9].

Experimental work aimed to support theories about elementary forms of life showed that a wide range of organic molecules (e.g. amino acids, nucleobases) can be created in the conditions that may have existed on the early Earth [4,5]. Other related research aims to identify the last universal common ancestor (LUCA) organism by analyzing genetic data and experimenting with primitive organisms [10]. Another related line of research aims to produce the most reduced life by simplifying experimentally the genome of organisms that have already a small genome [11]. A further approach is to build simple autocatalytic or self-reproducing biochemical systems that are validated experimentally [12].

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 208–215, 2011.

Here I present a computational interpretation of self-referentially reproducing living systems (in particular of simple life forms). This interpretation implies the equivalence of such systems with universal computers (Turing machines). A further implication of this interpretation and analysis is that autocatalytic systems of molecular interactions have to be infinite systems.

The rest of the paper is organized as follows. First I discuss briefly autocatalytic systems. This is followed by a computational interpretation of simple living systems. Then I discuss about the infiniteness of autocatalytic systems. Finally the paper ends with the conclusion section.

2 Autocatalytic Systems

Various self-regenerating self-referential systems have been proposed as models and descriptions of living systems in the second half of the last century [7-9]. While such systems capture conceptually fundamental features of living systems, most of these proposals remain obscure and difficult to understand due to the lack of their appropriate formalization and difficulty of their predictive application. For example, the theory of autopoietic systems of Varela and Maturana [7] provides an attractive interpretation of how living systems work, but lacks the formal analytic representation that would make it practically applicable in the context of experiments.

An alternative theory of living systems was developed by Rosen [8] around the same time, following the work of Rashevsky [13]. This theory of M-R systems postulates the composition of living systems by a metabolic (M) and a repair (or reproductive) (R) component and provides a formal framework in terms of category theory. While this approach is very exciting theoretically, it did not find yet its way to the interests of experimentalists.

Another more experimentally rooted approach was proposed by Ganti [9]. This approach focuses on chemical automata (chemoton) which perform self-reproducing interactions between molecules in a similar manner as reproducing patterns are generated in cellular automata. In this context cellular automata [14] can be also seen as models of self-reproducing autocatalytic systems. The chemoton theory builds on known chemical catalytic reactions and expands these in principle to describe molecular interaction systems that can maintain and reproduce themselves. However, so far there is no clear experimental proof of such chemoton systems except living systems, in which the interactions are known only partially.

Researchers starting from the experimental end first showed that key organic molecules can be generated in abiotic conditions [4,5]. These molecules self-organize in some way to lead to living systems. The way how this happens is not yet known. The discovery of catalytic activity of RNA molecules [4,12] led to the formulation of the RNA world hypothesis, according to which interacting RNA molecules may constitute autocatalytic systems that are living systems. Recently it has been shown that small sets of RNA molecules can act alternately as catalysts, reaction components and products such that their system is reproduced if the right prime materials (molecules) and physical and chemical conditions are provided [12]. However, yet there is no sufficient explanation how proteins, RNA and DNA molecules are created and kept together in self-reproducing chemical interaction systems that are alive.

3 Computational Interpretation

In some sense, a computational interpretation of living systems is present in the M-R systems theory of Rosen [8] and in the chemoton theory of Ganti [9]. Recent works on DNA computing also point in this direction [15]. Here I present an alternative way of looking at autocatalytic self-referential systems through a computational interpretation.

First, let us consider a living system of interacting molecules that maintains and reproduces itself in the context of its environment formed of other molecules and molecular interactions. The living system is an open system, which receives molecules from the environment and releases molecules into the environment. The system can maintain itself by reproducing its molecules and molecular interactions in the right spatio-temporal patterns that correspond for example to the functional maintenance of the cell membrane and of the various cellular organelles. This means that the system is ready to produce the right molecular interactions to incorporate new required molecules (e.g. pinocytosis) and to release molecules that are not required any longer (e.g. lysosomal defecation). Being ready for such molecular interactions means that the system is able to 'predict' in a sense what these interactions need to be (e.g. by displaying the right receptor and anchor molecules on the cell membrane, and having ready sufficient amount of ATP to energize the molecular import). This kind of prediction of what the system 'expects' can be also considered as a computational process performed by the system. Note that the predictive ability of the system works well in a range of appropriate environments. If a bacterium is put in an environment with antibiotics present, the molecular interactions in the bacterium do not fit well any longer the environment and the bacterium dies. However, if the bacterium has resistance against the antibiotic, this can be interpreted in the sense that it can adapt its prediction machinery to generate the right molecular interactions which lead to the decomposition or expulsion from the cell of the antibiotic molecules.

In self-referential autocatalytic systems, such as living cells, the interactions between molecules can be considered as computational operations on the data represented by molecules. Being autocatalytic it means that the current molecular interactions in the system make possible the realization of further molecular interactions within the system. This means that the computational process calculates a representation of the system itself as well, beside of calculating an approximation (or prediction) of the environment. In other words, the approximation of the environment leads to the regeneration of the system itself. Being self-referential it means that the computations depend (reference) on the available data (molecules), on past computational operations (patterns of molecular interactions), and on the earlier data as well. To achieve this, the system needs some way of providing the information about its past ready for referencing.

The requirement of providing reference to past interactions and molecules can be satisfied if all patterns of molecular interactions (reference-able computations) can be represented by molecules (e.g. molecules that formed through these interactions), and if all patterns of molecules (reference-able data) can be represented by ongoing molecular interactions (e.g. interactions which can happen only if the referenced pattern of molecules was present earlier). This circular referencing may appear irresolvable; however there is a mathematical formalism that can provide a solution, which is the theory of recursive domain equations [16].

Let us assume that R is a domain (e.g. a set, or possibly a mathematical object that is larger than any set, for example the collection of all sets – i.e. the category **Set**), which contains objects. There are transformations of the domain R into itself; these can be considered as functions $f{:}R \rightarrow R$. The collection of all these transformations is denoted as $[R \rightarrow R]$. The recursive domain equation is stated as

$$R \cong A+[R \rightarrow R] \qquad (1)$$

where A is a part of R (a sub-domain), which may be empty. In other words, each object of the domain R is either an unrepresented object (part of A) or it is represented by a transformation $f{:}R \rightarrow R$, and each transformation $f{:}R \rightarrow R$ is represented by an object of R. To build up the analogy with the autocatalytic self-referential molecular interaction systems, let us consider each molecule type and also types of combinations of such molecules (e.g. proteins and protein complexes composed of multiple subunits constituting independent molecules) as an object and the transformations of the cell that result from patterns of molecular interactions the analogues of transformations of the domain, which is the collection of all considered molecule and molecule combination types. It is possible that there are molecules which have no transformation representation (e.g. inorganic ions that flow into the cell from the environment), but all transformations have some representation in terms of molecules or patterns of molecules, and all molecules involved in the self-reproduction have a representation in form of some transformation representing molecular interactions. If there is a solution of the recursive domain equation (1) then that solution may be used as a model of autocatalytic self-referential molecular interaction systems that are living cells.

Equation (1) has solutions, for example the category of partial orders satisfies this equation (collection of all partial orders defined on sets, together with all functions that transform one partial order into another partial order). In general there are many categories, which satisfy this equation. These categories are the so called Cartesian-closed categories, which are defined by following features: (a) they have a Cartesian product object for each finite set of objects and (b) for any ordered pair of objects they have an object representing the set of morphisms between the first and the second object (for more information about categories and category theory the reader should consult relevant textbooks – e.g. [17]). This means that Cartesian-closed categories can be considered as models of self-referential autocatalytic systems. In fact considering the above identified requirements of self-referential autocatalysis any model of these systems has to be a such category.

Notably, Cartesian-closed categories are also models of λ-calculus, which are a representation of Turing machines [18]. Turing machines are universal computational machines that are able to represent any computational algorithm and consequently can compute anything that is computable. This means that if self-referential autocatalytic molecular interaction systems (i.e. living cells) are representable as Cartesian-closed categories then they can also be seen as representations of Turing machines that can compute anything computable in principle.

This implies that indeed the molecular interactions that happen in a living cell can be seen as a representation of computational processes. I argued earlier that these computations may be considered as computations leading to an approximation (or prediction) of the environment. In principle the Turing machine equivalence suggests that these computations may compute everything computable, and consequently living cells may approximate their environment arbitrarily correctly. However, the high precision approximation computations may need very much time and space, and time and space limits may limit the attainable precision of the environment approximation. So, in practice the living system is able to achieve only limited precision approximation of its environment, which implies that there are environments in which these system cannot survive (since they cannot approximate it or predict it sufficiently correctly) – e.g. consider the earlier example of antibiotics in the environment of non-resistant bacteria.

4 Infinite Autocatalytic Systems

The previously discussed computational interpretation of self-referential autocatalytic systems shows that such living systems of molecular interactions (cells) can be represented as Cartesian-closed categories, and cannot be represented by smaller formal representations. These categories include infinitely many objects, such that the level of this infinity is the same as the level of infinity in the case of the collection of all sets (i.e. more infinite than any infinite set). The morphisms between any ordered pair of objects always form a set, i.e. their number may be infinite, but not more infinite than the most infinite set (note that sets can be countable infinite – i.e. same as the infinity of natural numbers, and continuously infinite – i.e. the same as the infinity of real numbers). This implies that molecular interaction systems that compose living cells have to be infinite systems in principle.

Infinite autocatalytic systems have to have infinitely many molecules and molecular configurations corresponding to the infinitely many objects of the representing Cartesian-closed categories. The infinity of molecules and molecular configurations has to be comparable to the infinity of collection of all sets (i.e. more than the infinity of real numbers). This may seem to be contradictory with the evidence of observed molecules in living cells. However, considering the range of macromolecules (e.g. proteins, RNA, DNA), their complexes and polymers, and also the polymers of simpler organic molecules (these polymers may approximate infinitely long molecules in principle), it becomes more plausible the assumption that there are very infinitely many molecules, complexes and molecular configurations that may represent the infinitely many objects of Cartesian-closed categories. The morphisms between objects of a corresponding Cartesian-closed category can be represented as interactions between molecules or spatio-temporal patterns of such interactions.

Note that the DNA in itself can be seen as a computational machine equivalent of Turing machines [15]. This means that in principle (assuming that the DNA's length approximates countable infinity) it can represent any computational processes and the computational system represented by it is the equivalent of λ-calculus. Consequently the system of computations that can be represented by the DNA can represent a corresponding Cartesian-closed category. So, what the DNA can encode in principle is equivalent to a system with very infinitely many objects and sets of morphisms between these objects. This means that the molecular interaction system encoded by the DNA, i.e. involving proteins, RNA, their complexes and spatio-temporal patterns, is a representation of an infinite Cartesian-closed category. This confirms that the above conclusion that living cells form infinite autocatalytic systems is valid.

Still there is an issue about how to accommodate the observational fact of finite sets of types and instances of molecules in cells with the assumption of infinite autocatalytic systems considered to be represented by these cells. To resolve this issue, let us start by considering that cells are open systems. This means that molecules flow into and out of the cell. Many of these molecules are essential components of proteins or prime material for molecular components of other macromolecules. In this sense the molecular interaction system of the cell extends out of the physical boundaries of the cell, by involving other molecule types that are not produced in the cell. Next, let us consider the cell together with its ancestor and descendant cells. The system of the cell is open in this temporal sense as well, since molecular interactions in the cell depend on molecules and molecular interactions that composed its ancestor cells and determine the molecular composition and molecular interaction of its descendant cells. While the cell contains a finite set of molecules at any time, considering the molecular interaction system spanning through ancestor and descendant cells as well, the molecules and their interactions constitute an approximation of an infinite fragment of the autocatalytic system represented by the cell.

Different cells represent different fragments of the full system. They are adapted to their molecular environment in the sense that the computations represented by them assume the presence of this molecular environment, which may include required organic and inorganic molecules for the proper functioning of the cell, which are not produced by the cell. This means that it is sufficient for these cells to represent the appropriate fragment of the full autocatalytic system that can deal with the approximation / prediction of the characteristic environment of the cell. Since the part of the environment that is not given is at least comparably complex as cells themselves (e.g. many other cells contribute to it), the fragment of the autocatalytic system represented by the cell has to be infinite itself to make able the cell to compute good approximations / predictions of this environment.

An implication of the infiniteness of the autocatalytic system is that searching for finite autocatalytic systems is likely to be fruitless. Finite variants of autocatalytic systems can be considered at best as finite fragments of an autocatalytic system that can maintain itself in the context of a stable specific environment providing inputs to and taking away outputs from the self-reproducing molecular interaction system (see for example [12]) – significant variation of the environment cannot be dealt with by the finite autocatalytic system, since this is unable to perform the full range of required computations. The computational requirements imposed by a partially given environment mean that any molecular realization of a fragment of an autocatalytic system that can reproduce in this partially given environment should be able to represent an infinite fragment of the autocatalytic system. This means that the bottom-up search for autocatalytic systems by combining molecules and their interactions is unlikely to be successful unless the resulting molecular interaction system can be considered a representation of an infinite autocatalytic system. On the other side, it also means that top-down search for minimal self-reproducing autocatalytic systems is likely to stop at the level of relatively complex cellular systems that can still realize such infinite autocatalytic systems.

5 Conclusions

A formal computational interpretation of living systems (live cells) is presented in this paper. This interpretation leads to the conclusion that self-referential autocatalytic systems have to be infinite systems that have an equivalent representation in form of a Cartesian-closed category. It is argued that such autocatalytic systems are equivalent to Turing machines as well, and in principle can compute anything computable.

The realizations of such systems are live cells that have temporal and spatial constraints which imply that their computational ability in practice is limited and they can predict (or approximate) their environment to some extent that allows their survival and reproduction. However this ability depends on the assumptions of the system about its environment. If these assumptions are violated in a critical way that may imply the drop in the predictive ability of the system in the context of the modified environment and may lead to the termination of the system (e.g. bacteria in presence of antibiotics).

An important implication of this conclusion about autocatalytic systems is that searching for their finite variants is likely to be fruitless, and in the best case finite variants of them will depend on very stable environmental condition. It also implies that top-down search for the simplest live organism is also likely to stop at relatively complex organisms that can still represent an infinite fragment of the infinite autocatalytic system.

References

1. Miller, J.G.: Living Systems. McGraw-Hill, New York (1978)
2. Lovelock, J.: Gaia: A New Look at Life on Earth. Oxford University Press, New York (1987)
3. Kauffmann, S.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York (1993)
4. Joyce, G.F.: Booting up life. Nature 420, 278–279 (2002)
5. Miller, S.L., Orgel, L.E.: The Origins of Life on the Earth. Prentice Hall, Englewood Cliffs (1974)
6. Segre, D., Lancet, D.: Composing life. EMBO Reports 1, 217–222 (2000)
7. Maturana, H.R., Varela, F.J.: Autopoiesis and Cognition. In: The Realization of the Living. D. Reidel Publishing Company, Dordrecht (1980)
8. Rosen, R.: Life Itself. Columbia University Press (1991)
9. Ganti, T.: The Principles of Life. Oxford University Press, Oxford (2003)
10. Glansdorff, N., Xu, Y., Labedan, B.: The Last Universal Common Ancestor: emergence, constitution and genetic legacy of an elusive forerunner. Biology Direct 3, 29 (2008)
11. Lartigue, C., Glass, J.I., Alperovich, N., Pieper, R., Pramar, P.P., Hutchinson, C.A., Smith, H.O., Venter, J.C.: Genome transplantation in bacteria: changing one species to another. Science 317, 632–638 (2007)
12. Lincoln, T.A., Joyce, G.F.: Self-sustained replication of an RNA enzyme. Science 323, 1229–1232 (2009)
13. Rashevsky, N.: Outline of a unified approach to physics, biology and sociology. Bulletin of Mathematical Biophysics 31, 159–198 (1969)

14. Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)
15. Paun, G., Rozenberg, G., Salomaa, A.: DNA Computing. Springer, Heidelberg (1998)
16. Pierce, B.J.: Basic Category Theory for Computer Scientists. MIT Press, Cambridge (1991)
17. Mac Lane, S.: Categories for the Working Mathematician. Springer, New York (1998)
18. Crole, R.L.: Categories for Types. Cambridge University Press, Cambridge (1993)

Facing N-P Problems via Artificial Life: A Philosophical Appraisal

Carlos E. Maldonado and Nelson Gómez

Modeling and Simulation Laboratory,
Universidad del Rosario,
Bogotá, Colombia
{carlos.maldonado,nelson.gomez}@urosario.edu.co

Abstract. Life as an N-P problem is a philosophical, scientific and engineering concern. N-P problems can be understood and worked out via artificial life. However, these problems demand a new understanding of engineering, since engineering is basically a way of acting upon the world. Such a new engineering is known as non-conventional engineering or also as complex systems engineering. Bio-inspired systems are more flexible and allow a higher number of degrees of freedom. As a consequence, AL enlarges our understanding of living systems in general and can be taken as a step forwards in grasping the complexity of life.

Keywords: Complexity, N-P Problems, Non-conventional Engineering, Non-Conventional Computing, Living Systems, Heuristics, Experimentation.

1 Life Is an N-P Problem

Generally speaking, life is an N-P problem. However, a feasible way for dealing with N-P problems and trying to solve them is via thinking, experimenting and working on and with AL. The workings and research on and with artificial life (AL) cover three main action domains, namely philosophy, science, and engineering. The first two deal with understanding and explaining phenomena, systems and dynamics exhibiting life –natural or artificial– whereas the third is mainly centered around building, optimizing, predicting and/or controlling engineering-like systems that behave as living beings, or having living features or characteristics. Two outcomes can be derived hereafter [1]: firstly, models and simulations for studying life (*scientific orientation*); secondly, artificial systems bearing biological properties and capabilities applicable to solving problems (*engineering orientation*).

In both cases, though, the role of computing is crucial in order to literally be able to *see* non-linearity in phenomena characterized by emergence, self-organization, growing, adaptation, evolution, and increasing complexity –in one word: life! However, we claim, many if not all examples of AL and its applications in other disciplines and sciences are systems imposing tremendous computational challenges –they are N-P systems.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 216–221, 2011.

2 Engineering Life and AL

Unlike the techniques used in artificial intelligence, AL does not work uniquely on the existing computational paradigm (based on Von Neumann architecture). Rather, it *creates* in many cases new paradigms inspired by the forms of processing, architecture and dynamics of biological, social and evolutionary systems such as *evolutionary computation* (among others, J. Holland, J. Koza), *swarm intelligence* (E. Bonabeau, M. Dorigo, G. Theraulaz), *membrane computing* (Gh. Paun), *DNA computing* (L. Adleman), *artificial immune systems* or *immunological computation* (D. Dasgupta) and *cellular computing* (M. Sipper). These new *computational paradigms* belong to the so-called *non-conventional computing* and cross other paradigms from other areas and contexts such as physics and logics; such is notoriously the case of *quantum computation* (P. Benioff, S. Lloyd), *fuzzy systems* or also *hipercomputation* (foreseen somehow by A. Turing circa 1938 when working on the idea of non-computable tasks; in other words, tasks that are not carried out by a conventional Turing machine [2]).

The most generic example of the new computational paradigms arisen within the frame of AL has been pointed out by M. Sipper [3], related to cellular computing (CC). Here, most of the features and claims of AL, even philosophical ones, are gathered. The core of CC turns around principles such as simplicity, vast parallelism and locality:

- *Simplicity:* unlike current complex units, the processing unit in CC, called *cell*, can carry out very simple tasks.
- *Vast parallelism:* This principle is based on the interaction of large number of cells (around 10^x) that interact in order to carry out complex tasks at a high level that one single cell could not achieve (emergence). This is evidently a proposal quite different to those that we normally find within the frame of parallel computing and massive parallelism.
- *Locality:* The connectivity patterns –interaction- among cells are entirely local, and no single cell is able to see the whole system. In other words, it has to do with the absence of local control or with the implementation of distributed and local control techniques.

These three principles are strongly connected with each other and are necessary to create a CC; otherwise, if one of them is modified we converge to another type of computational paradigm (Fig. 1). Nonetheless, within every principle there are diverse degrees that are *eligible* according to the context of the problem one wishes to attack. The kind of cells (discrete or continuous), the scheme of connectivity (regular or irregular grid) and the dynamics in time (synchrony/asynchrony, discrete/continuous) are some of the configurable variables in CC [3].

However, the really important feature of CC and hitherto of the remaining computational paradigms of AL consists in that within the possible applications, among others, are the complete N-P problems, opening thus a wide horizon for the study of complex systems.

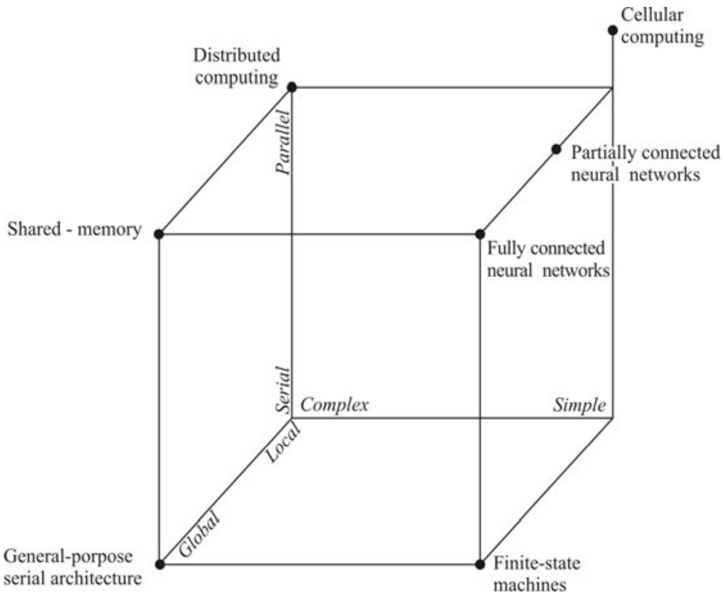


Fig. 1. Computing Cube. Adapted from Sipper [3]

3 About Non-conventional Engineering

Now, the majority of paradigms in non-conventional computation are inspired by natural systems –whether biological or not, and they are gathered under the generic name of *natural computing*. Here, as f. i. does Nunes de Castro [4] swarm intelligence or evolutionary computation are categorized as independent and parallel to artificial life; beyond such recognition, that can be debatable, what is really important consists in its underlying goals [4], namely i) to develop computation inspired on nature in order to solve complex problems; ii) to construct tools to synthesize behaviors, patterns, forms, and natural systems; and iii) to use natural materials in order to carry out computational tasks.

These underlying goals set out clearly that paradigms such as cellular computing or any other in AL or from natural computation in general are not applicable as such to any kind of problems set out by science, engineering and computation, for in many cases the conventional techniques and models are better suited and work better. In contrast, we only use those models and techniques that entail a large number of variables, non-linearity, multiple goals, N-P time, a space of solutions or, what is equivalent, more than one potential solution – whereas standard techniques provide a unique or singular solution.

M. Bedau et al. [5] and [6], have worked on the structuring of AL around fourteen open problems. Being as they are all in all relevant, these problems could make more sense if they were set as P or N-P problems. In any case, even if for the sake of agreement we can say that those are N-P problems it remains to be set whether any open problem is an N-P problem.

Here, we claim that AL is a most viable way to deal with “real” N-P problems of real life; hence its utility and significance.

In engineering in general, we have come to talk about *non-conventional engineering* or *complex systems engineering* [7] or also about a “new engineering” – three different ways referring to one and the same idea. Non-conventional engineering is rooted in the theory of non-linear dynamic systems, namely the sciences of complexity[1], and thereafter on the bio-inspired systems of AL. We strongly believe that AL can and must play a crucial role in the structuring of the frame of this new engineering, not only from the technical and engineering-like scope, but also from a philosophical, scientific, heuristic and methodological standpoint [9].

Such a new way of thinking on, and doing, engineering has made possible firstly moving away from the basic principles of classical engineering – namely centralized control, preprogramming, the search for a unique solution: optimality – towards more flexible and robust principles such as distributed and local control, emergent or bottom-up programming, a space of solutions, and secondly it has set out previously unforeseen research lines in order to implement features and characteristics of living systems, such as autonomy, adaptation, evolution, self-organization, self-synchronization, immunological response and metabolizing, among others.

4 P Problems and N-P Problems within the Frame of AL

A problem to which any problem in P can be reduced is called P-hard; if it also belongs to P, is called P-complete. Thus, to show that a problem is P-hard, it is enough to reduce a P-complete problem to it. Similarly, a system is universal if it may simulate a universal Turing machine. If so, then the problem arises about the decidability or undecidability of the system, i.e. whether we can safely say whether the program stops or not and when, and also whether the program can be compressed. A system is P-hard, when it admits a P-hard problem.

Any P problem, whether P-hard or P-complete, assesses and presupposes at the same time a polynomial time. Polynomial time is critical to living organisms as circadian cycles, or also at the scale of developmental biology. The critical case is the study of apoptosis and, hence, the biological clock [10]. ALife systems however not always “die”. They just vanish in simulated programs. What is it for an AL system to die? It is conspicuous to notice that artificial organisms may die, as they do indeed, but the program does not! Inversely, an AL-organism is born as the program creates it, but the very process of giving birth and developing on the evolution of an AL system is the very success of the program! The program can be different as it happens: artificial chemistry, swarm intelligence, and the like.

This idea could lead us back to the belief of a “programmer of the universe”, a dream that has already been dreamt. The reply to the belief can be positive if we think of conventional engineering, but it can be negative if we take into consideration non-conventional engineering, i. e., artificial chemistry, and the very possibility of programs that create other programs [11] and [12].

[1] For example, the thermodynamics of non-equilibrium, the theory (or science) of chaos, fractals, the theory of bifurcations, self-organized criticality, and the science of complex networks. See Maldonado [8].

What can be thought of as a kind of biological clock within AL-systems, is simply a matter of programming. The question arises: can we really talk about N-P problems in the frame of AL? So far the answer clearly seems to be: no. If so, then AL is at most a P-complete problem; no more, no less.

The implications of such an acknowledgement are twofold: on the one hand, AL is susceptible only of P-treatments; hence, N-P problems are a patrimony of carbon-based life, not to talk about human systems. On the other hand, we could figure out as a question of possibilities N-P problems for AL. If so, the best chance comes from engineering AL in that direction.

As a consequence, three big axes are useful as references when working with or studying AL, namely the philosophical significance, the scientific endeavor and the engineering featuring of AL. As for the N-P problems, we want to asses that engineering AL is most useful to deal, understand and try to solve them. Nonetheless the road remains open and not fully experienced or crossed. Just some steps, although steady, have been done so far.

The crux of the studies on life, whether natural or artificial, is about the meaning of life. To be sure, any scientific answer to this question crosses necessarily throughout the fields of N-P problems. Living systems live by solving P problems, but when solving these kind of problems, they happen to encounter N-P; for instance, problems of optimization. Hence, the most basic form of appearance of N-P problems is via problems of optimization, but N-P problems do not, alas, reduce to questions concerning optimization.

Therefore, it can be useful to introduce a sort of phenomenology of N-P problems. Simulation via swarm intelligence or artificial chemistry allows us to truly "see" N-P times.

An N-P time should not be taken as a physical stance, namely quantitatively, i.e. as a large or big span of time. Such would be indeed a Newtonian understanding of time – as a "mass" or "force". Grasping N-P time in such a way is misleading and it can simply be taken as a limit as in mathematics.

5 N-P Time and Complexity

An N-P time is, accordingly to complex science, rather a quite surprised time, unexpected in principle and unforeseeable. This opens up the door to the Greek notion of *kairós*, but it can be better grasped as Nietzsche put it: *unzeitmässige* (*Betrachtungen*). Here we get a hint for further exploring of N-P time as different from a classical mechanical background (as an extensive – long span). Computationally speaking this leads us necessarily to quantum computing a most intriguing and passionate field underlying, to be sure, complex theory.

Life in general, and ALife in particular, obeys from time to time these kinds of N-P- times, and they emerge unexpectedly, as it happens. Several accounts of such phenomena have been recounted by Ch. Langton [13], [14] and [15], T. Ray [16] and [17], or J. Conway [18], among others. We can experience those accounts not just as physical phenomena, but also as temporal or time happenings not assailable to, or as, a P time.

N-P time has been conceived in terms of its computability and, thereafter, its complexity (computational complexity) [19]. The question remains whether AL can bring new insights on a broader scope on computational complexity. This, we recall, is at the same time a scientific, a philosophical and an engineering challenge and endeavor that remains, so far, still open.

References

1. Heudin, J.: Artificial Life and the Sciences of Complexity: History and Future. In: Feltz, B., Crommelinck, M., Goujon, P. (eds.) Self-Organization and Emergence in Life Sciences, pp. 227–247. Springer, Heidelberg (2006)
2. Copeland, J., Proudfoot, D.: Un Alan Turing Desconocido. Investigación y Ciencia 273, 14–19 (1999)
3. Sipper, M.: The Emergence of Cellular Computing. IEEE Computer 32(7), 18–26 (1999)
4. Nunes de Castro, L.: Fundamentals of Natural Computing: An Overview. Physics of Life 4, 1–36 (2007)
5. Bedau, M.: Artificial Life: Organization, Adaptation and Complexity from the Bottom-Up. Trends in Cognitive Sciences 7(11), 505–512 (2003)
6. Bedau, M., et al.: Open Problems in Artificial Life. Artificial Life 6(4), 363–376 (2000)
7. Braha, D., Minai, A., Bar-Yam, Y. (eds.): Complex Engineered Systems: Science Meets Technology. Springer Complexity, Cambridge (2006)
8. Maldonado, C.E.: Ciencias de la complejidad: Ciencias de los Cambios Súbitos. In: Odeón. Observatorio De Economía y Operaciones Numéricas, pp. 85–125. Universidad Externado de Colombia, Bogotá (2005)
9. Maldonado, C.E.: La heurística de la vida artificial. Revista Colombiana de Filosofía de la Ciencia II (4 & 5), 35–43 (2001)
10. Lane, N.: Power, Sex, Suicide. Mitochondria and the Meaning of Life. Oxford University Press, Oxford (2005)
11. Lloyd, S.: Programming the Universe. Alfred A. Knopf, New York (2006)
12. Kauffman, S.: Reinventing the sacred: A new view of science, reason, and religion. Basic Books, New York (2008)
13. Langton, C.: Studying Artificial Life with Cellular Automata. Physica 22D, 120–149 (1986)
14. Langton, C.: Artificial Life. In: Boden, M. (ed.) The Philosophy of Artificial Life, Oxford University Press, Oxford (1996)
15. Waldrop, M.: Complexity. In: The Emerging Science at the Edge of Order and Chaos. Touchstone, New York (1993)
16. Ray, T.: An approach to the synthesis of life. In: Langton, C., Taylor, C., Farmer, D., Rasmussen, S. (eds.) Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity, vol. XI, pp. 371–408. Addison-Wesley, Redwood City (1991)
17. Ray, T.: Jugué a ser Dios y Creé la Vida en mi Computadora. In: Gutiérrez, C. (ed.) Epistemología e Informática, pp. 257–267. UNED, Costa Rica (1994)
18. Gardner, M.: Mathematical Games: The fantastic combinations of John Conway's new solitaire game ``life". Scientific American 223, 120–123 (1970)
19. Chaitin, G.: Metamaths. In: The Quest for Omega. Atlantic Books, London (2006)

Observer Based Emergence of Local Endo-time in Living Systems: Theoretical and Mathematical Reasoning

Igor Balaz[1] and Dragutin T. Mihailovic[2]

[1] Scientific Computing Laboratory, Institute of Physics,
P.O. Box 57, 11001 Belgrade, Serbia
ibalaz@polj.ns.ac.yu
[2] Faculty of Agriculture, University of Novi Sad,
Dositej Obradovic Sq. 8,
21000 Novi Sad, Serbia

Abstract. In this paper, we analyze the temporal dimension of the problem of emergence of functional networks in living systems and give an abstract mathematical framework for dealing with those issues without considering the structure of the particular mappings or the resulting dynamics. We found that formal structures equivalent to emergence of local ordering and local chronologies can be defined within the framework of topological relations based on local observers. By using category theory, we further represented higher levels of transformations (between closures), with a special focus on observer turnover.

Keywords: Time, Emergence, Category theory, Observer, Artificial Life.

1 Introduction

The problem of understanding the flow of time has a long tradition and often presents inextricable difficulties derived from disputes in basic conceptual approaches. Demarcation points are numerous: reductionism *vs.* Platonism, different views of the topology of time, presentism vs. non-presentism, and so on [1, 2]. We do not strive to understand the philosophical considerations of the nature of time itself. Instead, we will deal only with the experience of time on the very elementary level of interactions of intracellular proteins. These proteins are the basis for the emergence of functional networks where transformations and regulations work together to create metabolism. Since each protein acts as a state machine with determined inputs which can be mapped into defined outputs, we will consider them as elementary observers. As observers they reside within some universe and are able to assimilate a segment of external changes with an appropriate set of internal operations to produce reactions. In addition, the operational environment is defined from the perspective of that individual so that external changes functionally exist for individuals only if they can be observed.

Starting from this framework, we will develop a theoretical and mathematical treatment of the emergence of these processes, their chronological organization and the possibilities of internal manipulations.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 222–229, 2011.

2 Theoretical Treatment

When considering the perception of time, changes in the observer's environment are only one basis upon which a perceptive entity or a group of entities can construct their own time flow. Such construction is established locally in three senses: (i) the act of observing is synonymous with the translation of a vast diversity of external physical and/or chemical changes into an "understandable" signal, so it is functionally local, (ii) the life span of observing agents are always limited (temporal locality) and (iii) spatial constraints are inherent to the act of observation. We consolidate these notions into a single idea under the term *local endo-time*.

However, perceptivity itself is not enough for the establishment of systemic time flow, since the prerequisite for establishing systemic time relations is the ability to compare different systemic states. As long as perceptivity is not incorporated into the network of systemic relations, the system will remain in a state of independent linear processes flows. If, according to Luhmann [3], we define a process as a mutually connected succession of events where the scope of selections is re-established at each stage, the specific characteristic is defined by anticipation as an accumulation toward less and less probable states (less probable from the perspective of the beginning of the process) which emerges as a necessary consequence of previous stages. However, there is a very important difference between elementary processes (e.g., separate enzymatic transformations) and the processes developed from them (e.g., metabolic pathways). The first ones can be considered as unambiguous transformations toward a determined state (in an ideal case) or a group of very similar states. However, in living systems, that kind of almost indispensable flow enclosed in a rigid structure is only a basis for the further development of functionality. These irreversible sequences can be rearranged into higher order structures (metabolic pathways) gradually relativizing process indispensability with every superposed level of functionality constructed. In such a structure, individual events are no longer necessary for continuing the line of transformations (which is the case with elementary processes), but become "one of" the possible realizations of functionality. In spite of such relativization, what remains inherently connected for these processes is their necessary differentiation: former/latter – as the primary form of local temporal differentiation.

Through this relativization of necessities, the system becomes able to construct anticipatory structures which, from the available (perceptively constructed) context, choose indicators that are in correlation with changes in the future[1], associating them with adequate systems of transformations and preparing themselves for the following events. From such a perspective it is obvious that mutual interactions of anticipatory structures are not only based on the possibility of perceiving signals, but also on the possibility of anticipating future states (e.g., establishment of regulations based on feedback). Such anticipations are inherently connected with systemic expectations in which realization or non-realization becomes a powerful intrasystemic regulative factor. Therefore, it is not only important to accomplish some function; equally important is temporal compatibility with other, parallel processes. Consequently, only

[1] As future changes in this context we consider only those which are within a scope of results of other systemic-induced transformations.

a combination of these two factors--the possibility of anticipation and regulation by anticipation--can create a basis for the construction of an autonomous, systemic time flow, as a generalization of the validity of partial intrasystemic time horizons across functional elements and (organizational) structures within certain subsystems. In other words, during interactions, along with perceptive normativity, subsystems also impute time (their own construction of time) to their environment. The organizational dynamics of mutual influences results purely from such imputations.

Although such relativizations make organizational manipulations available, we still cannot talk about systemic comparisons (since an abstract measurement of empty intervals in such systems is not possible) and hence, about systemic time. The final precondition is to establish parallel and multiple repetitions of transformations which are successively superpositioned. Then, the system enters a state where consequences of transformations are transferred to the following cycles of successive processes. In this way, the dynamics of cycles are not self-sustained but always exist with reference to previous operations in their environment; i.e., by a rudimentary “comparison” of intervals. Only then processes became autonomous axes for establishing systemic time relations, and consequently for construction of organizational regulations and controls.

However, at the same time, the physical duration of constitutive elements is limited (in this case it is called protein turnover) [4]. This means that pattern of observation, with all of the consequences described above, is not fixed to some location in space nor is the structure of functional relations invariable. Although the theory of autopoietic systems make it obvious that systemic elements should always be self-reproduced, [5] mathematical models of global regulations in organisms mainly neglect this fact and its consequences. With this in mind, it is necessary to pay particular attention to the development of functionality based on cyclic degradations and reconstructions of systemic material structures. First, it should be emphasized that it is not a mere repetitive circle that produces sameness, but rather a production with deviations, due to locality inherent to perception. The continuous decomposition of segments of processes compels them to be in constant reconstruction, thus making space available in the organizational structure of different insertions, divergences and re-routings without the need to construct specific mechanisms (in the form of localized regulators) for each specific case. In addition, through turnover, the system purposefully eliminates groups of elements, concordantly eliminating them from the possibility of direct reaction (with respect to other elements, subsystems, etc.). In this way, the system’s internal structure perpetually reconstitutes the causal basis for its own processes and the past is not merely a fixed set of preceding events, but is rather a dynamic accumulation where, according to its relevancy to the current state (a relevancy constantly updated with causal reconstitution) some elements and structures can be summoned while others disappear without any further functional influences. Through such cycles, the system is forced not to *be* self-adapted but rather *in* self-adaptation.

Finally, it is necessary to answer one more question: What is the main precondition for turnover of elements without destroying systemic functionality? It is obvious that since an organism’s survival is inherently connected with undisturbed metabolism function, turnover should not influence the continuity of metabolic processes. Is it justified then to assume the existence of elementary functional units which cannot be, and should not be perturbed? If we analyze metabolism as a whole, at each stage we can identify some segments that are essential for survival of the organism and that will be safeguarded from the possibility of internal violations (e.g., excessive synthesis of groups of enzymes coupled with a decrease in the level of the specific chaperone). However, determination of such "elementary units" is highly dependent on context: both materially (e.g., availability of certain nutrients, the constellation of environmental factors), as well as functionally (e.g., hierarchical variations regarding actual distribution of subsystems). Therefore, what is usually considered a central property of elementary units, namely their perseverance through different contexts, is lost. However, if we set our focus down to the level of concrete, material transformations, we can see that every single step in the processes of metabolic transformations is performed by enzymes whose actions are not susceptible to cutting or dividing into independent phases. In this manner, single enzymatic transformation is not only an elementary event, but is also an inherently unambiguous process—an atom of functionality. Only through the enclosure of single occurrences in a web of functionally meaningful events does it become possible for systems to base their functionality on recursive, reflexive reproduction of elements. The rise of elementary events from the level of discrete, meaningless occurrences to the level of finished processes, lays the groundwork for a situation where any kind of interruption (in the sense of physical elimination of functional elements) or rearrangement cannot violate the fundamentality of such units. Without that kind of organization, temporalizing constitutive elements of the systems would destroy them.

3 Mathematical Treatment

In the theoretical construction of the problem we raised several notions important for understanding emergence of endo-time in living systems and their consequences for the pattern of functional organization of living systems. Based on these considerations, we will develop an appropriate mathematical framework.

Definition 1 (*Relations*). If we denote by $M = \{f, g, h, ...\}$ the set of proteins and by $U = \{a, b, c, ...\}$ the set of environmental objects, then relations on U can be represented as

$$R = (U, \mathrm{G}(R)) \tag{1}$$

where $\mathrm{G}(R) : \{U_1 \times ... \times U_k \, \big| U_1, ..., U_j, ..., U_k \subset U\}$ is determined by M (in any given context internal determination of $\mathrm{G}(R)$ by U itself is not important) such that

$$a\mathrm{G}(R)b \, \big| a, b \in U, \mathrm{G}(R) \equiv f \in M \tag{2}$$

Set of all $a \in U$ is called the domain of the relation R and is denoted **dom***f* , while set of all $b \in U$ is called the codomain of the relation R and is denoted **cod***f* . Since the relation R is structure, preserving it can be called a morphism. The relation $G(R)$ is:

- irreflexive: $\neg aG(R)a$;
- symmetric: $aG(R)b \Rightarrow bG(R)a$;
- nontransitive: $aG(R)b \wedge bG(R)c \Rightarrow \neg aG(R)c$;
- with a unique resultant: $aG(R)b \wedge aG(R)c \Rightarrow b = c$.

Here, **dom***f* represents the observed subset of the environment, while *a*G(*R*)*b* incorporates the intrinsic unambiguity of elementary processes. In addition, it should be noted that being member of e.g., **cod***f* is context dependent: **cod***f* can became **dom***f* and *vice versa*.

Definition 2 (*Mapping into Shared Metric Space*). If both sets M and U can be mapped into the metric space (C,d) such that $C \equiv M \oplus U = \{(0,m) | m \in M\} \cup \{(1,u) | u \in U\}$ then C is a coproduct of M and U , while (C,d) is shared metric space over M and U .

Definition 3 (*Domain Determination*). For **dom***f* , $a \in$ **dom***f* iff there exist $a \in U$ such that:

- if we define a set of many-valued attributes $P = \{p,q,r,...\}, \forall p \in P, p = \{p_1, p_2, p_3,..., p_n\}$, and each $a \in U$ and $f \in M$ are associated with the $P_n^\wedge$, where $n \in \{M \cup U\}$, $P_n \subseteq P$, and $P_n^\wedge$ is a strictly ordered set $(P_n,<)$, where the ordering relation is determined in accordance with the underlying structure of a and f, then the Cartesian product ${}_vP_a^\wedge \times {}_vP_f^\wedge$ where ${}_vP_a^\wedge \subseteq P_a^\wedge$, ${}_vP_f^\wedge \subseteq P_f^\wedge$ represents visibility of a for f ;
- $d(f,a) = 1 \text{for} \, d(f,a) = \begin{cases} 0, & \text{if} \, |f-a| > |1/2\, p_f^r + 1/2\, p_a^r| \\ 1, & \text{if} \, |f-a| \leq |1/2\, p_f^r + 1/2\, p_a^r| \end{cases}$ where p_f^r and p_a^r are called the "activity radius" values for f and a, respectively, and are determined by the corresponding attribute values.

In other words, in order to be recognized, each element of the environment must be within the scope of a certain enzyme, and recognizable by it. All elements recognized in such manner are collected into an equivalence class $[a]_{\mathbf{dom}f}$.

Definition 4 (*Topology Based on Observations*). If the metrical radius $R(f)$ for **dom***f* , is defined as $R(f) = \{a,b | a \in \mathbf{dom}f, b \in \mathbf{cod}f \wedge d(a,f) = 1, d(b,f) = 1\}$, then the collection of all $R(f)$ for a given metric space (C,d) is a basis for the topology $\mathcal{T}$ on C . Therefore, a subset N of C is in $\mathcal{T}$ (is an open set) if it is a union of members of the collection of all $R(f)$. Such a topology $\mathcal{T}$ is induced by the metric d and $(C,\mathcal{T})$ is the induced topological space. It should be emphasized that each open set is defined with respect to a particular $f \in M$.

Definition 5 (*Emergence of Functional Closure and Boundaries*). In order to describe functional relations induced by local perceptions, we will focus our attention on the emergence of closure and boundaries for open sets in an induced topological space:

- for $R(f)$, a point $a \in C$ is defined as limit point of $R(f)$ if every open set of induced topology containing a contains a point of $R(f)$ different from a. The set of all limit points of $R(f)$ is denoted $R(f)'$. Then, $R(f) \cup R(f)'$ is called the closure of $R(f)$, denoted $\overline{R(f)}$. Within the given framework, $\overline{R(f)}$ signifies the emergence of metabolic processes.
- for an open set $R(f)$, the interior, denoted $\mathbf{int}(R(f))$, is the largest open set contained in $R(f)$. In other words, $\mathbf{int}(R(f))$ is $R(f)$ itself. All points in the closure of $R(f)$ not belonging to the $\mathbf{int}(R(f))$ define the boundary of $R(f)$ denoted $\mathbf{bd}R(f)$. An element of the boundary of $R(f)$ is called a boundary point of $R(f)$. Less formally, boundary points of $R(f)$ are those points which can be approached both from $R(f)$ and from the outside of $R(f)$. Within the given framework, $\mathbf{bd}R(f)$ signifies points of possible divergence or differentiation of emerging metabolic processes.

In order to have ordering, a relation over the set has to be transitive and asymmetric. However, Definition 1 states that relations induced by enzymes are nontransitive and symmetric. Therefore, it is clear that one single relation determined by $f \in M$ cannot induce ordering over the environment. Instead, we need to introduce a relation which would naturally follow from the existence of a number of different members of the set M within a given range.

Definition 6 *(Ordering).* If $a \in (\mathbf{bd}R(f) \wedge \mathbf{bd}R(g))$ and $b \in (\mathbf{bd}R(g) \wedge \mathbf{bd}R(h))$ then between a, b is established a causal influenceability with respect to $\overline{R(M)}, M = \{f, g, h, ...\}$ denoted $<_{\overline{R(M)}}$, which is transitive and asymmetric.

Definition 7 *(Chronology).* $a, b \in \bigcup \mathbf{bd}R(M), M = \{f, g, h, ...\}$ ordered by the relation $<_{\overline{R(M)}}$ can be naturally mapped into a finite index set I such that $I = \{i, j, k, ...\}, I \subset \mathbb{N}$. Then, index set I represents a chronology over closure. This chronology is not imposed onto the structure but emerges from the bottom as a consequence of established causal relations.

Definition 8 *(Locality of Chronology).* Chronology defined by the relation $<_{\overline{R(M)}}$ is local in the sense that due to Definitions 3 – 5 is applicable only to observable subsets of U determined by structures and metrics.

Therefore, the closure chronology of living systems is globally linear. However, if we try to broaden our scope beyond $\overline{R(M)}$ over which $<_{\overline{R(M)}}$ is established, it is nonlinear in the sense of Robb's axiom, [6] which states that *"For every element x, there is another element y such that neither $x > y$ nor $y > x$"*, and its strengthened form: $(\forall x)(\forall y)(\exists z)(y \prec x \quad z \prec x \wedge \neg(z = y \vee z \prec y \vee y \prec z))$ [7].

Definition 9 (*Category Based on Observations*). If we assume:

- a collection of closures of $R(M), M = \{f, g, h, ...\}$ for a given topology is a basis for collections of objects such that each object consists of a quadruplet $G = (A, O, \partial_0, \partial_1)$ where A is a set of directed edges defined by Eqs. (1) and (2), O is a set of objects whose members are equivalence classes $[a]_{\mathbf{dom}M}$, $[b]_{\mathbf{cod}M}$ where $M = \{f, g, h, ...\}$ and mappings $\partial_i (i = 0,1)$ from A to O such that ∂_0 is a source map that sends each directed edge to its source and ∂_1 is a target map that sends each directed edge to its target;
- homomorphisms exist between objects G such that if $D : G \rightarrow G'$, where $G = (A, O, \partial_0, \partial_1)$ and $G' = (A', O', \partial'_0, \partial'_1)$ then D consists of two maps $D(A) : A \rightarrow A'$ and $D(O) : O \rightarrow O'$ such that following diagram commute:

$$\begin{tikzcd}
A \arrow[r, "D_A"] \arrow[d, "\partial_i"] & A' \arrow[d, "\partial'_i"] \\
O \arrow[r, "D_O"] & O'
\end{tikzcd} \tag{3}$$

- an arrow is assigned to each object $\mathrm{id}_G : G \rightarrow G$ called the identity on G;
- the composition of homomorphisms is defined as an associative operation assigned to pairs of morphisms such that for $D : G \rightarrow G'$ and $E : G' \rightarrow G''$, a map $E \circ D : G \rightarrow G''$ is defined by $(E \circ D)(G) = E(D(G))$,

then the category Γ of graphs and their homomorphisms is defined over interacting agents residing within subjective environment.

Finally, using categories we are able to represent protein turnover. In category theory a forgetful functor is defined as a functor that "forgets" some or all of the structure of an algebraic object [8]. For example $U : Grp \rightarrow Set$ assigns to each element of $G \in Grp$ its underlying set and to each morphism $D : G \rightarrow G'$ the same function regarded just as a function between sets. Since in our framework the structure of an object is defined as a collection of closures determined by $R(M) | M = \{f, g, h, ...\}$, a function that "forgets" any f and the corresponding interior of $R(f)$ will accurately represent protein turnover, and at the same time will cause reorganization of the structure of subjective environment.

Definition 10 *(Forgetful Function).* For an object $G = (A, O, \partial_0, \partial_1), G \in \Gamma$ we define forgetful function Φ as a homomorphism such that in mapping

$\Phi : (A, O, \partial_0, \partial_1) \rightarrow (A^{'}, O^{'}, \partial_0^{'}, \partial_0^{'})$ one or more $R(M), M = \{f, g, h, ...\}$ are ignored. In $G^{'}$ equivalence classes $[a]_{\mathbf{dom}M}$, $[b]_{\mathbf{cod}M}$ are changed and the according structure of $G^{'}$ is reconfigured.

4 Conclusions

Organization of living systems is unique in the sense that it cannot at any moment passively exist, from the mere fact that it is already formed. They are not finished systems, but rather systems in continuous self-construction, which require them to always be in the process of adaptation to both the external and internal environment. The results presented here are only a part of the global framework which could be comprehensive enough to stands as an evolvable, self-organizing model of an abstract organism. Merging the presented framework with the internal coding system which would ensure that generation of observers is autonomously determined, will represent an important step toward that goal, which is a task for future work.

Acknowledgments. The research work described here has been funded by the Serbian Ministry of Science and Technology under the project “Modeling and numerical simulations of complex physical systems”, No. ON141035 for 2006-2010.

References

1. Newton-Smith, W.H.: The Structure of Time. Routledge & Kegan Paul, London (1980)
2. Zimmerman, D.: Persistence and Presentism. Philosophical Papers 25, 115–126 (1996)
3. Luhmann, N.: Soziale Systeme. In: Grundrisseiner allgemeinen Theorie. Suhrkamp Verlag GmbH, Frankfurt (1987)
4. Hershko, A.: Ubiquitin: roles in protein modification and breakdown. Cell 34, 11–12 (1983)
5. Zeleny, M. (ed.): Autopoiesis: A Theory of Living Organization. Elsevier North-Holland, Amsterdam (1981)
6. Robb, A.A.: The Absolute Relations of Time and Space. Cambridge University Press, Cambridge (1921)
7. Lucas, J.R.: The Conceptual Roots of Mathematics: An essay on the philosophy of mathematics. Routledge, New York (2002)
8. MacLane, S.: Categories for the Working Mathematician. Springer, New York (1998)

Life and Its Close Relatives

Simon McGregor and Nathaniel Virgo

Centre for Computational Neuroscience and Robotics
University of Sussex, Brighton, UK
nathanielvirgo@gmail.com

Abstract. When driven by an external thermodynamic gradient, non-biological physical systems can exhibit a wide range of behaviours usually associated with living systems. Consequently, Artificial Life researchers should be open to the possibility that there is no hard-and-fast distinction between the biological and the physical. This suggests a novel field of research: the application of biologists' methods for studying organisms to simple "near-life" phenomena in non-equilibrium physical systems. We illustrate this with some examples, including natural dynamic phenomena such as hurricanes and human artefacts such as photocopiers. This has implications for the notion of agency, which we discuss.

Keywords: Near-life, Thermodynamics, Dissipative Structures, Autopoiesis, Developmental Systems Theory.

1 Introduction

1.1 A-Life and Near-Life

The premise of Artificial Life is that there is more to being alive than the details of terrestrial biology; that there are abstract principles which underlie those particular instantiations of life we happen to share a planet with. In this paper we will argue that, besides "life as it could be" [6], we should also be considering "near-life as it is": life-like properties in the familiar inanimate world. Indeed, we will see that hurricanes and even photocopiers have thought-provoking features in common with living organisms.

When we use the word "life" in this paper, we do not refer to a definition but to a class of examples, namely those things existing on this planet which biologists consider alive: the eukaryotes, prokaryotes and occasionally viruses. Similarly, we use "life-like" informally to refer to various properties shared by all or most of these examples.

Central to our approach is the study of life-like phenomena that arise naturally within a wider system, either simulated or physical. We propose to study "near-living" systems in the same kind of way that a biologist studies living organisms: by observing them in their natural habitat; by studying their behaviour; and by trying to see how their anatomy functions. The processes behind life-like behaviour in non-living systems are likely to be much easier to understand than in living systems. Our purpose in this paper is not merely to point out similarities between some non-living systems and living systems, but to suggest that principled further studies of such similarities could be a gold-mine of new results.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 230–237, 2011.

1.2 Non-equilibrium Thermodynamics

Physical systems in thermal equilibrium are not life-like: the second law of thermodynamics means that isolated macroscopic systems tend towards more or less homogeneous equilibrium states. However, this does not apply to open systems in which an externally imposed "thermodynamic gradient" causes a continual flow of matter or energy through the system. In such *non-equilibrium* systems complex patterns can arise and persist. Authors such as Prigogine, Schneider & Kay and Kauffman [12,14,5] have observed that life is an example of such a "dissipative structure." Life exists within the Earth system, which is an open system with a gradient imposed primarily by the sun[1].

Living organisms exist within this system and are constrained by the second law of thermodynamics. Therefore many of their properties have to be realised in non-trivial ways. No organism can maintain its structure by isolating itself from the world: the need to "feed on negative entropy" (Schrödinger [15]) guarantees that interaction with the world is required in order to find food or other sources of energy. Ruiz-Mirazo and Moreno connect this to the theory of autopoiesis [13].

Simulation work in artificial life has often ignored thermodynamics; it is common to see research using physically unrealistic cellular automata, abstract artificial chemistries in which there is no quantity corresponding to entropy, or simulated agents which eat abstract "food" (if indeed they need to eat at all).

In contrast, our approach builds upon and extends previous authors' observations about the relationship between life and thermodynamics. An abiotic non-equilibrium system has many parallels with an ecosystem; our focus is on the equivalents to organisms that exist in such systems, and the specific properties that they do or do not share with biological organisms. These properties vary depending on the system under consideration, so our methodology is to investigate many different types of non-equilibrium system, looking for the general circumstances under which various life-like properties arise.

2 Some Examples of Near-Life

2.1 Hurricanes

A hurricane is a classic exemplar of a dissipative structure. It is instructive to focus on this example because hurricanes exhibit a phenomenon we call individuation. Emanuel [3] gives the following characterisation of a hurricane's operation:

"[Air] flows inward at constant temperature within a thin boundary layer [above the sea surface], where it loses angular momentum and gains moist entropy from the sea surface. It then ascends and flows outward to large radii, preserving its angular momentum and moist entropy. Eventually, at large radii, the air loses moist entropy by radiative cooling to space..." These processes occur simultaneously. Their rates balance so that the hurricane as a whole is stable.

[1] More precisely, the gradient is formed by the difference in temperatures between the incoming solar radiation and deep space.

The net result of the interactions between these processes is that the hurricane is formed and remains stable to perturbations. Moreover it is formed as a spatially distinct individual, separate from other hurricanes (this can be compared to the definition of autopoiesis by Maturana & Varela [8,9] as a "network of processes" that "constitute" a "unity.").

A hurricane has functionally differentiated parts: near the sea surface the water is drawn towards the eye, picking up moisture from the sea and rotational speed from the Coriolis force; in the eyewall itself the air is moving rapidly upward. Each part, together with its associated processes, is necessary for the whole to persist. This is analogous to an organism's anatomy.

A hurricane remains an individual entity because of its vortex structure, whereas a cell is surrounded by a membrane. We see this as the same phenomenon, individuation, occurring by different mechanisms. In both cases the result is that the system is localised in space and distinct from other individuals.

Although the research is so far preliminary, hurricanes may also be able to exhibit behaviour that could be called adaptive. Shimokawa et al. [16] claim that when the prevailing wind is subtracted from the data, tropical cyclones tend to move towards regions which are better able to sustain them, namely those with a greater temperature gradient between the sea and the upper atmosphere[2].

The hurricane has been given as an example of a "self-organising" system before but its similarities to living cells have not been studied in depth. We feel that such a study would provide many novel scientific insights.

2.2 Reaction-Diffusion Spots

Hurricanes are large, comparatively complex and difficult to study. A much simpler and easier-to-study example of an individuated dissipative system can be found in the patterns that form in reaction-diffusion systems [11]. Reaction-diffusion systems are very simple and easily simulated non-equilibrium chemical systems in which reactions take place among chemical species that are able to diffuse along a plane. The non-equilibrium conditions are maintained by continually adding reactants and removing products from the system. Under some parameter regimes the system can form a pattern in which there are spatially distinct "spots" of an autocatalytic substance, separated by regions in which no autocatalyst is present (see fig. 1, left).

We can observe interesting life-like properties in the form and behaviour of reaction-diffusion spots: like all dissipative systems they export entropy and are dependent on specific thermodynamic gradients; they are patterns in matter and energy; like organisms, they exist mostly as identifiable individuals; under certain parameter regimes they reproduce [11].

Studying reaction-diffusion spots according to our methodology involves treating a single spot as a model agent and studying it from a "spot-centric" point of view. The following simulation-based results have been demonstrated in [17],

[2] More strictly, higher Maximum Potential Intensity, a measure that takes into account both the temperature and pressure gradients that power hurricanes.

and a paper giving details of the experiments is in preparation. Briefly, we found that reaction-diffusion spots, perhaps like hurricanes, tend to move towards areas where more food is available, a behaviour that could be called "adaptive." We also found that individuated spots are very likely to arise when there is a negative feedback added between the whole system's activity and overall supply of food (this situation is common in natural systems and we think the result is suggestive of a possible general phenomenon). Using this as a method for producing individuated entities we were able to produce agents with a more complex 'anatomy' than just a single spot (fig. 1, right). Some of these more complex agents exhibited a very limited form of heredity in their reproduction.

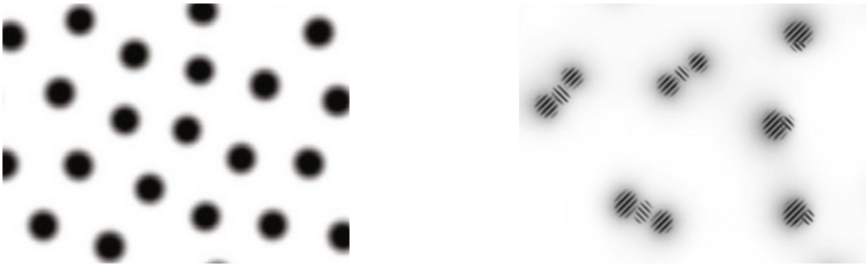


Fig. 1. *left*: "individuated" spots of autocatalyst in a reaction-diffusion system; *right*: structurally complex individuated entities in a reaction diffusion system. Each consists of spots of two different autocatalysts (represented with diagonal stripes in this image) coexisting symbiotically, mediated by an exchange of nutrients, one of which is shown in grey. Details will be published in a forthcoming paper.

2.3 Photocopiers

In this section we will examine a photocopier — a stereotypical inanimate object — from a point of view that might be called "enactive." That is, it is a point of view in which the photocopier itself is the central player, maintaining its identity and behaving adaptively as a result of dynamical interactions with its environment. This is a difficult viewpoint to take at first, since our intuition tells us that a photocopier is not the sort of thing that can "act." Rather, we normally think of it as acted upon by human beings. However, this intuition can be stretched, and we hope the reader will agree that it is worthwhile to do so.

We chose a photocopier for this example because it is usually repaired when it breaks down; any such machine would have done. In particular, the fact that the photocopier performs a copying task has no special significance.

We focus on this example because it shows how far our intuitions can be stretched without reaching a *reductio ad absurdum*: a photocopier is an archetypal example of an inanimate object, but when considered as an agent engaged in complex interactions with its environment it becomes in many ways the most life-like of our three examples.

When trying to support our natural intuition of a qualitative difference between photocopiers and bacteria, we may cite a variety of apparently relevant facts. Bacteria have DNA, and we can observe the complex process of bacterial reproduction under the microscope, whereas the same is not true of photocopiers. A photocopier, unlike a bacterium, consists of mostly static and chemically inert parts; if a part of the photocopier becomes damaged, it does not reconstitute itself internally. Bacteria display complex adaptive behaviour including chemotaxis and habituation; photocopiers don't appear to.

This may seem like a conclusive list of differences between a cell and a photocopier. And some of these differences are genuine, if perhaps arbitrary: photocopiers do indeed lack DNA. However, on closer scrutiny some of the other issues will turn out to be less clear-cut. We think that many if not all of the most significant differences between cells and photocopiers can be seen as differences of degree rather than kind.

Dynamic Identity. One prototypical property of bacteria and other living organisms is their identity as patterns of matter and energy. Individual atoms flow through the organism, and the overall organism is maintained even if all the material parts change. We can consider a photocopier that has this property, although the rate of material turnover is much slower than in a cell: when the parts of this photocopier break they are replaced by an engineer. Like the hammer which retains its identity despite having a new handle and a new head, matter flows through the photocopier leaving its photocopier-ness unchanged.

In ordinary discourse, we would not describe the process as *self*-repair, since we prefer to locate causal primacy in the engineer rather than the photocopier. But seen from a logical point of view, both photocopier and engineer are necessary parts of the physical process; the repair is caused by the interaction between the two. It is no different for bacteria: ongoing cell repair is caused by the interaction between the cell and its environment, since the organism must be able to absorb relevant nutrients and excrete waste.

Note that some important physical principles can be observed in action here. In order for this process to continue, the photocopier's environment has to be in a very specific state of thermodynamic disequilibrium: it has to contain appropriately competent and motivated engineers.

From a more photocopier-centric point of view one could say that the photocopier causes the repair to be carried out: simply by performing a useful office task, the photocopier is able to co-opt the complex behaviour of humans in its environment in just such a way that the raw materials needed to maintain its structure are extracted from the ground, fashioned into the appropriate spare parts and correctly installed. Seen from this perspective the photocopier is a master of manipulating its environment. It needs no deliberative intelligence to perform these feats, however: one is reminded of species of orchid that cause their pollen to be spread by mimicking the form of a female bee.

Many of these processes take place outside the physical (spatial) bounds of the photocopier itself, with most of them involving human activity in some way. This is in contrast to the usual conception of an organism, since these systems rely heavily on a network of metabolic or dynamical processes that occur within the system's physical boundary. We argue however that this difference between artefacts and organisms is one of degree rather than a difference in kind, since all organisms must rely on some processes that are external to their spatial boundary, some of which will often involve the action of other organisms.

Individuation. The phenomenon of individuation is evident in this example: photocopiers are maintained as individual photocopiers. Half a photocopier will not function as one and will not be maintained as one. We can imagine an environment in which half a photocopier might be maintained: we might find one inhabiting a display case in a museum, for example. But in this case it is being maintained as a museum exhibit rather than as a photocopier: it would be a different species of artefact.

Reproduction and Evolution. Although there are obvious differences between the process of "evolution" in photocopiers and in living organisms, we can still observe some similarities. The photocopier phenotype has become better adapted to its ecological niche over time, as successful photocopiers are re-produced in factories and successful designs retained and modified. A photocopier's external casing does not contain its blueprints, whereas we often regard the "design" for a cell as being in its DNA (with developmental influences from the internal and external environment). But this notion is perspectival rather than factual: as Oyama [10] points out, the interaction of environment and genome forms the cell, with both being required. Reproduction and development in *any* organism relies heavily on environmental machinery external to the organism itself. The photocopier phenotype interacts with its blueprint indirectly through the phenotype's effects on photocopier sales, which fund the production of more photocopiers from the blueprint. In turn the sales depend on the successful operation of the photocopiers, among other factors.

3 Discussion

When we picture something which symbolises the fundamental properties of a living being, the chances are good that we imagine a biological object: a living organism such as a cell, plant or animal, or perhaps the DNA helix. Alternatively, we may think of our favourite simulation model in Artificial Life. It is unlikely in the extreme that we picture an inanimate object such as a rock, a hurricane or a photocopier. Due in part to our evolutionary heritage, humans have a definite sense of what is alive and what is not.

We assert that scientists should treat this intuition with suspicion. In the past, it has given rise to erroneous theories of an *animal spirit* or *elan vital* as an explanation for the remarkable behaviours of living organisms. Although these ideas have long been discarded by formal science, the underlying psychological stance is more enduring, and it has led science astray in the past.

In this sense, we echo Oyama's insights in *Developmental Systems Theory*(e.g. [10]): human beings like to postulate chains of effect which attribute causal primacy to a particular part of a holistically integrated system. In the scientific imagination, part of the system becomes the *agent* and the rest is the *environment.*

Oyama argues that attributing causal primacy to genes is a conceptual error within developmental biology; we contend that similar problems arise when we think about living and non-living systems.

Thermodynamic gradients can be found nearly everywhere in the physical universe, and consequently we should not be surprised if near-life is abundant in interesting forms. By "near-life", we simply mean non-biological systems which share important characteristics with living organisms, including any of the following: reproduction, particularly with heritable variation; maintenance of a dynamic pattern of matter and energy; production of spatially separated individuals; "goal-directed" behaviour. All of these properties can be observed in comparatively simple non-equilibrium systems; there is every reason to suppose we will find more such properties. It is quite possible, for example, that one could find life-like structures that exhibit a developmental trajectory over their existence; or show a permanent memory-like change in behaviour in response to a stimulus; or have a more complex form of heredity.

It is not a new idea to look for life-like processes in non-biological systems. Lovelock [7] describes the entire Earth system as a "single living system", using terms such as "physiology" and "anatomy." Our ambitions concern simple and physically numerous systems which can be studied experimentally.

While many of the similarities between life and dissipative structures are well-known, most previous commentators (e.g. Prigogine, Schneider & Kay [12,14]) have considered only natural (as opposed to artificial) phenomena. However, artificial structures are part of the same physical world as natural structures. Our perspective allows us to observe physical characteristics associated with biology even in apparently prototypical inanimate objects such as photocopiers. Of course, it is not a new idea to attribute biological properties to cultural artefacts (this occurs in "meme theory," which concentrates on "selfish replicator" properties, e.g. Blackmore [1]) or to consider physical artefacts as an integral part of a biological system (Clark & Chalmers [2]). We add the observation that ordinary physical artefacts can also be a type of life-like dissipative structure, providing they exist in a human environment which maintains them against decay. This is particularly instructive because it illustrates the extent to which our intuitions can be stretched without breaking.

3.1 Proposed Future Research

Searching for life-like properties of phenomena that arise in physical rather than computational systems is an under-researched area of A-Life that has potential for some very important results. Current "wet" A-Life research (i.e. *in vitro* chemical experiments), tends to focus on the deliberate design of life-like structures (e.g. synthetic bacteria [4] or formation of lipid vesicles), rather than on open-ended observation of structures that form naturally. However, interesting behaviour can also be observed spontaneously in non-equilibrium systems, including ones which are simple enough for physically realistic computer simulation (e.g. [17] and forthcoming work). More research along these lines would help to bridge the gap between biology and physics.

Acknowledgments. This paper arose from discussions in the Life And Mind seminar series at the University of Sussex. The support of Inman Harvey has been especially helpful.

References

1. Blackmore, S.: The Meme Machine. Oxford University Press, Oxford (1999)
2. Clark, A., Chalmers, D.: The extended mind. Analysis 58, 10–23 (1998)
3. Emanuel, K.A.: The dependence of hurricane intensity on climate. Nature 326, 483–485 (1987)
4. Gibson, B., Hutchison, C., Pfannkock, C., Venter, J.C., et al.: Complete chemical synthesis, assembly, and cloning of a mycoplasma genitalium genome. Science 319(5867), 1215–1220 (2008)
5. Kauffman, S.: Investigations. Oxford University Press, US (2000)
6. Langton, C.: Introduction: Artificial life. In: Langton, C. (ed.) Artificial Life, Santa Fe Institute, Los Alamos, New Mexico (1989)
7. Lovelock, J.: Gaia: a New Look at Life on Earth. Oxford University Press, Oxford (1987)
8. Maturana, H.R., Varela, F.J.: Autopoiesis and Cognition: The Realization of the Living. Kluwer Academic Publishers, Dordrecht (1980)
9. Maturana, H.R., Varela, F.J.: The Tree of Knowledge: The Biological Roots of Human Understanding. Shambhala Publications, Boston (1987)
10. Oyama, S.: The Ontogeny of Information: Developmental Systems and Evolution. Duke University Press, Durham (2000)
11. Pearson, J.E.: Complex patterns in a simple system. Science 261, 189–192 (1993)
12. Prigogine, I.: Time, structure and fluctuations. Science 201(4358), 777–785 (1978)
13. Ruiz-Mirazo, K., Moreno, A.: Searching for the roots of autonomy: the natural and artificial paradigms revisited. Communication and Cognition–Artificial Intelligence 17, 209–228 (2000)
14. Schneider, E.D., Kay, J.J.: Life as a manifestation of the second law of thermodynamics. Mathematical and Computer Modelling 19(6-8), 25–48 (1994)
15. Schrödinger, E.: What is Life? Cambridge University Press, Cambridge (1944)
16. Shimokawa, S., Kayahara, T., Ozawa, H., Matsuura, T.: On analysis of typhoon activities from a thermodynamic viewpoint (poster). In: Asia Oceania Geosciences Society 4th Annual Meeting, Bangkok (2007)
17. Virgo, N., Harvey, I.: Reaction-diffusion spots as a model for autopoiesis (abstract). In: Bullock, S., Noble, J., Watson, R., Bedau, M.A. (eds.) Artificial Life XI: Proceedings, Winchester, UK, p. 816. MIT Press, Cambridge (2008)

A Loosely Symmetric Model of Cognition

Tatsuji Takahashi[1], Kuratomo Oyo[1], and Shuji Shinohara[2]

[1] Tokyo Denki University, School of Science and Engineering,
Hatoyama, Hiki, Saitama, 350-0394, Japan
tatsujitakahashi@me.com
http://web.me.com/tatsuji2_takahashi_/

[2] Advanced Algorithms & Systems, Co. Ltd., Ebisu IS bldg. 7F,
Ebisu 1-13-6, Shibuya, Tokyo 150-0013, Japan

Abstract. Cognitive biases explaining human deviation from formal logic have been broadly studied. We here try to give a step toward the general formalism still missing, introducing a probabilistic formula for causal induction. It has symmetries reflecting human cognitive biases and shows extremely high correlation with the experimental results. We apply the formula to learning or decision-theoretic tasks, n-armed bandit problems. Searching for the best cause for reward, it exhibits an optimal property breaking the usual trade-off between speed and accuracy.

Keywords: cognitive bias, heuristics, multi-armed bandit problems, stimulus equivalence, mutual exclusivity.

1 Introduction

Recently, cognitive biases and heuristics have attracted widespread attention, as typically in psychology and behavioral economics (e.g. [1–4]). Logicality and rationality must be distinguished, and we are required to inquire their origin, development and the relationship between them. Considering the effects of the biases may be the key to construct artificial agents. It is interesting to see that animals do not appear to have some of these illogical biases. It means that in some senses animals are more logical, or machinelike, than us. There is considerable circumstantial evidence that the biases deeply involve the difference between human and other animals.

In this study, we try to show the importance of loosely symmetric cognition through a constructive approach. We introduce a certain formula as an elementary model of causal inductive cognition and apply it to decision making or reinforcement learning. The formula, a loosely symmetric (LS) model [5, 6], implements key biases in a flexible way, autonomously adjusting the bias intensity according to the situation. The derivation of LS is given, along with its best humanlike behavior and high performance. The model's descriptive validity for causal induction, beating all tens of preexisting models, is shown. Its high performance is presented in binary bandit problems (BBP), a simplest class of decision theoretic tasks. BBP are the simplest dilemmatic situations where an

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 238–245, 2011.

agent has difficulty in balancing exploration (searching for better options) and exploitation (persisting to the known best option). The LS model is shown to break the speed-accuracy trade-offs resulting from the dilemma. Even though it has no parameter, it gives better performance than existing standard models like the epsilon-greedy softmax methods. First we introduce our LS model in the context of causal induction.

2 Human Biases in Causal Induction and *LS* Model

It is critical for survival to inductively reason a causal relationship from event co-occurrence. In psychology, symmetrical biases are considered to be crucial in general [1]. It is also argued that the biases enhance information gain from the co-occurrence [4]. Stimulus equivalence and mutual exclusivity have been originated in behavior analysis [2] and language acquisition [3]. They can be generalized to the inferential biases that we call symmetry (S) and mutual exclusivity (MX) biases. From a conditional $p \rightarrow q$, S and MX biases respectively induces $q \rightarrow p$ and $\bar{p} \rightarrow \bar{q}$ (where $\bar{p}$ means the negation of p). These conditionals correspond to the following three conditions that are important to our intuition of causality, that p is the cause of q:

1. High conditional probability of q given p, $P(q|p)$.
2. Simultaneously high
   (a) "converse" probability $P(p|q) \approx P(q|p)$ (S bias) and
   (b) "inverse" probability $P(\bar{q}|\bar{p}) \approx P(q|p)$ (MX bias).

Condition 1 is fundamental in prediction. However, it is not directly connected to causality. Even if the sun rises always after ravens croak, it does not mean that the croaking is the cause of the sunrise.

2.1 The Models

Two events co-occurrence or covariation information is expressed in a 2×2 contingency table (Table 1). The cells a, b, c and d respectively mean the joint probabilities $P(p, q), P(p, \bar{q}), P(\bar{p}, q)$ and $P(\bar{p}, \bar{q})$.

Table 1. A 2 x 2 contingency table for causal induction

		posterior event	
		q	$\bar{q}$
prior event	p	a	b
	$\bar{p}$	c	d

Table 2. The completely symmetric contingency table for RS

		posterior event	
		q	$\bar{q}$
prior event	p	$a+d$	$b+c$
	$\bar{p}$	$c+b$	$d+a$

If it is just to predict a posterior event q from the occurrence of a prior event p, conditional probability (CP) suffices.

$$CP(q|p) = P(q|p) = P(p, q)/P(p) = P(p, q)/(P(p, q)+P(p, \bar{q})) = a/(a+b). \quad (1)$$

However, it does not satisfy the conditions (2a) nor (2b). They are satisfied by a transformation of the information. If we identify the b and c, the first condition

$$CP(q|p) = a/(a+b) = a/(a+c) = CP(p|q) \tag{2}$$

is satisfied. In a similar way, the second condition holds if a and d mean the same thing as well:

$$CP(q|p) = a/(a+b) = d/(d+c) = CP(\bar{q}|\bar{p}). \tag{3}$$

The transformation $P(p,q) \rightarrow P(p,q) + P(\bar{p},\bar{q})$ satisfies the identities; the result is in Table 2. Denoting the probabilities on Table 2 by $\widehat{P}(p,q) = a+d$ and so on, we get a completely biased model RS (rigidly symmetric) as:

$$RS(q|p) = \widehat{P}(p,q)/(\widehat{P}(p,q) + \widehat{P}(p,\bar{q})) = (a+d)/((a+d)+(b+c)). \tag{4}$$

RS satisfies both of the two symmetric conditions, 2a and 2b. It is important to see that there is another way to get it: it is to change how to refer to the untouched experience in Table 1. Since $\Sigma_{u,v}P(u,v) = 1.0$,

$$RS(q|p) = \big(P(p,q) + P(\bar{p},\bar{q})\big)/\Sigma_{u,v}P(u,v) = P(p,q) + P(\bar{p},\bar{q}). \tag{5}$$

In this form, the diagonal elements a and d are summed and the sum is divided by the whole, $a+b+c+d = 1.0$. Conversely, CP can be expressed in this reference form. Ignoring c and d cells by giving the coefficient 0 in $CP(q|p)$ realizes

$$CP(q|p) = \big(a + 0 \cdot d\big)/\big(a + b + 0 \cdot c + 0 \cdot d\big). \tag{6}$$

It is to ignore $\bar{p}$, the information in the absense of p. Conversely, for calculating $CP(q|\bar{p})$, it is required to neglect a and b, the information in the presence of p.

$$CP(q|\bar{p}) = \big(c + 0 \cdot b\big)/\big(c + d + 0 \cdot a + 0 \cdot b\big) \tag{7}$$

While the focal shift from p to $\bar{p}$ changes the cells in the contingency table from $a, b, 0, 0$ to $0, 0, c, d$ and vice versa, the value of the unfocussed information is trivially conserved as zero. This is *figure-ground segregation* that is essential to cognition [8]. While the figure information is straightly used, the ground is treated in an invariant way, against focal shift. It is *ground-invariance*.

2.2 The *LS* Model

Here we introduce our loosely symmetric (LS) model.

The Conditions for the Model. LS is defined as a probabilistic formula satisfying the following conditions:

(i) (Loosely symmetric) LS is not always rigidly biased.

(ii) (Flexible) However, its bias intensity is flexibly adjusted according to the situation, that is here (a, b, c, d).

(iii) (Cognitive) It reflects a cognitive constraint or ability: figure-ground segregation and the ground invariance.

Condition (i) can be satisfied by a family of intermediate models between CP and RS, PS (parametrically symmetric) that has two parameters $0 \leq \alpha, \beta \leq 1$. We adopt it as a prototype of our LS model.

$$PS_{\alpha,\beta}(q|p) = (a + \beta d)/((a + \beta d) + (b + \alpha c)). \tag{8}$$

However, PS can not satisfy neither condition (ii) nor (iii), as far as α and β are constants. Condition (iii) demands CP-like weakening of ground information, but condition (ii) requires α and β to be a function of the situation, here a, b, c, d. So we determine $\alpha(q|p) = \alpha(a, b, c, d)$ and $\beta(q|p) = \beta(a, b, c, d)$ so as to satisfy condition (iii). The condition is to equate αc and βd in Table 3 with αa and βb in Table 4, that are all in the row of the unfocussed event in the two tables. Constants can not be the solution, so the problem is to find functions α and β that satisfy $\alpha(q|p)c = \alpha(q|\bar{p})a$ and $\beta(q|p)d = \beta(q|\bar{p})b$. Because $0 \leq \alpha, \beta \leq 1$, probability formulae are suitable. The numerator of $\alpha(q|p)$ and $\alpha(q|\bar{p})$ can respectively be $a = P(p, q)$ and $c = P(\bar{p}, q)$ to make the numerator of the both terms ac. The denominators are the ones invariant against the change between p and $\bar{p}$. The sum of a and c satisfies the condition and hence $\alpha(q|p) = a/(a+c) = P(p|q)$ and $\alpha(q|\bar{p}) = c/(c + a) = P(\bar{p}|q)$. Similarly, $\beta(q|p)$ is determined to be $b/(b + d) = P(p|\bar{q})$ and $\beta(q|\bar{p}) = d/(d + b) = P(\bar{p}|\bar{q})$. Now we get the following formula:

Table 3. The table for $PS(q|p)$, when focussed on p

	q	$\bar{q}$
⇒ p	a	b
$\bar{p}$	αc	βd

Table 4. The table for $PS(q|\bar{p})$, when focussed on $\bar{p}$

	q	$\bar{q}$
p	αa	βb
⇒ $\bar{p}$	c	d

$$LS(q|p) = \frac{a + P(p|\bar{q})d}{a + P(p|\bar{q})d + b + P(p|q)c} = \frac{a + \frac{b}{b+d}d}{a + \frac{b}{b+d}d + b + \frac{a}{a+c}c}. \tag{9}$$

As we see in Table 3, the relation (ratio) between a and c, $a/(a+c)$, is recursively or self-referentially introduced as the coefficient α to c.

2.3 Results

We here introduce two models for comparison. One is called contingency (DP) model. [9] The other is the presently best dual-factor heuristics (DH) model. [10] They are defined as follows:

$$DP(q|p) = P(q|p) - P(q|\bar{p}) \tag{10}$$

Table 5. The determination coefficient r^2 of the model with human estimations

	CP	DP	DFH	RS	LS
H03 [10]	0.000	0.000	0.964	0.158	0.969
AS95 [11]	0.823	0.781	0.905	0.761	0.904
WDK90 [12]	0.944	0.884	0.961	0.888	0.969

$$DH(q|p) = \sqrt{P(q|p)P(p|q)} \tag{11}$$

In Table 5, we compare the estimations by the models with the average human ratings data from three experiments available in the literature. We can see that LS describes the data best as well as DH, while DH has been comprehensively shown the superior descriptive power compared to existing 41 models. [4]

3 Application to Reinforcement Learning

The findings just from contemplation can not be definitive. Some experiment or intervention is needed to identify a conclusive. Relationships inferred from causal induction must be then utilized in decision making. So we apply the models to BBP. We can apply the causal inductive models to decision making by reading p and $\bar{p}$ as "trying A" and "trying B", and q and $\bar{q}$ as "win" and "lose", in Table 1. The models calculate the value of the options (action value). The option o to choose can be determined by a model M as:

$$o = \underset{o' \in \{p, \bar{p}\}}{\mathrm{argmax}} M(q|o'). \tag{12}$$

This way of managing action values for decision making is called the *greedy method*.

3.1 The Settings for the Simulation

The simulation is executed in the following framework.

Initial Condition. We need some stipulations for the initial setting. To avoid the exception "division by zero", we define the initial covariation information a, b, c and d as respectively a uniform random number from $[0, 1]$. They are here not the joint probability but the frequency, so 1 is added to a when it wins trying A, and so on. The adoption is justified by the fact that the change of initial condition setting preserves the ordering of the results (correct rate, explained later) by the models. We could uniformly set a, b, c and d as $a = b = c = d = \theta$ where the θ may be 0, 1, 0.5 or $\ll 0.1$ or try all the options in the first stage.

Problems. A binary bandit problem is totally defined by a pair of two probabilities, $(P_A, P_B) = (\widetilde{P}(q|p), \widetilde{P}(q|\bar{p}))$. We generated the problems as follows.

$$\widetilde{P}(q|p) = \Sigma_{i=1}^{n} rnd_i/n \tag{13}$$

where rnd_i is a uniform random number from $[0, 1]$. We fix the parameter n controlling the variance to 6. $P(q|\bar{p})$ is determined in the same fashion.

Index. As the index for the performance of the models, we adopt the correct rate, the percentage of choosing the optimal option $o = \underset{o' \in \{p,\bar{p}\}}{\mathrm{argmax}} \widetilde{P}(q|o')$, that has the highest win probability, in all $1,000,000$ simulations.

3.2 Non-deterministic Models

Here we introduce some non-deterministic models for the bandit problems that are standard in reinforcement learning.

Epsilon-Greedy Methods. There are three ways to enhance the effectivity of initial search by probability.

- *Epsilon-first* (EF) *method.* Randomly choose an option in the initial n steps.
- *Epsilon-constant* (EC) *method.* Randomly choose an option at the constant probability ϵ through the whole series of trials.
- *Epsilon-decreasing* (ED) *method.* ϵ in EC method gets decreased according to a monotonously decreasing function. Here we define the function as

$$\epsilon(t) = 0.5/(1.0 + \tau t). \tag{14}$$

All the above methods, EF, EC and ED show a tradeoff that corresponds to the exploration-exploitation dilemma, as shown in Figure 1-4. The parameters for the methods are n for EF, ϵ for EC, and τ for ED.

Softmax Method (SM). The softmax action selection rule has a different way to manage the action values calculated by probabilistic formulae. It determines the probability of choosing an option according to the following formula that uses the Gibbs distribution:

$$\hat{P}(q|r) = \frac{exp(M(q|r)/\tau)}{\Sigma_{s \in \{p,\bar{p}\}} exp(M(q|s)/\tau)}, \forall r \in \{p, \bar{p}\}. \tag{15}$$

Note that $\hat{P}(q|p)$ is the *probability* of choosing the option p, not like other methods that compare $M(q|r)$ for all r in *argmax* function. τ is temperature.

3.3 Results and Discussion

Among five causal models in Table 5, LS performs the best, with RS just slightly worse, with the greedy method. CP and DP give the same result and DH chooses the optimal option at the 15th and 2000th step with the same percentage 55%. CP, LS and RS are employed in the probabilistic methods. The result by ED method in Figure 1 most prominently show the trade-off between the initial (short-term) and the final (long-term) rewards. The average correct rate of CP, LS and RS by the non-probabilistic greedy method are plotted as well. The parameter τ is moved in $\{0.01, ..., 1.5\}$ (21 variations). The arrow direction means

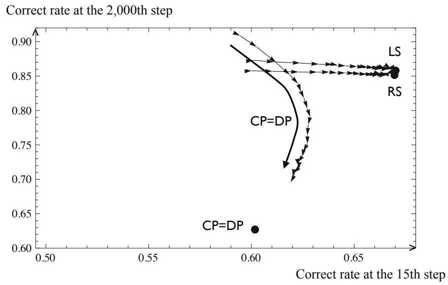


Fig. 1. Epsilon-decreasing method

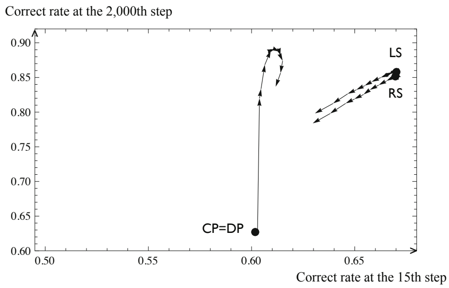


Fig. 2. Epsilon-constant method

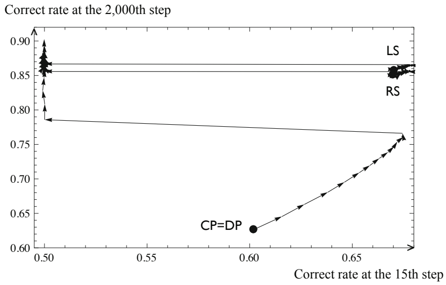


Fig. 3. Epsilon-first method

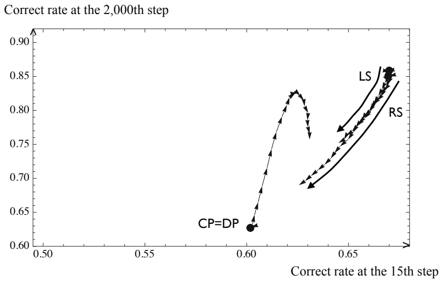


Fig. 4. Softmax method

the increase of τ. Infinitely large τ conforms ED to the greedy method. The change of the method from greedy to ED does not enhance the performance of LS and RS. It means that they deterministically prosecute the optimal balancing of exploitation and exploration. Other methods, EC, EF and SM are shown in Figure 2, 3 and 4. As for the parameters, $\epsilon \in \{0.01, ..., 0.25\}(11)$ for EC, $n \in \{1, ...120\}(21)$ for EF, and $\tau \in \{0.002, ..., 0.1\}(21)$ for SM.

4 Concluding Remarks

We have introduced a causal inference model, LS, and shown its descriptive validity for human causal induction. The model, when applied to reinforcement learning tasks, has shown its optimal nature. Although we here limited the testing framework elementary one to make the result general, it is easy to generalize. The reward was restricted to binary values: win or lose, but it can be given from a distribution. Some generalizations of LS toward another directions can be executed. Preserving the ground-invariance, a LS formula for n candidate causes or options can be obtained [6]. For n options or causes, we can derive a formula from the original LS with formal definition of probability. See Appendix.

LS formula can be applied to every area having use of conditional probability. We can implement it in a multi-agent system and see what it changes. It might show some characteristics that human society exclusively possesses. Because it also loosely satisfies many important relations such as Bayes' rule, we will be able to utilize it in practical uses with Bayesian filter and so forth.

Acknowledgments. The authors would like to thank Research Institute for Science and Technology of Tokyo Denki University for research grant 0509020001670, Gakushuin University Computer Centre for research project grant 2009, and Japan Women's University for research project grant 41-2009.

References

1. Gilovich, T., Griffin, D.W., Kahneman, D. (eds.): Heuristics and Biases: the Psychology of Intuitive Judgement. Cambridge University Press, Cambridge (2002)
2. Sidman, M., Tailby, W.: Conditional Discrimination vs. Matching to Sample: an Expansion of the Testing Paradigm. J. Exp. Anal. Behav. 37, 5–22 (1982)
3. Markman, E.M., Wachtel, G.F.: Children's Use of Mutual Exclusivity to Constrain the Meaning of Words. Cognit. Psychol. 20, 121–157 (1988)
4. Hattori, M., Oaksford, M.: Adaptive Non-Interventional Heuristics for Covariation Detection in Causal Induction: Model Comparison and Rational Analysis. Cognitive Sci. 31, 765–814 (2007)
5. Shinohara, S., Taguchi, R., Katsurada, K., Nitta, T.: A Model of Belief Formation Based on Causality and Application to N-armed Bandit Problem. Trans. Jpn. Soc. Artif. Intell. 22(1), 58–68 (2007)
6. Takahashi, T., Nakano, M., Shinohara, S.: Cognitive Symmetry: Illogical but Rational Biases. Symmetry: Culture and Science 21(1-3), 275–294 (2010)
7. Takahashi, T., Shinohara, S., Oyo, K., Nishikawa, A.: Cognitive Symmetries as Bases for Anticipation: a Model of Vygotskyan Development Applied to Word Learning. Int. J. Comp. Anticip. Sys., 24 (in press)
8. Parkin, A.J.: Essential Cognitive Psychology. Psychology Press, Sussex (2000)
9. Jenkins, H.M., Ward, W.C.: Judgment of Contingency between Responses and Outcomes. Psychol. Monogr. - Gen. A. 79, 1–17 (1965)
10. Hattori, M.: Adaptive Heuristics of Covariation Detection: A Model of Causal Induction. In: Proc. of ICCS/ASCS 2003, vol. 1, pp. 163–168 (2003)
11. Anderson, J.R., Sheu, C.F.: Causal Inferences as Perceptual Judgements. Mem. Cognition 23(4), 510–524 (1995)
12. Wasserman, E.A., Dorner, W.W., Kao, S.F.: Contributions of Specific Cell Information to Judgments of Interevent Contingency. J. Exp. Psyhol. Learn. 16, 509–521 (1990)

Appendix

LS model considers only "two candidate causes" p and $\bar{p}$. If there are n prior events p_i ($i \in \{1, 2, ..., n\}$), the contingency table becomes $n \times 2$ and the formula is defined as follows, if we define $a_i = P(p_i, q)$ and $b_i = P(p_i, \bar{q})$:

$$LS(q|p_i) = \frac{a_i + P(p_i|\bar{q})(\Sigma_{j \neq i} b_j)}{a_i + P(p_i|\bar{q})(\Sigma_{j \neq i} b_j) + b_i + P(p_i|q)(\Sigma_{j \neq i} a_j)}. \tag{16}$$

There is yet another consistent generalization to $n \times m$ contingency table. [7]

Algorithmic Feasibility of Entity Recognition in Artificial Life

Janardan Misra

HTS Research
#151/1, Doraisanipalya, BG Road,
Bangalore 560026, India
janardan.misra@honeywell.com

Abstract. Observations are an essential implicit component of the simulation based artificial-life (ALife) studies by which entities are identified and their behavior is observed to uncover higher-level "emergent" phenomena. Building upon the axiomatic framework of Henz&Misra [2], we analyze computational complexity bounds for the algorithmic implementation of an observation process for an automated discovery of the life-like entities in arbitrary ALife models. Among other results of such analysis is the conclusion that the problem of entity recognition in a simulation using syntactic constraints is a NP-hard problem and therefore cannot always be solved in polynomial number of steps. The computational complexity bounds are established distinguishing further between those ALife models which allow entities with overlapping structures to coexist in a state and others which do not.

1 Background

The very identification of life is still an existential problem in ALife studies since in general there is no known method to decide beforehand the kind of entities, which might emerge demonstrating non-trivial life-like behavior, without closely observing the simulations. An important aspect where ALife studies demand increasing focus is the *automated discovery* of life-forms and their dynamics in the simulated environments. Since ALife studies mostly involve digitized universes and their simulations, it is actually desirable to explore by algorithmic means potentially varied possibilities which these simulations hold yet usually require such detailed observations that it may not always be feasible to carry out for human observers alone. Such an automated discovery of life-forms and the evolving dynamics may bring much promise in ALife studies as compared to what could possibly be achieved only with manually controlled observations. An example of such an automated discovery is discussed by Samaya in [1]. In order to identify the living loops in his CA model, another "Observer CA" system is designed and embedded within the original CA simulator software. The observer CA can perform complex image processing operations on the input configuration given to it by the simulator CA to automatically identify the living loops of different types.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 246–253, 2011.

However, because of its implicit nature and the multitude of ALife models, a precise characterization of the observation process is generally a difficult problem. Importantly any sufficiently generic framework for studying the observational processes needs to be defined independent of the low-level micro dynamics or the "physical laws" of the underlying universe of any specific ALife model and should permit the study of higher-level observationally "emergent" phenomena. Initial attempt in this direction appeared in [2]. The central idea proposed there in is a high level formal axiomatic characterization of the observational processes, which leads to abstractions on the model universe, which are consequently used for establishing the necessary elements and the level of life-like behavior in the model. With a focus on the problem of entity recognition, in this work we extend that axiomatic framework and derive results on computational complexity theoretic analysis of the observational processes.

The paper is organized as follows: In Section 2, we will review the framework followed by computational complexity theoretic analysis in Section 3. Section 4 presents further discussion and pointers for future research.

2 The Framework

Preliminaries: Apart from usual set theoretic notations e.g., $\setminus$ (set difference), $\mathcal{P}$ (power set), $\leadsto$ (partial function), $\mapsto$ (total function), following notation from first order logic [3] will be used: $\wedge$ (and), $\neg$ (not), $\Rightarrow$ (implication), $\Leftrightarrow$ (iff), $\exists$ (existential quantifier), $\forall$ (universal quantifier), and $\models$ (satisfies). Also concepts from multiset theory [4] (e.g., $\uplus$ (multiset join)) and the theory of computational complexity [5,6] (e.g., 'big-Oh' notation - $\mathcal{O}$) would be used in the formal exposition of the derived results. $\mathcal{N}^+$ is the set of positive integers.

2.1 The Formal Structure of the Framework

In this section we will review and extend the axiomatic framework presented in [2]. We limit our attention to only those aspects which pertain to the problem of entity recognition.

Axiom 1 (The Axiom of Observable Life). *Life-like phenomena in a ALife model exists only if it can be observed using its simulations.* Ξ

Definition 1 (Observation Process). $Obj : \Gamma \mapsto \Pi$*: An observation process* Obj *is defined as a computable transformation from the underlying model structure* $\Gamma = (\Sigma_0, \Sigma, \delta)$ *to observer abstractions* $\Pi = (Abs_{ind}, Abs_{dep})$*, where* Abs_{ind} *is the set of process independent abstractions and* Abs_{dep} *is the set of process dependent abstractions.* Ξ

Definition 2 (States). Σ*: set of observed states of the model across simulations.* Ξ

Definition 3 (Observed Run). $\mathcal{T} : \Sigma \leadsto \mathcal{P}(\mathcal{N}^+)$*: An observed sequence of states ordered with respect to the temporal progression of the model during its simulation.* Ξ

$\mathcal{N}^+$ acts as a set of indexes for the states in the sequence. Since a state may appear multiple times in a simulation, subsets of $\mathcal{N}^+$ are used to denote that. Each such sequence represents one *observed run* of the model. We let $\Sigma_{\mathcal{T}}$ denote the set of states appearing in a specific run $\mathcal{T}$.

Σ and a set of observed runs $\mathcal{T}_1, \mathcal{T}_2, \ldots$ thus define the underlying dynamic structure of the model as a state machine $\Gamma = (\Sigma_0, \Sigma, \delta)$ with respect to a given observation process. Where

- $\Sigma_0 \subseteq \Sigma$ is the set of starting states, i.e., $\forall s \in \Sigma_0 . \exists \mathcal{T}_i$ *such that* $\mathcal{T}_i(s) = 0$.
- $\delta \subseteq \Sigma \times \Sigma$ is the transition relation between states, i.e., $(s, s') \in \delta \Leftrightarrow \exists \mathcal{T}_i$ *such that* $\mathcal{T}_i(s') = \mathcal{T}_i(s) + 1$.

2.2 Axioms of Entity Recognition

Observer Abstraction 1 (Entity Set). *E_s: Multiset of entities observed and uniquely identified by the observer in a state s of the model for a given run $\mathcal{T}$. $E_{\mathcal{T}} = \biguplus_{s \in \Sigma_{\mathcal{T}}} E_s$ is the multiset of entities observed and uniquely identified by the observer across the states in the given run $\mathcal{T}$.* Ξ

Defining a sound criterion to identify entities often requires a careful attention since arbitrariness in defining entities might well lead to the problem of false positives as discussed in [2]. "Tagging" can be used as a mechanism for the identification of individual entities whenever there exist multiple entities in the same state which are otherwise indistinguishable.

In general an observation process may recognize an entity in a state as a subset of the observable atomic structures, that is, (structure of) an entity can be defined as a subset of the state itself. Formally, $E_s \subseteq \mathcal{P}(s)$. Notice that such a subset based structural identification of entities also allows entities with overlapping sets of atomic elements. In such scenarios also application of additional tagging is essential. Formally, $Tagging : E_{\mathcal{T}} \mapsto I_{Tag}$, where I_{Tag} is the set of unique tags to be associated with the entities.

Also we have the following two axioms imposing consistency requirements on the entity identification. First axiom states that every entity in a state is uniquely identified. Second axiom states that the set of entities identified in identical states should be the same.

Axiom 2 (Axiom of Unique Identification of Entities). *An entity must be uniquely identified in a given observed run $\mathcal{T}$. Formally, Tagging is a one-to-one function on states, that is, $\forall s \in \Sigma_{\mathcal{T}}.\ \forall e, e' \in E_s.\ Tagging(e) = Tagging(e') \Rightarrow e = e'$.* Ξ

Axiom 3 (Axiom of Unique Identification in States). *If two states are identical, i.e., consist of the identical multisets of atomic observable structures, then an observer must identify the same multisets of entities in these states irrespective of their temporal ordering in the observed run $\mathcal{T}$. Formally, $\forall s, s' \in \Sigma_{\mathcal{T}} :\ s = s' \Rightarrow E_s = E_{s'}$.* Ξ

Note that converse is not true, $E_s = E_{s'} \not\Rightarrow s = s'$, i.e., an observer could also identify identical multisets of entities in different states. This axiom may also be considered as the *soundness axiom* for entity recognition.

Finally we have the following complementary *completeness axiom* for entity recognition.

Axiom 4 (Axiom of non-Ignorance). *It must not be true that an observer omits identification of an entity in a state s but in a different state s' identifies it as consisting of the same atomic elements which were also available in s.*

3 Computational Complexity of Entity Recognition

In this section, we will derive safe (though pessimistic) upper bounds on the worst case time complexity to the problem of entitiy recognition for arbitrary ALife models.

An important problem to be considered while providing estimates on the computational complexities is that observed state progression during simulations might not correspond to the actual underlying reaction semantics for a specific entity. In other words observed states during simulations progress according to the underlying updation rules for the model, which determine which subset of entities would react in any state. Since an automated discovery of the updation rules in not what is considered in this paper and automated discovery of entities is considered to be independent of the underlying updation semantics, we assume that all those entities, which are enabled to react in each state, are indeed allowed to react. In cases where it is not true, an observation process could be assumed to store state subsequences of certain finite lengths where all (or most of) the enabled entities have been observed to react and then merges all those states in a subsequence into a single meta state, which would reflect the effect that most of those entities which could react in the initial state have indeed reacted. However, which of the entities will react and the outcomes is still determined by the underlying updation rules and to be followed by observation process as well.

In general, an automated (algorithmic) discovery of entities with life-like behavior in arbitrary simulation models is a non trivial problem. Currently most of the ALife studies either depend upon the researchers to identify the entities of interest (e.g., using geometric features) or the models are built using predetermined entities (e.g., programs, λ expressions etc). Difficulty arises owing to the fact that in a state s of size n, total possible entity sets $E_s \subseteq \mathcal{P}(s)$ could be as large as 2^{2^n} in case where entities may have overlapping structures or in case where entities do not have overlapping structures it could be in the range $[\mathbf{P_n}, \mathbf{B_n}]$ [7], where $\mathbf{P_n}$ is the n^{th} *Partition number* denoting the number of partitions of an integer n and $\mathbf{B_n}$ is the n^{th} *Bell number* denoting the number of partitions of a set (not multiset) with n elements. Both these estimates become very large even for small values of n. Therefore, in absence of the knowledge of the underlying reaction semantics of the model, searching a specific entity set from these super exponentially many possibilities may require large amount of computational resources. Even before a search for such an entity set can be performed, algorithmically determining the criterion to select the entities based upon their characteristics is itself a difficult problem.

Therefore we consider the case of expert guided semi automated discovery of entity sets, which requires an ALife researcher to a priori define the criterion, which possible entities in the model may satisfy. Actual discovery of these entities which satisfy such criterion may then be performed algorithmically. For example, the "Observer CA" system designed to automatically extract living loops in the Sayama's model [1] is based upon the fact that the set of living loops of type n would consist of all the subsets of non-quiescent CA cells that contain an open square made of sheath '2' and signal '3' whose edges are n sites long. Interestingly, *the no free lunch theorems in search* [8,9] indicate that such expert assisted discovery could be algorithmically much more efficient than a search without any prior knowledge of the object-model.

We next present a result on time complexity requirement for such an expert assisted discovery of entities in a representative abstract model by reducing the problem of entity recognition into the problem of logical formula satisfiability checking.

3.1 Computational Complexity of Entity Recognition Using Syntactic Constraints

Let us assume that the distinct atomic structures in the model can be represented as the alphabets collected in $X = \{a_1, a_2, \ldots, a_l\}$. A state $s \in \Sigma_{\mathcal{T}}$ would be expressed as a multiset over X. Entities in s would thus be the multisubsets of s comprising of these alphabets. These entities can indeed be encoded as binary strings of size $n = s(a_1) + s(a_2) + \ldots + s(a_l)$, where first block of size $s(a_1)$ bits is for the alphabet a_1 and so on such that if an entity e is encoded as a binary string $b_1 \ldots b_{s(a_1)} \ldots b_{s(a_1)+s(a_2)} \ldots b_n$, then $b_i = 1$ when k^{th} copy of a_j is in e and $b_i = 0$ when it is not, where $i = k + \sum_{z=1}^{j-1} s(a_z)$.

Since the process of recognition of an entity can also be considered as a sequence of the choices made on the atomic structures as to whether they are in the entity or not, we can consider the expert given criterion guiding these choices as a boolean formula $\mathcal{F}$ on the variables $\{b_1, \ldots, b_n\}$. Thus an entity set E_s would consist of those entities which satisfy $\mathcal{F}$, i.e., if $e \in E_s$ then $e \models \mathcal{F}$. With the well known results in the theory of computational complexity we know that checking the satisfiability of a boolean formula is *NP-hard* [5], and currently there is no known algorithm which can exactly solve it always in polynomial number of time steps. Therefore we have the following result, which is proved using *the reduction method*, often used in proving such claims.

Theorem 1. *The problem of entity recognition using structural (syntactic) constraints is NP-hard.*

Proof. For the purpose of reduction, we consider the 3 SAT problem, which was among the first problems to be demonstrated in the class NP-Hard[1] [10]. For reduction any instance of 3-SAT problem would be reduced into an instance of an expert guided entity recognition the following way:

Consider a 3 SAT formula $\mathbf{f}$ in conjunctive normal form over the Boolean variables $\mathcal{X} = \{b_1, \ldots, b_n\}$ and their negations $\bar{\mathcal{X}} = \{\bar{b}_1, \ldots, \bar{b}_n\}$. For translation into an instance of expert guided entity recognition, the set of the variables $\mathcal{X}$ gives rise to a set of atomic structures, such that there exists exactly one unique atomic structure corresponding to each variable in $\mathcal{X}$. The formula $\mathbf{f}$ is considered as it is as a Boolean structural constraint - $\mathcal{F} \equiv \mathbf{f}$ - such that presence of a negated variable $\bar{b}_i$ in a clause is interpreted as forcing the absence of the atomic structure corresponding to the true variable b_i. Conjunction and disjunction also have natural interpretations of 'include both' and 'include either'. Hence each recognized entity represented as a subset of the atomic structures by the observation process following the constraint $\mathbf{f}$ would correspond to a solution of the original 3 SAT formula in the sense that if an atomic structure corresponding to the variable b_i is present in the recognized entity, then that variable needs to be set to 1 in the corresponding solution and if it is not present then it needs to be set to 0, e.g., if an recognized entity is $\{b_1, b_3, \ldots, b_{n-1}\}$, corresponding solution vector would be $1010 \ldots 10$. It is clear that $\mathbf{f}$ is satisfiable if and only if there exist at least one entity which can be recognized as satisfying the structural constraint $\mathcal{F}$.

This reduction proves that the problem of entity recognition using structural (syntactic) constraints is as hard as the 3 SAT problem, which is known to be a NP-hard problem. Therefore the problem of entity recognition using structural constraints is also NP-hard. Ξ

Let us also consider the case, where entities do not have overlapping structures. In that case, the entity set E_s would be a partition of a subset of s. However this constraint cannot be formalized only as a (Boolean) formula over the variables $\{b_1, \ldots, b_n\}$ alone and instead needs to be specified over the solution space of $\mathcal{F}$. Formally, it can be stated as a first order formula: $\forall e, e' \in E_s \Rightarrow [e, e' \models \mathcal{F}] \wedge [\forall i. \neg(e[i] \wedge e'[i])]$, where $e[i]$ is the i^{th} bit of the bit string representing entity e and the clause $[\forall i. \neg(e[i] \wedge e'[i])]$ specifies that no two entities e, e' in E_s can share any of the atomic structures.

Since in order to satisfy this constraint, it would require as a prior step to identify the entities as per the given constraint $\mathcal{F}$, computational complexity of generating the entity set would be no less than the what is specified in the theorem 1 for the case of overlapping entities. In the worst case, when an initial entity set E_s might consist of all the 2^n subsets of s (without the constraint of non-overlap), checking the clause $[\forall i. \neg(e[i] \wedge e'[i])]$ may require searching though all these entities (as strings) to see if the i^{th} bit is not set to 1 in more than one string and whenever this is so, arbitrarily selecting only one of these entities in the final set. This gives an upper bound of $\mathcal{O}(2^n) + \mathcal{O}(n2^n) = \mathcal{O}(n2^n)$.

[1] Actually NP-Complete.

However, in practice, we may not define such boolean constraint $\mathcal{F}$ as a criterion for entity recognition and instead may use more sophisticated and well expressed constraints using higher level programming languages. This leads to a much larger question as to what extent checking the presence of entities which may satisfy these criterion is feasible and how much computational resources are required for answering that. The corresponding formula satisfiability problems have been well studied for long in the fields of computability theory and mathematical logic [11]. For example, currently known techniques for syntactic pattern recognition [12, chapter 1] use regular, context free, context sensitive, tree, graph, shape, and matrix grammars to specify the structural constraints for the objects to be recognized.

3.2 Computational Complexity of Entity Recognition Using Semantic and Statistical Constraints

A natural question which may occur to the reader at this point is that structural constraints are possibly not always the right criterion to select the set of entities in ALife models. Often characteristics of the life-like behavior are expected to be exhibited on *semantic* level by entities, e.g., reproduction, autopoisis etc. However algorithmically deciding these semantic constraints would require much more computational resources then the relatively simpler syntactic criterion analyzed above. The reason for this is that the identification of entities which satisfy certain semantic constraints inevitably requires larger search space of entities to explore. Consider for example, "reproduction" as a criterion for selecting the entities in a state. Since the very determination of the fact that an entity e in a state would reproduce would require observations in the future states, the observation process needs to assume all possible entities in the current state as potentially reproducing entities and hence would require at least $O(2^n)$ steps as estimated above. Furthermore even if an entity is not reproducing in a specific state, it does not preclude the possibility that it would reproduce in some future state. This increases the resource requirements for the observation process much further and precise estimates are usually controlled by the actual semantic criteria at hand.

A third kind of criterion which may also be used sometimes to select an entity set is the *statistical* criterion. Statistical constraints over the possible entities in a single state is relatively less resource centric and would possibly only require linear increment over the syntactic constraints. However statistical constraints over a sequence of states may be as challenging as the semantic criterion discussed above. Indeed, in some sense statistical constraints could also be considered as semantic criterion but over a population of (potentially) identifiable entities. For example a criterion specifying the conservation of certain statistical properties of population of entities would demand adequate resources to consider many possible sets of entities over a sufficiently long sequence of states. Heredity and natural selection are important examples of statistical criterion to establish life-like evolutionary behavior [2].

4 Conclusion and Further Work

In this work, we defined an inference process specifying necessary conditions, as axioms, which must be satisfied by the outcomes of observations made upon the model universe in order to infer the presence of entities satisfying the prespecified constraints. Computational complexity theoretic analysis of the expert guided entity recognition reveals that an automated discovery of life-like entities could be computationally intensive in practice and techniques from the fields of pattern recognition and machine learning in general can be of significant use for such purposes.

The work can can be further extended in several interesting directions, including the following: As an immediate next step the framework could be extended to include an algorithmic discovery of the emergent evolutionary behavior by the population of observed entities and its associated computational complexity theoretic analysis. Associated computational complexity analysis can be further refined and strengthened by considering specific classes of models for which most of the parameters have precise bounds compared to the generic analysis presented in this paper.

References

1. Sayama, H.: Constructing Evolutionary Systems on a Simple Deterministic Cellular Automata Space. PhD thesis, Department of Information Science, Graduate School of Science, University of Tokyo (December 1998)
2. Henz, M., Misra, J.: Towards a framework for observing artificial life forms. In: Proceedings of the 2007 IEEE Symposium on Artificial Life (IEEE-ALife 2007), pp. 23–30. IEEE Computational Intelligence Society, Los Alamitos (2007)
3. Mendelson, E.: Introduction to mathematical logic, 4th edn. Chapman and Hall, Boca Raton (2001)
4. Syropoulos, A.: Mathematics of multisets. In: Multiset Processing, pp. 347–358. Springer, Heidelberg (2001)
5. Papadimitriou, C.: Computational Complexity. Addison-Wesley, Reading (1994)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)
7. Yorgey, B.: Generating multiset partitions. The Monad Reader (8), 5–20 (2007)
8. Schaffer, C.: A conservation law for generalization performance. In: Proceedings of the International Conference on Machine Learning, pp. 265–295. Morgan Kaufmann, San Francisco (1994)
9. Wolpert, D., Macready, W.: No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute (1995)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability. In: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)
11. Boolos, G.S., Burgess, J.P., Jeffrey, R.C.: Computability and Logic, 5th edn. Cambridge University Press, Cambridge (2007)
12. Huang, K.Y.: Syntactic Pattern Recognition for Seismic Oil Exploration. World Scientific Publishing, Singapore (2002)

Creative Agency: A Clearer Goal for Artificial Life in the Arts

Oliver Bown and Jon McCormack

Centre for Electronic Media Art, Monash University,
Clayton 3800, Australia
{oliver.bown,jon.mccormack}@infotech.monash.edu.au

Abstract. One of the goals of artificial life in the arts is to develop systems that exhibit creativity. We argue that creativity *per se* is a confusing goal for artificial life systems because of the complexity of the relationship between the system, its designers and users, and the creative domain. We analyse this confusion in terms of factors affecting individual human motivation in the arts, and the methods used to measure the success of artificial creative systems. We argue that an attempt to understand *creative agency* as a common thread in nature, human culture, human individuals and computational systems is a necessary step towards a better understanding of computational creativity. We define creative agency with respect to existing theories of creativity and consider human creative agency in terms of human social behaviour. We then propose how creative agency can be used to analyse the creativity of computational systems in artistic domains.

1 Introduction

Both artificial intelligence (AI) and artificial life (Alife) have been used to study artistic creativity and to create new forms of art. Traditionally, AI has focused on the artificial simulation of human intellectual capacities, whereas Alife takes its inspiration from the creative power of nature through processes such as self-organisation, natural selection and autonomy. The study of Alife therefore holds special significance for the arts due to its inherent concern with creativity beyond human agency, paying special attention to systems that exhibit the emergence of new, higher-level primitives in a system *de novo* [1]. Despite these differences of focus, in both approaches artificial creativity is a commonly stated goal, whether represented as a means for better understanding human creativity, creativity in general, or towards new systems for artists. But although the intent is clear, a perspicuous definition of this goal or means of objective measurement remains conspicuously hazy. As Saari and Saari put it, “Creativity is fascinating! We know so much about the topic without having the slightest idea what it is” [2, p. 79].

Our motivation is a lack of focus on *agency* in the literature on creativity. We argue that a better understanding of creative agency will help clarify the goals of achieving creative behaviour in computational systems.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 254–261, 2011.

2 Defining Creative Agency

A typical definition of creativity (e.g. [3]) is as follows:

Definition 1. *A system is creative if it produces novel and valuable (appropriate, useful) output.*

Understandably, the novelty and value of the output of a system have been predominant areas of interest in the literature on creativity. In this paper we turn to the process of production itself: the relationship between subject (the system) and object (the output). We address this relationship in terms of what we call *creative agency*: the extent to which the subject is responsible for *producing* the object.

Definition 2. *The creative agency of a system is the degree to which it is responsible for a creative output.*

Identifying creative agency therefore involves the (apparently subjective) evaluation of responsibility. It is not the output itself that we are interested in, but the creativity invested in the output, in other words, the intangible qualities of novelty and value. Thus a master artist could employ skilled students to create a work, not once touch the work, but still be attributed with the creative agency associated with the production. By the same reasoning, a wealthy patron commissioning such a work could take some credit for making the work come about, but their choice to employ a reputed artist would be to borrow already existing creativity.

In computational creativity, the problem of creative agency is often taken as being of secondary importance to the novelty and value of the output produced by a system. A lack of attention to the nature of creative agency is common when discussing creativity in humans, because it is generally taken as given that humans are the only kind of creative agent we need consider. In the case of computational creativity, however, this can be a source of opacity, since we cannot directly translate the notion of creative production that applies to humans straight onto computational systems. Computational systems have a completely different relationship to their environments from people. Not least, they are invariably brought into the world by human design. By highlighting this relationship, computational creativity throws into light the problem of creative agency not only in computational systems, but also in human and natural systems.

With respect to creative agency, systems that exhibit a low degree of creative agency make a smaller genuine contribution to the novelty and value of the output they are involved in producing; in such cases the creative agency should instead be attributed to the designer of the system. A system that has a high degree of creative agency, on the other hand, should have a greater claim to the novelty or value identified in any output produced by that system. If the output is indeed novel and valued (to be determined separately) then by virtue of its greater contribution to that output, the system itself can be deemed creative.

In short: novelty and value that cannot be attributed in some measure to the computational system should have no weight in supporting claims about the creativity of that system.

We can think of the assignment of creative agency to computational systems as akin to assigning royalties to a collection of artists who collaborated on a creative work – the greater their original contribution to the the output, the higher the attribution of creative agency.

A simplified representation of the problem of creative agency is shown in Figure 1. Agency and creativity are placed on distinct axes (without necessarily implying that they are independent), and we consider two hypothetical computational systems. System A has a high degree of agency but does not produce particularly novel or valuable output, whereas System B's output is highly novel and valuable even though the system itself is not particularly responsible for the creativity of that output. The diagonal line represents a hypothesised limit of current systems. At present, designers of computationally creative systems are forced to find compromises between systems of type A and systems of type B, but one of the ultimate goals of computational creativity is to find systems that exhibit the agency of System A, but with true creative output as in System B.

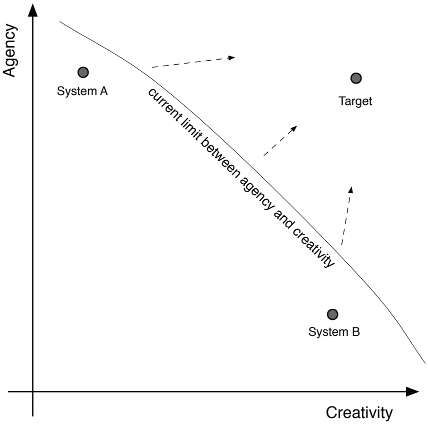


Fig. 1. Graph representing a hypothesised limit to the combination of agency and creativity in current computationally creative systems

3 A Layered View of Traditional Dimensions of Creativity

The human species is eminently capable of introducing novel structures into the world, and the same is patently true of nature. Since we would not expect to identify the creative agency of nature in the random mutations of genetic variation, it would be reasonable to suspect that the creative agency of culture as a whole is greater than the sum of the creative agencies of individual humans. A number of contemporary trends in human evolutionary theory have expounded this argument. Meme theory, for example, proposes that cultural behaviour can be explained in terms equivalent to genetic theory, by positing the meme as an abstract cultural *replicator* [4].

This points to the need for a multi-layered model of creativity that unifies individual creativity with super-individual cultural processes responsible for driving the emergence of creative domains themselves. The need for distinguishing creativity on different levels is also driven at the sub-individual level by the perspective in cognitive science typified by Andy Clark's *Extended Mind* hypothesis, which postulates that cultural artefacts offer cognitive cybernetic enhancement [5]. According to this point of view human creative agency is already a highly distributed network of elements with human brains at the centre (for the time being). This distributed model complements the point of view that the brain itself is a distributed set of functional units, as typified by the *Swiss Army-knife* model of mental modularity, proposed by Barkow, Cosmides and Tooby [6].

For the creativity theorist Csikszentmihalyi, the problem of defining value in the attribution of creativity necessitates a view centred on the embeddedness of individuals within creative domains:

> There is no way to know whether a thought is new except with reference to some standards, and there is no way to tell whether it is valuable until it passes social evaluation. Therefore, creativity does not happen inside people's heads, but in the interaction between a person's thoughts and a sociocultural context. It is a systemic rather than an individual phenomenon. [7, p.23]

Csikszentmihalyi therefore defines a creative person as "someone whose thoughts or actions change a domain, or establish a new domain" (p28). Since modifying a domain influences the way that domain will respond to future potential creativity, individual and domain are strongly interdependent. Csikszentmihalyi's characterisation of the relationship between individual and domain extends naturally to a general relationship between creative agency at different levels, that has an unmistakably Darwinian, or perhaps more appropriately ecosystemic, feel. The creativity of certain individuals is determined by processes occurring at a higher level (the creative domain), mediated by the generation of a system of value. This is Darwinian in that a higher level process selectively filters elements being produced and reproduced at lower levels. The system of value, like the Darwinian concept of fitness, is implicit and mostly revealed in hindsight.

Novelty itself must also be seen as domain specific if it is to have any non-trivial meaning. Trivially, everything that is different is novel. It is less trivial, and far more meaningful, to measure the *degree of novelty* of things. But measurement occurs in a metric space, and metric spaces are not real things, but are constructed by perceiving agents. This is not a problem for creativity *per se*: novelty is our evidence for creativity, but creative systems don't need to recognise novelty to be creative. This suggests that novelty-seeking alone may have little functional utility. Some human cultures, such as Western industrialised society, seem to have fostered neophilia, forging an inherent link between novelty and value. It is not self-evident that this has any functional utility, however [8].

Boden discusses the cognitive requirements for humans to find new ways to achieve goals, distinguishing between three kinds of creative process: combinatorial creativity is the combination of existing elements to create new elements; exploratory creativity is search through an existing conceptual space; and transformational creativity is the transformation of an existing conceptual space. A problem for the precise application of Boden's theoretical work has been the formulation of what these conceptual spaces actually are [9], particularly with respect to understanding how transformational creativity differs from exploratory creativity [9,10]. Viewing creativity at multiple levels allows us to hypothesise that transformational creativity is really a process occurring at the higher cultural level, for example in the way described by Kuhn in his theory of scientific revolutions [11], and that value (which itself can be emergent) is the means by which the products of lower level creative acts are shunted up to higher levels.

4 Categorising Agency in Computational Creativity

The multilevel approach to creativity helps to identify three distinct ways in which computational systems can exhibit creative agency. The first is by actively contributing to, and enhancing, the creative agency of individual humans, as an active component in a distributed creative process. Most computationally creative systems to date fit this category, although they may be at odds with their designers' original goals of establishing human-like creativity. This adheres to the extended mind perspective that our individual creativity is already highly distributed and enhanced by cultural artefacts, some of which may be computational systems performing complex tasks. We already use computers creatively, but their role in our individual creativity is creeping towards an increasingly active status. Programs like Cohen's *Aaron* [12], and the general increase in popularity of generative art demonstrate how this shift is taking place.

Disappointingly, the predominant tool of Alife-based art – the interactive genetic algorithm (IGA) [13] – has had limited success as a tool for enhancing creativity. The IGA aims at fixing the problem of formally defining complex human aesthetic preferences by letting humans take the place of the fitness function, but this arguably leads to a poor creative partnership where both user and algorithm assume roles of little creative agency. The genetic algorithm is passive in that it relies on the user for the crucial step of selection, but the user is rendered passive by being unable to control the long-term course of evolution or the underlying structure of the developmental process. Nevertheless, interactive genetic algorithms are beginning to emerge in commercially available creative software where their use makes sense. Dahlstedt's *Mutasynth*, for example, assists a user to search a vast space of possible synthesiser sounds using an IGA with visual representation of synthesis parameter space [14]. Anyone who has played with a synthesiser will be familiar with the mild sensation of blind search already inherent the mapping from parameters to sounds.

The second approach is to consider how computational systems can fit into existing processes at the higher cultural level. These systems need to identify how individual interactions lead to social structures and cause cultural change. Pockets of research have been conducted in this area, spanning a variety of disciplines. The *DrawBots* project [15] attempted to accentuate the social construction of a robotic art system's creative agency as far as possible by allowing its creations to be exhibited in an art gallery without human intervention, illustrating the potentially vast variety of ways creative agents might manipulate creative domains. This includes the potential circularity that perhaps the legitimisation of the art gallery is enough to *make* the work acceptable to a receptive audience. That said, if, in hindsight, the robot did have an impact on its creative domain in this way, the sticky problem is that the agency of this particular act (putting the work into the gallery) falls yet again to the human agent that curated the event.

Romero, Machado and Santos' ongoing *Hybrid Society* project aims to build a virtual social system coinhabited by human and computer artists, all operating as both producers and critics and interacting in social networks such that the real artistic value systems of the humans influence the world of the artificial agents [16]. In principle, in such an environment (as with DrawBots), agents may potentially influence the creative domain of human participants. By Csikszentmihalyi's definition, nothing could provide a better indication of creative agency than this.

Earlier Alife style models, based completely *in silico* (e.g. [17,18,19]) have already established the potential of exploring basic cultural or bio-cultural dynamics using multi-agent systems, yet it is hard to ground those dynamics in a way that produces anything we would recognise as creative (novel, yes, but of any aesthetic interest, no!). This overlaps smoothly with our third suggested approach, which is to work out how to exploit the creative potential already under investigation in *in silico* research in Alife, but in artistic domains. A pioneering example of such research is the Italian composer Agostino di Scipio's musical performances, which work by building *sonic ecosystems* that transform the latent sound of the performance space into musical works using a series of complex variations on the process of audio feedback [20]. Di Scipio's insight is to begin with the medium that he is interested in, and construct complex networks of processes within that domain (sound itself). In other artistic domains, elements from Alife can be used more literally, such as Jon McCormack's installation, *Eden*, which presents a population of artificial learning agents whose environment is 'fed' by the presence of audience members, who are lured to stay in the installation space by the agent's ability to create interesting music [21]. *Eden* creates an evolving symbiotic relation between the audience and artificial agents. In these domains, creative emergence can occur that is inherent to the environment defined by the work, and as such, the works do achieve an internal creative agency, without conflicting with the creative achievements of the artists involved in making them. However, this internal creative agency only becomes of interest when it is sufficiently coupled to the creative domain in which it exists.

5 Conclusion

In this paper, we have argued, for purely practical reasons of evaluation, the need to consider the creative agency of systems that are involved in producing a creative output. Although we believe that this focus will help to clarify the goals of computational creativity and the potential role of Alife in this domain, our contribution does not take the form of a mathematical definition of creative agency which could be easily applied by researchers to various creative systems. Instead, it appears necessary that assigning creative agency will continue to be a subjective matter based on disparate evidence. Our goal has been to attempt to form an appropriate perspective with which to simultaneously view creative processes in nature, human culture, individual human behaviour and existing computationally creative systems. We have argued for a perspective that recognises creative agency and the role of value in mediating between levels in a hierarchy of creative processes. This replaces the dominance of the human individual as the exemplary creative agent with a more distributed set of interacting elements into which computational systems can more easily situate themselves. We propose that this clarifies the potential creative role of Alife systems in the cultural domain of the arts. Such a perspective can ultimately lend itself to more detailed numerical analysis of creativity, however, further discussion combining sociological, philosophical and Alife-based reasoning will be needed before this can be achieved.

Acknowledgements

This research was funded by the Australian Research Council under Discovery Project grant DP0877320.

References

1. Cariani, P.: Emergence and Artificial Life. In: Artificial Life II, SFI Studies in the Sciences of Complexity, vol. 10, pp. 775–797. Addison-Wesley, Redwood City (1991)
2. Saari, D.G., Saari, A.L.: Toward a mathematical modelling of creativity. In: Andersson, A.E., Sahlin, N.-E. (eds.) The Complexity of Creativity, pp. 79–103. Kluwer Academic Publishers, Netherlands (1997)
3. Boden, M.: The Creative Mind. George Weidenfeld and Nicholson Ltd. (1990)
4. Blackmore, S.J.: The Meme Machine. OUP, New York (1999)
5. Clark, A.: Natural-Born Cyborgs: Minds, Technologies, and the Future of Human Intelligence. Oxford University Press, Oxford (2003)
6. Barkow, J.H., Cosmides, L., Tooby, J.: The Adapted Mind: Evolutionary Psychology and the Generation of Culture. OUP, New York (1992)
7. Csikszentmihalyi, M.: Creativity: Flow and the Psychology of Discovery and Invention. Harper Collins, New York (1996)
8. Gray, J.: Straw dogs: thoughts on humans and other animals, vol. 246. Granta Books, London (2002)

9. Wiggins, G.A.: Towards a more precise characterisation of creativity in AI. In: Weber, R., von Wangenheim, C.G. (eds.) Case-Based Reasoning: Papers from the Workshop Programme at ICCBR 2001, pp. 113–120. Naval Research Laboratory, Navy Centre for Applied Research in Artificial Intelligence, Washington, DC (2001)
10. Thornton, C.: How thinking inside the box can become thinking outside the box. In: Cardoso, A., Wiggins, G.A. (eds.) Proceedings of the 4th International Joint Workshop on Computational Creativity, pp. 113–119. University of London, Goldsmiths (2007)
11. Kuhn, T.S.: The Structure of Scientific Revolutions, 3rd edn. University of Chicago Press, Chicago (1996)
12. McCorduck, P.: AARON's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen. W. H. Freeman and Co., New York (1990)
13. Takagi, H.: Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. Proceedings of the IEEE 89, 1275–1296 (2001)
14. Dahlstedt, P.: A mutasynth in parameter space: interactive composition through evolution. Organised Sound 6(2), 121–124 (2006)
15. Bird, J., Stokes, D.: Evolving minimally creative robots. In: Colton, S., Pease, A. (eds.) Proceedings of The Third Joint Workshop on Computational Creativity (ECAI 2006), pp. 1–5 (2006)
16. Romero, J., Machado, P., Santos, A.: On the socialization of evolutionary art. In: Giacobini, M., Brabazon, A., Cagnoni, S., Caro, G.A.D., Ekárt, A., Esparcia-Alcázar, A., Farooq, M., Fink, A., Machado, P., McCormack, J., O'Neill, M., Neri, F., Preuss, M., Rothlauf, F., Tarantino, E., Yang, S. (eds.) EvoWorkshops 2009. LNCS, vol. 5484, pp. 557–566. Springer, Heidelberg (2009)
17. Saunders, R., Gero, J.S.: Artificial creativity: Emergent notions of creativity in artificial societies of curious agents. In: Proceedings of Second Iteration (2001)
18. Miranda, E.R., Kirby, S., Todd, P.M.: On computational models of the evolution of music: From the origins of musical taste to the emergence of grammars. Contemporary Music Review 22(3), 91–111 (2003)
19. Werner, G., Todd, P.M.: Too many love songs: sexual selection and the evolution of communication. In: Husbands, P., Harvey, I. (eds.) Proceedings of the Fourth European Conference on Artificial Life, pp. 434–443. MIT Press/Bradford Books, Cambridge, MA (1997)
20. di Scipio, A.: Sound is the interface: from interactive to ecosystemic signal processing. Organised Sound 8(3), 269–277 (2003)
21. McCormack, J.: Eden: An evolutionary sonic ecosystem. In: Kelemen, J., Sosík, P. (eds.) ECAL 2001. LNCS (LNAI), vol. 2159, pp. 133–142. Springer, Heidelberg (2001)

Agent-Based Toy Modeling for Comparing Distributive and Competitive Free Market

Hugues Bersini

IRIDIA/CODE CP 194/6
Université Libre de Bruxelles
1050 Bruxelles
bersini@ulb.ac.be

Abstract. Is a competitive free market the most efficient way to equally allocate rare resources among economical agents ? Many economists tend to think it is the case. This paper presents a preliminary attempt through a very Alife like model to tackle this question. Agents which are alternatively producer, seller, buyer and consumer participate in a free market to increase their welfare. The simulation is organized and presented in a UML class diagram and two types of economy, competitive and distributive, are compared.

Keywords: Artificial economy, agent-based model, OO software, competitive vs distributive.

1 Introduction

The dramatic economical crisis occurring these days raises an intense and controversial discussion about the possibility for the market economy to self-regulate (stabilizing the prices) while, in the same time, ensuring the most equalitarian distribution of wealth among the agents involved in this market. Disciples of Adam Smith's doctrine of the invisible hand and famous historical advocates of competitive free market, such as Friedrich Von Hayek, Milton Friedman and so many others, have always seen in this type of decentralized and self-organized economical exchanges between agents the most efficient way to equally allocate rare resources among them. Many mechanisms such as "the parallel interactions among simple agents (since just motivated by profit)", "the reactivity of these agents through the stigmergic effect of price - buying less when prices increases and pushing price to decrease by selling more", "the self-organized stabilization of prices which equilibrate supplies and demands", testify of the clear connections existing between the sort of computational modeling popular in Alife and the working of free markets. As a matter of fact, John Holland was among the authors of a 15 years old paper entitled "Artificial Economic Life: A Simple Model for a Stockmarket" [1] where they state in the introduction: "*This stockmarket model may also be seen as a case-study in artificial life; from a random soup of simple rules an intelligent system spontaneously arises…*".

Many agent-based models [1-5] have been developed and run to illustrate or deny the capacity of free markets, in which seller and buyer agents compete to sell and buy their goods, to spontaneously converge to stable prices which balance the demand

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 262–269, 2011.

against the supplies rather than having the prices largely fluctuating. The purpose of the simplistic model presented in this paper is different. It rather questions the capacity of free-market to equally distribute the wealth among the agents as a function of the nature of the economical interaction: distributive vs competitive. So the emphasis is rather on the "competitive" nature of the economy instead of its capacity to stabilize price. How a competitive system which is supposed to install inequality among the agents by promoting the winners could by chance equally distribute the wealth ? While free markets per se do not obligatory require the existence of competition among agents, this competition seems inherent to the nature of the law of supply and demand which rules the dynamics of these markets. As a result of this competition among agents, the price of a rare product to acquire should increase, limiting the buyers, and the price of an abundant product to sell decrease, favoring performant sellers.

The next chapter will describe the main classes of the object oriented free market simulation presented in this paper. The following one will differentiate the two styles of economy to be compared: competitive (where the product to buy goes to the most offering agent) vs distributive (where the selection of the buyers to satisfy is done in a random way). The final one will present some preliminary results where the wealth among the agents, after many simulation steps of economical exchanges, will be compared for the two types of economy and as a function of the quantity and diversity of available products i.e. the strength of the potential competition. As usual with Alife models, the simulation presented here is not intended to depict any precise reality but has to be construed as a software thought experiment, the conception and the execution of virtual worlds helping to understand in outlines the behavior of a caricatured reality.

2 The Object-Oriented Agent-Based Toy Model

The model can't be better explained than by decorticating the UML class diagram represented in figure 1. Eleven classes need to be sketched. The class "*Resource*" is described by its quantity and its scarcity. These same resources will also serve to characterize the tastes and skills of the economical agents. Any product to be exchanged is composed of a certain quantity of some of these resources. The more the resources the more the different products that can be built out of them. The initial quantity of a resource is inversely related to its scarcity. The class "*Product*" describes a product indeed as a random composition of a random quantity of some of these resources. It is thus possible to compute how well a product satisfies a consumer by computing its welfare as a mapping between the nature of the agent's tastes and the composition of the product. Additionally, the basic fabrication price of a product can be deduced as a function of its composition.

The main actor of the program is depicted by the class "*Agent*", characterized by its welfare, its tastes, its skills (both expressed as a random vector of resources), and the quantity of money at its disposal. At each agent are associated four classes of behavior: "*Consumer*", "*Producer*", "*Buyer*" and "*Seller*". The consumer possesses a certain number of products each with its respective quantity. At each consumption, the associated agent's welfare increases as a result of the mapping between its tastes and the product being consumed. At every time step, this welfare decreases by one unity. The producer constructs a certain quantity of products. The product to be constructed can be determined either randomly (but with a low frequency) and more often based on the knowledge of the market in a way to be described later. At any construction of new products, the resources which compose these products decrease in quantity as well as the money possessed by the agent spent for the production. The seller is just able to make a selling offer composed of the certain amount of the products in the possession of its associated producer. The price of the offer is based on the fabrication price plus a benefit which depends on the type of economy being practiced: either random in a distributive economy or inversely related to the money the agent possesses in a competitive economy. The richest competitive agent can sell this product at the lowest price.

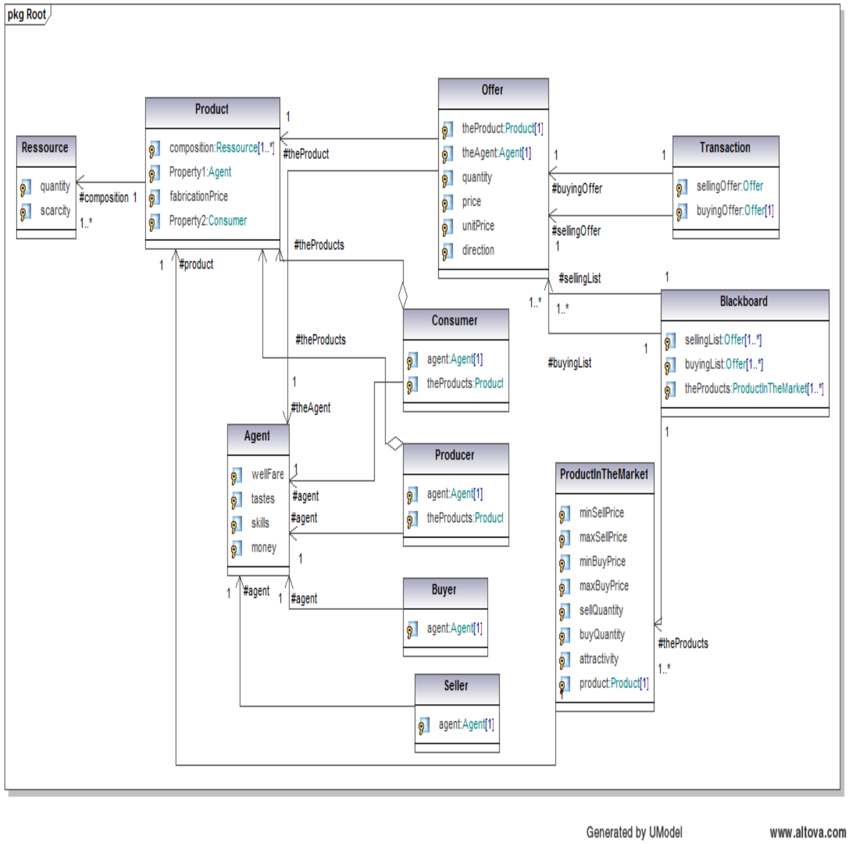


Fig. 1. The UML class diagram of the model with its eleven classes to be described in the text

The buyer also makes offer, but buying offers this time. The nature of the offer to be made depends on the current state of the market i.e. the current list of the selling offers (maintained in the blackboard as explained later). The buyer will first select in this list a selling offer the product of which corresponds the most to its tastes. Then, if rich enough, its offer will consist in the price of the selling offer plus a bid depending again on the nature of the practiced economy: random or proportional to its current money situation. In a competitive economy, the richest agent can make the highest offer for the same product.

An instance of the class "*Offer*" is simply characterized by the product in question, the quantity proposed, the agent that makes that proposal and the proposed price (the price to sell in the case of a selling offer or to buy in the other case). The direction of the offer indicates if it is of a buying or selling type. The "*Blackboard*" class inspired by the work described in [4] just gives birth to one object which is composed of both the lists of the buying offers and of the selling offers (these two lists are limited in size in order to give priority to the most recent offers). The blackboard also maintains a list of the products currently exchanged in the market and designated as members of the class "*ProductInTheMarket*". Each of its objects is associated with one given product, the maximum and minimum buying price that has been so far proposed for it as well as the maximum and minimum price it has been sold. It also maintains as attribute a first quantity obtained by summing all the buying offers that concerns it and a second one obtained by summing all the selling offers that equally concerns it. Based on these attributes, it is possible to compute the *attractivity* of a "*ProductInTheMarket*" as:

*attractivity = (buyQuantity*maxBuyPrice - sellQuantity*minSellPrice)*

Whenever a producer has to produce a new product, it will give its preference to the most attractive ones. Obviously, the fabrication price of this product will be related to the agent's skills (cheaper in the case of numerous skills). In such a way (and like argued by any advocate of the economical "laissez-faire"), the market consists in the best and most reliable source of information, for instance for the producer to be informed on the most wanted products. The existence of this blackboard, impacted by the agents actions and exerting in turn an effect on the agents behavior, is what connects the most this model with the more biological Alife ones. Finally, the "*Transaction*" class is composed of two offers, the selling and the buying one. A transaction will be created out of the blackboard in a way to be described in the next chapter. If possible (if the buyer is still rich enough and the seller still in possession of the products to sell), the execution of any transaction will result in a money exchange between the buyer and the seller, in the buyer agent acquisition of the products and the corresponding seller agent loss of these same products. It will also provoke a decrease in both the selling and the buying quantity of the corresponding productInTheMarket.

3 Distributive vs. Competitive Economy

The figure 2 illustrating the constitution of the blackboard and the way a transaction can be created out of it should help to distinguish between the two types of economy.

The only possible transaction to create in a competitive economy will be the one that maximizes the profit made both by the buyer and the seller agents. Here it will be the one which, for a given product, maximizes the difference between the buying unit price and the selling unit price. So in the case of the two lists represented in the figure, the only possible transaction to execute will be:

Transaction: Selling Offer 2 and Buying Offer 2.

In a distributive economy, any transaction for which the buying price is superior to the selling price will go, so for instance a possible transaction selected randomly among many others could be:

Transaction: Selling Offer 3 and Buying Offer 4.

Selling Offer	Agent	Product	Quantity	Price	UnitPrice
1	A	P1	100	200	2
2	B	P2	50	100	2
3	C	P3	50	50	1
4	A	P1	25	100	4
5	B	P2	50	100	2

Buying Offer	Agent	Product	Quantity	Price	UnitPrice
1	A	P1	50	150	3
2	D	P2	100	600	6
3	B	P1	25	100	4
4	C	P3	25	75	3
5	B	P3	100	300	3

Fig. 2. The two lists of offers which compose the blackboard, above the selling list below the buying list

Once a transaction selected, its execution will first see the two agents negotiating the unit price, here simply by cutting it in two. So, in the case of the transaction resulting from a competitive economy, the buying agent D will pay 200 for the 50 units of the product P2. In the distributive case, the buying agent C will pay 50 for the 25 units of the product P3.

The competitive economy forces the agent to more aggressively sell and buy, as a function of their current richness since, acting this way, the chance of their offer to be selected is increased. A distributive economy, selecting the offers at random, allows the agents to just care for their benefits while selling, and to simply identify the most tasty products to acquire while buying.

4 Preliminary Results

The simulation goes as follows. First 100 agents are created, each with the same amount of initial money and initial wellfare. The skills and tastes of each agent are determined at random as a subset of the resources to consume. Then to initiate the simulation, 50 agents selected randomly activate their “producer” and “seller” behaviors, selling the results of their productions. Finally, for a given amount of simulation steps, the agents (installed in an array of size 100 called “theAgents”) execute the following succession of instructions:

```
foreach (Agent a in theAgents) { a.decreaseWellFare(1);}
ra = ran.Next(0, theAgents.Length);
theAgents[ra].actProduce();
ra = ran.Next(0, theAgents.Length);
theAgents[ra].actSell();
ra = ran.Next(0, theAgents.Length);
theAgents[ra].actBuy();
foreach (Agent a in theAgents) { a.actConsume(); }
blackboard.operateCompetitiveTransaction();
// OR blackboard.operateRandomTransaction()
// depending on the type of economy
```

So successively, at each time step, an agent taken at random produces, then another one sells, another one buys and they all together consume. The selection of the transaction to operate is either done competitively or randomly. Two sets of simulation results will be now presented, comparing a competitive and a distributive economy for the exact same list of agents, first in the presence of only one resource (making the market more competitive) then for three resources.

Figure 3 shows the result for the one-resource case.

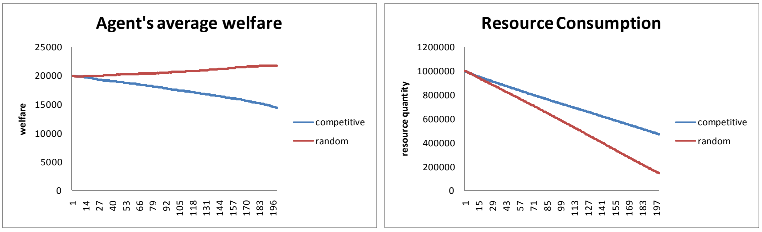


Fig. 3. Comparative results for the one-resource case

The average welfare of the 100 agents turns out to be better in the random case, due to a more intense and distributive consumption of the resource. The figure illustrating the single resource consumption (the y axis is the remaining quantity) shows how the random economy entails a stronger consumption to give greater satisfaction to the agents.

Figure 4 shows the results in the three-resources case.

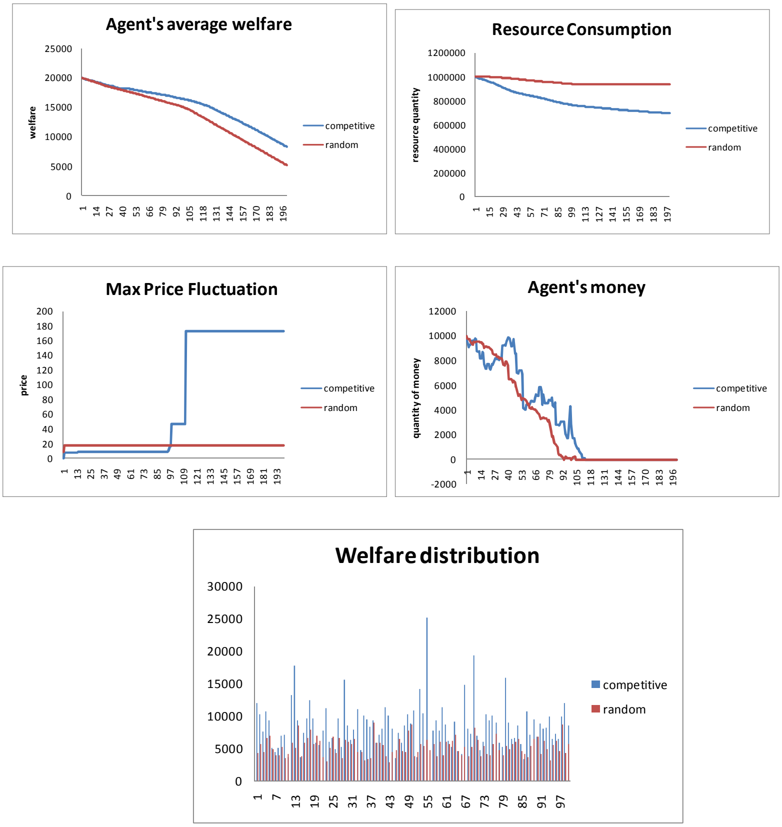


Fig. 4. Comparative results for the three-resources case

The situation turns out to be different with the competitive agents presenting a better welfare in average (associated with a greater resource consumption) although, as can be seen in the last figure plotting the distribution of this welfare, the variance is much larger in the competitive case. This result tends to confirm what is well known in economy. In a situation with enough resources to share, a competitive and more stimulating economy drives in average to a better welfare, but at the expense also of a greater inequality. Two complementary results tend to explain this increase of welfare in the competitive case: a more important consumption of resources but, above all, a much larger increase in the price of the products (due to the competitive incentive, remember that richer agents will pay more and sell for cheap) and thus a much larger fluctuation in the quantity of money in the possession of the agents during these products exchange. Agents tend to alternate more between the richer and poorer situation and this better circulation of money allows more agents to consume more resources. However, again not so surprisingly, competition leads to a larger inequality than in the distributive case.

5 Conclusions

This paper presents a very simple economical model which, although deliberately naïve (but not more naïve than the autopoietic cellular automaton or the game of life for illustrating biological processes), allows the appearance of interesting although not so unexpected collective outcome. The OO global architecture of the model should be robust enough to allow a lot of refinements and economical realism in the classes describing the agent's different behaviors (the producer or the buyer for instance). However, it remains interesting to discover the incentive effect for the collectivity of a competitive economy, once the resources are in sufficient quantity (though this does not resolve the problem of inequality among the agents), and the benefit of a more distributive economy (alleviating from the competitive pressure) when the resources are much less diversified. Its organization and inherent dynamics makes it similar to many alife simulations in which, by acting, these simple agents modify their environment whose perception by the other agents impact in turn their future behavior.

References

1. Palmer, R.G., Arthur, W.B., Holland, J.H., Le Baron, B., Taylor, P.: Artificial Economic Life: A Simple model of a stockmarket. Physica D 75, 264–274 (1994)
2. Derveeuw, J.: Market dynamics and agent behaviors: a computational approach. Artificial Economics 564, 15–27 (2005)
3. Bouchaud, J.-P., Mézard, M.: Wealth condensation in a simple model of economy. Physica A 282 (2000)
4. Derveeuw, J., Beaufils, B., Mathieu, P., Brandony, O.: Un modèle d'interaction réaliste pour la simulation de marchés financiers. In: Proceedings of the Quatrièmes Journées Francophones des modèles formels de l'interaction (2007)
5. Ball, P.: Critical Mass. How one thing leads to another, 1st edn. Farrar, Straus and Giroux, New York (2004)

Swarm Cognition and Artificial Life

Vito Trianni and Elio Tuci

Institute of Cognitive Sciences and Technologies (ISTC)
National Research Council (CNR), Rome, Italy
{vito.trianni,elio.tuci}@istc.cnr.it
http://laral.istc.cnr.it/{trianni,tuci}

Abstract. Swarm Cognition is the juxtaposition of two relatively unrelated concepts that evoke, on the one hand, the power of collective behaviours displayed by natural swarms, and on the other hand the complexity of cognitive processes in the vertebrate brain. Recently, scientists from various disciplines suggest that, at a certain level of description, operational principles used to account for the behaviour of natural swarms may turn out to be extremely powerful tools to identify the neuroscientific basis of cognition. In this paper, we review the most recent studies in this direction, and propose an integration of Swarm Cognition with Artificial Life, identifying a roadmap for a scientific and technological breakthrough in Cognitive Sciences.

1 Introduction

What do ants and neurons have in common? A bit of reasoning reveals that they share more than one would intuitively think. An ant is part of a colony, much as a neuron is part of the brain. An ant cannot do much in isolation, but a colony is a highly resilient adaptive system. Similarly, a neuron is individually able just of limited interactions with other neurons, but the brain displays highly complex cognitive processes. In other words, both ants and neurons behave/act in perfect harmony with other conspecifics/cells to accomplish tasks that go beyond the capability of a single individual. Self-organisation is the common mechanism that allows simple units—e.g., ants and neurons—to display complex spatio-temporal patterns. As a consequence, colony behaviour and cognitive processes can be explained in terms of self-organising rules of interaction among the low-level units and their environment. By describing the behaviour of ants, Aron *et al.* recognise that "while no individual is aware of all the possible alternatives, and no individual possesses an explicitly programmed solution, all together they reach an 'unconscious' decision" [1]. This is particularly true also for neural systems, where the relevance or the meaning of the self-organised pattern is not found at the individual level, but at the collective one.

Recent work recognises this close relationship between brains and swarms [2–4], giving birth to a novel approach in the study of collective intelligence and computational neuroscience. This is the *Swarm Cognition* approach, which aims at encompassing the above mentioned disciplines under a common theoretical

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 270–277, 2011.

and methodological framework. In this paper, we suggest that Artificial Life can give an essential contribution to Swarm Cognition studies. In fact, by synthesising distributed models of cognitive processes through ALife techniques, it could be possible to discover the underlying mechanisms common to swarms and to the vertebrate brain.

In the following, we will outline the background of Swarm Cognition, identified in studies of self-organising behaviours and in computational neuroscience (see Section 2). We continue in Section 3 by reviewing two recent studies that belong to Swarm Cognition, and we finally discuss how ALife can contribute in this direction in Section 4. Section 5 concludes the paper.

2 Background

The foundations of Swarm Cognition has to be found in the study of self-organising systems, particularly biological systems that can display cognitive behaviour, which are treated in Section 2.1, and in computational models of brain functions, discussed in Section 2.2.

2.1 Self-organisation in Biological Systems

Self-organising systems can be found in living and non-living matter. Self organisation refers to a spatio-temporal pattern (e.g., a collective behaviour or a physical structure) that is not explicitly programmed in each individual component of the system, but emerges from the numerous interactions between them. Each component only follows simple individual rules, which are performed with approximation on the basis of local information only, without any global map or representation [5]. Self-organised behaviour has been demonstrated in real biological societies, particularly in insects, but also in fish, birds and mammals, including humans (for some recent reviews, see [5–8]).

The basic ingredients of self-organisation are often recognised in *multiple interactions*, which generate *positive* and *negative feedback* mechanisms that allow the system to amplify certain *random fluctuations*, and to control the evolution of a coherent spatio-temporal pattern. A self-organising system is therefore able to achieve and sustain a certain spatio-temporal structure despite external influences [5]. A slightly different view of self-organisation focuses on its dynamical aspects, by describing the self-organising system as a complex dynamical system close to a bifurcation point. This means that the system, upon variation of some control parameter—e.g., temperature or chemical concentration—rapidly changes presenting new spatio-temporal patterns—e.g., a new type of collective behaviour or physical structure. This latter view of self-organisation is particularly relevant for Swarm Cognition studies. Indeed, it suggests that decision-making processes can be seen as the result of a bifurcation of a complex dynamical system. This system is formed by simple units that interact to produce the global spatio-temporal pattern, which results in the self-organised decision. There is clearly room to include in this definition also distributed processes that take place in the vertebrate brain. Here, the system units are individual neurons or neuronal assemblies, and the interactions are in form of inter-neuron communication. As we shall discuss in the next section, the dynamical and self-organised aspects of cognition are recently acquiring more and more attention.

2.2 Computational Neuroscience

Modelling of brain regions is not a novel endeavour: a long research tradition attempted to shed some light on the mechanisms at the basis of human reasoning, not without any success. Early in the mid fifties, Connectionism postulated the use of artificial neural networks (ANNs) as tools to study cognitive phenomena, without the need of knowledge representation, symbols and abstract reasoning [9]. With the advent of Computational Neuroscience, researchers have started to recognise the exquisitely dynamical traits of cognitive processes [10]. Dynamical systems theory is recently acquiring more and more attention in cognitive sciences as it can give explanations of cognitive phenomena while they unfold over time. Concepts like "attractor" and "bifurcation" start to be commonly used, and dynamical models are developed—just to name a few—to give new answers to classic psychology debates such as the A-not-B error in infant reaching [11], or to account for intrinsically dynamical processes such as interlimb coordination [12, 13]. To date, connectionist models are merged with the dynamical systems approach, recognising that cognitive processes are the result of a complex web of interactions in which both time-dependent and topological factors play a crucial role. In [14], Deco *et al.* propose the study of brain functional organisation at different space-time description levels, in order to understand the fundamental mechanisms that underpin neural processes and relate these processes to neuroscience data. However, so far there has been only limited room for holistic explanations of cognitive processes at different levels of description. Neither the relation with embodiment and environmental interactions has been thoroughly investigated. As we shall see in the following, the Swarm Cognition approach, by drawing parallels between swarm behaviours and the vertebrate brain, targets distributed processes in which cognitive units act in interaction with their environment, therefore attempting an holistic explanation of the phenomena under observation.

3 Case Studies: From Collective Intelligence to Cognition

Animal groups often display collective behaviours that allow to regulate the activities of the group maintaining a coherent organisation. In [2], Couzin observes that the dynamics of group behaviour show interesting similarities with those of cognitive processes in the brain. Multistability, non-linear responses, positive and negative feedback loops, population averaging and consensus decision-making (winner-takes-all) are the ingredients of cognitive process both in animal groups and in the brain. Recent studies argue that the massively parallel animal-to-animal interactions which operationally explain collective processes of natural swarms are functionally similar to neuron-to-neuron communication which underlie the cognitive abilities of living organisms, including humans [3, 4]. In this section, we briefly review these studies highlighting the main features of Swarm Cognition.

3.1 Swarm Cognition in Honey Bees

The nest site selection behaviour of honey bees *Apis mellifera* is the starting point taken by Passino *et al.* for a comparison between the decision-making abilities displayed by the swarm and the cognitive functions of primate brains [3]. Honey bees select a new nest site through a self-organising process, which is mainly based on a positive feedback mechanism that differentially amplifies the perceived quality of discovered nest sites. Scout bees explore the area surrounding the swarm in search of valuable sites. When they discover a potential nest that has a *supra-threshold* perceived quality, they return to the nest and perform a *waggle dance* to recruit other scouts. The higher the perceived quality, the longer the waggle dance, the stronger the recruitment. In this way, the differences between low quality nesting sites are amplified, allowing to quickly discard poor sites in favour of the better ones. When a sufficient number of scouts has been recruited to a nesting site (i.e., a *quorum* is reached), a second phase is triggered that leads to the lift-off of the entire swarm.

It is important to notice that the selection of the best nest site is not performed by individual bees that directly compare different options by visiting different sites. Neither it is based on the comparison of different waggle dances. The competition between sites is performed at the level of the group through recruitment and quorum sensing, and not at the level of the individual bee. In this respect, a strong parallelism with brain functions can be recognised. Scout bees perform functions similar to individual neurons in the brain. Waggle dances are analogous to action potentials, and the threshold in the estimated quality of a discovered nest corresponds to the neuron activation threshold. The parallelism between swarm and brain goes beyond these similarities, including lateral inhibition, feature detection and attention. By developing a model of nest site selection, tests have been performed to assess the discrimination abilities between different sites, as well as the ability of the swarm as a whole to discard distractors and focus the attention on the highest quality site [3].

3.2 Decision-Making in Brains and Insect Colonies

The work of Marshall *et al.* [4] goes a step further. They again focus on nest site selection in rock ants (*Temnothorax albipennis*) and in honey bees, and show that it has the same properties of diffusion models used to characterise decision-making in the cortex [15]. Diffusion models describe the accumulation of evidences trough time during a decision-making process as a random walk with normally distributed step size (Wiener process or Brownian motion), subject to a constant drift toward the better choice. When a threshold is passed toward one or the other alternative, the decision is taken.

The remarkable fact is that similar diffusion models provide a statistically optimal speed-accuracy tradeoff in decision-making, which reflects the tension between the need to take a quick decision and the need to wait until enough evidence is accumulated in favour of one or the other option. In fact, by varying the decision threshold, the model can account for quick but unsafe decisions, or for more conservative but time-demanding ones. The speed accuracy tradeoff is well known from psychological experiments in humans and animals, and has been also recognised in the nest site selection behaviour of rock ants: under stormy weather conditions, ants lower their decision threshold (i.e., the quorum necessary to select a site), therefore performing a quick decision at the expense of a higher error rate [16].

In [4], the authors analyse a model of the ants nest site selection, as well as two models of the same process performed by honey bees, also described above. These models differ mainly in the possibility for scouts of direct switching of commitment between alternative sites, without passing through an "uncommitted" state. The model that allows direct switching corresponds to a diffusion model, accounting for statistical optimality of the nest selection behaviour, and suggesting that neural and swarm decision-making can be explained by functionally similar mechanisms.

4 The Artificial Life Approach

In the previous section, we have described how comparative studies of cognitive processes and swarm behaviours highlight surprising similarities. We believe that this is not a fortunate case, and we suggest that similar comparisons should be further developed, in search of common working mechanisms. This is the goal of Swarm Cognition studies that involve the observation of the biological reality. In this paper, we propose Artificial Life as a complementary approach to the investigation of Swarm Cognition. ALife is intimately connected to Cognitive Sciences. Bedau recognises this as he notices that "one of the fundamental open problems in artificial life is to explain how robust, multiple-level dynamical hierarchies emerge solely from the interactions of elements at the lowest-level. This is closely analogous to the problem in cognitive science of explaining how cognitive capacities ultimately emerge from the interactions of non-cognitive elements like neurons" [17].

We propose the development of an ALife approach to Swarm Cognition, aiming at improving our understanding of the mechanisms behind cognitive processes by synthesising such processes in artificial systems. By paraphrasing Langton [18], we claim that ALife and Swarm Cognition can contribute to Cognitive Sciences by locating *cognition-as-we-know-it* within the larger picture of *cognition-as-it-could-be*. This means that the ALife approach to Swarm Cognition, by building bridges between computational neuroscience and swarm intelligence, searches for the underlying mechanisms of cognition being inspired, rather than constrained, by the biological reality.

4.1 Beyond Connectionism

A first contribution of ALife to Swarm Cognition is providing explanations of cognition as the result of self-organising processes through computational models. Indeed, there is no doubt that cognitive processes involve a massive amount of neuron-to-neuron interactions. There is also no doubt that neurons are organised in assemblies of coherent activities, and that they are spatially and functionally segregated in different brain areas. It is anyway difficult to unveil causal relationships between neurophysiological phenomena and cognition, without reducing the latter to the former. The Swarm Cognition approach is expected to shed light on such complex issue by explicitly searching for the emergence of measurable phenomena from the interaction of low-level *cognitive units*. These cognitive units should not necessarily be related to biological reality—e.g., neurons, neuronal assemblies or populations—but may well be closer to a bee or to a generic artificial agent.

The main goal of these studies should be the identification of the mechanisms underlying cognitive processes, as a result of the dynamical interactions among cognitive units. The simulated approach brings these activities closer to computational neuroscience, and cross-fertilisation between the two disciplines should be promoted whenever possible, in the attempt to complement neruophysiological models and fit, at least qualitatively, experimental data.

4.2 Embodiment and Swarm Robotics

A distinctive feature of ALife is the attempt to study how life occurs not only in computer simulation, but also in the physical world. "Wet" ALife seeks the synthesis of living systems out of biochemical substances. Apart from this, robotics is the other field of confrontation with the physical world. In Bedau's view, (evolutionary) robotics "is artificial life's most direct overlap with cognitive science, as its aim is to synthesize autonomous adaptive and intelligent behavior in the real world" [17]. When adaptive behaviour is performed by a swarm of robots, we deal with a *Swarm Robotics* system, characterised by limited abilities at the level of the individual robot, which can anyway perform complex tasks by coordinating in a group.

There are multiple reasons that justify the swarm robotics approach to cognition. First of all, it is important to stress the relevance of using robots to study cognitive processes. Robots are artifacts with a physical body situated in the physical world, with physical sensors and actuators to perceive and act within their environment. The embodiment of the robots is a very important aspect for the study of cognitive behaviour, which is not the result of "reasoning" alone, but is rather the result of the dynamical interactions between brain, body and environment. Robots therefore are excellent tools to study such brain-body-environment dynamics and their bearing on the emergence of cognitive abilities such as categorisation, decision making, attention and learning [19].

Additionally, a peculiar feature of Swarm Robotics systems is the transfer of behavioural complexity from the individual to the interactions among individuals. Brought to the limit, this vision sees robots as neuron-like devices that can move in the environment and interact, physically or trough communication, with other robots, while bringing forth complex cognitive processes as a whole. Within the Swarm Cognition framework, this transfer of complexity from the individual behaviour to the interactions among individuals is fundamental to understand how cognitive processes can be supported by distributed systems. Swarm Robotics is therefore the only mean to study self-organisation in embodied and situated systems. Each robot is a cognitive unit, in this case, playing either the role of the individual insect in a swarm, or the role of a neuron or an assembly in the brain. In our opinion, all these aspects make Swarm Robotics the most appropriated method to instantiate the Artificial Life approach to Swarm Cognition.

4.3 Bridging the Gap between Behaviour and Cognition

Comparative studies in Swarm Cognition can pinpoint the relevant mechanisms that support cognition, a significant breakthrough in Cognitive Sciences. The ALife approach offers the possibility to synthesise cognitive process in artificial brains as well as in artificial swarms. With such a dual approach, it is possible to study similar problems, such as decision-making or attention, in search of common mechanisms. Similar discoveries in artificial systems may well be generalisable to natural ones, when some biological plausibility has been preserved into the models.

Additionally, the knowledge acquired in Swarm Cognition studies could also be integrated in a single experimental scenario in which a swarm of robots is governed by neurocomputational controllers. In this way, the ALife approach to Swarm Cognition is expected to advance the state of the art in robotics and computational neuroscience. In fact, by integrating neurocomputational controllers in swarm of robots, an highly complex system could be synthesised, composed of three different organisational levels hierarchically stacked, from the neurocontroller internal dynamics, through the embodied cognition displayed by the individual robot, up to the cognitive processes displayed by the group dynamics. In this way, we could have a physical realisation of multiple-level dynamical hierarchies that truly generate cognition from the bottom-up.

5 Conclusions

In this paper, we have introduced Swarm Cognition as a multidisciplinary research field that bridges studies in collective intelligence and computational neuroscience under a common theoretical and methodological framework. We suggest that ALife can give a significant contribution, by developing synthetic models of *cognition-as-it-could-be.* This concerns both simulated models of the brain and swarm robotics systems. The goal is understanding how cognitive processes are brought forth as transient dynamics emerging from massively parallel interactions among cognitive units, be they simulated neurons or physical robots.

References

1. Aron, S., Deneubourg, J.L., Goss, S., Pasteels, J.M.: Functional self-organization illustrated by inter-nest traffic in ants: The case of the argentinian ant. In: Alt, W., Hoffman, G. (eds.) Biological Motion. Lecture Notes in BioMathematics, vol. 89, pp. 533–547. Springer, Berlin (1990)
2. Couzin, I.: Collective cognition in animal groups. Trends in Cognitive Sciences 13(1), 36–43 (2009)
3. Passino, K., Seeley, T., Visscher, P.: Swarm cognition in honey bees. Behavioral Ecology and Sociobiology 62, 401–414 (2008)
4. Marshall, J.A.R., Bogacz, R., Dornhaus, A., Planqué, R., Kovacs, T., Franks, N.R.: On optimal decision-making in brains and social insect colonies. Journal of the Royal Society Interface 6, 1065–1074 (2009)
5. Camazine, S., Deneubourg, J.L., Franks, N., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-Organization in Biological Systems. Princeton University Press, Princeton (2001)
6. Couzin, I.D., Krause, J.: Self-organization and collective behavior of vertebrates. Advances in the Study of Behavior 32, 1–75 (2003)
7. Sumpter, D.: The principles of collective animal behaviour. Philosophical Transactions of the Royal Society of London: Series B 361, 5–22 (2006)
8. Strogatz, S.H.: Sync: The emerging science of spontaneous order. Hyperion Press, New York (2003)
9. Rumelhart, D., McClelland, J.: Parallel Distributed Processing, vol. 1(2). MIT Press, Cambridge (1986)
10. Beer, R.D.: Dynamical approaches to cognitive science. Trends in Cognitive Sciences 4(3), 91–99 (2000)
11. Thelen, E., Schöner, G., Scheier, C., Smith, L.B.: The dynamics of embodiment: A field theory of infant perseverative reaching. Behavioral and Brain Sciences 24(1), 1–34 (2001)
12. Fitzpatrick, P., Schmidt, R.C., Carello, C.: Dynamical patterns in clapping behavior. Journal of Experimental Psychology: Human Perception and Performance 22(3), 707–724 (1996)
13. Schöner, G.: Timing, clocks, and dynamical systems. Brain and Cognition 48, 31–51 (2002)
14. Deco, G., Jirsa, V., Robinson, P., Breakspear, M., Frinston, K.: The dynamic brain: From spiking neurons to neural masses and cortical fields. PLoS Computational Biology 4(8) (2008)
15. Ratcliff, R., Smith, P.L.: A comparison of sequential sampling models for two-choice reaction time. Psychological Review 111, 333–367 (2004)
16. Franks, N.R., Dornhaus, A., Fitzsimmons, J.P., Stevens, M.: Speed versus accuracy in collective decision-making. Proceedings of the Royal Society B: Biological Sciences 270(1532), 2457–2463 (2003)
17. Bedau, M.A.: Artificial life: organization, adaptation and complexity from the bottom up. Trends in Cognitive Sciences 7(11), 505–512 (2003)
18. Langton, C.G.: Artificial life. In: Langton, C.G. (ed.) Artificial life, pp. 1–47. Addison-Wesley, Reading (1988)
19. Harvey, I., Di Paolo, E.A., Wood, R., Quinn, M., Tuci, E.: Evolutionary robotics: A new scientific tool for studying cognition. Artificial Life 11(1-2), 79–98 (2005)

Life Engine - Creating Artificial Life for Scientific and Entertainment Purposes

Gonçalo M. Marques[1], António Lorena[1], João Magalhães[1], Tânia Sousa[1],
S.A.L.M. Kooijman[2], and Tiago Domingos[1]

[1] Environment and Energy Section, DEM, Instituto Superior Técnico,
Lisbon, Portugal
{goncalo.marques,antonio.lorena,joao.magalhaes,
tania.sousa,tdomingos}@ist.utl.pt

[2] Department of Theoretical Biology, Vrije Universiteit,
Amsterdam, The Netherlands
bas.kooijman@falw.vu.nl

Abstract. The Dynamic Energy Budget (DEB) theory has become a fundamental tool in modeling the metabolic behaviour of organisms. Its capacity to describe the biological aspect of life alone justifies its applicability in Artificial Life. Aware of this potential, the DEB research group in Instituto Superior Técnico (IST) in Lisbon has joined the videogame company Biodroid Entertainment in the Life Engine project. This project aims to develop a library for scientific purposes but also to create a biology engine for videogames. From the scientific point-of-view, this library is intended to be the standard tool for DEB researchers and, at the same time, to popularize DEB theory in other scientific communities, such as the AL community.

Keywords: Dynamic Energy Budget, Artificial Life.

1 Introduction

Bedau [1] published a list of open problems in Artificial Life (AL) in which the simulation of a unicellular organism during its life cycle was included. It was argued that this should be done through a bottom-up simulation of the genetic and regulatory networks of the cell. From the interaction of these low-level entities, global properties and processes should emerge. The search for patterns that emerge from the interaction of multiple low-level entities, while having well known merits, may have caused not only AL but also, for some time, Biology to overlook the similarities and patterns that are common to all organisms.

The aim of Dynamic Energy Budget (DEB) theory [2, 3, 4] is to explain these patterns. DEB theory is a general mathematical theory at the organism level applicable to all taxonomic groups with implications at the sub- and supra-organism levels. Since DEB is a non-species specific theory, it can describe all types of organisms, from bacteria to trees, with the same theoretical framework.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 278–285, 2011.

DEB theory is based on simple mechanistic rules for the uptake of energy and nutrients and the consequences of these rules for physiological organization along the life cycle of organisms. It has many empirical models as special cases, such as Droop's model for the nutrient limited growth of algae, von Bertalanffy's model for the growth of animals or Kleiber's law for respiration. The large collection of empirical support for all these empirical models that accumulated in the literature and the evidence that people working with DEB have accumulated during the 30 years of DEB research makes DEB theory probably one of the best tested theories in biology. It has already several practical applications, namely in toxicology (where its use is recommended by ISO and OECD), environmental engineering and biological engineering.

The capacity of DEB theory to describe the biological aspect of life alone justifies its applicability in AL. Moreover, the theoretical richness of DEB theory allows the generation of complex and novel individuals or agents, which has practical gains for entertainment purposes. Aware of this potential, the DEB research group in Instituto Superior Técnico (IST) in Lisbon has joined the videogame company Biodroid Entertainment in the Life Engine project. This project aims to develop a library not for scientific purposes but also to create a biology engine for videogames.

From the scientific point-of-view, the Life Engine project has three main objectives: (i) supply a standard tool for DEB researchers around the world, which allows simulations from the individual to the ecosystem level and, with this tool, (ii) develop the body of knowledge concerning DEB theory, particularly the interaction between different organizational levels, and, finally, (iii) popularize DEB theory in other scientific communities such as the AL community.

The present paper reports the first stage of the project: the development of the individual organism with the capacity to prey, process food, grow, mature, reproduce and die by ageing. These processes are well defined in the DEB theory, assuring a realistic description of the organism. Section 2 details the standard DEB organism, in which emphasis is given to state variables, fluxes and processes, and from the life cycle perspective, in which emphasis is given to life stages and transitions. This is followed by section 3 where focus is put on how a DEB framework helps in building complexity automatically. Finally in section 4 we summarize the main points, describe the current phase of implementation of the project and outline future work.

2 Standard DEB Model

At a given instant, an organism of a particular species is completely defined by a set state variables. DEB theory presents a set of mechanistic rules that determines the organism's energy and mass fluxes, which will be responsible by the evolution of the state variables through time (see Fig. 1).

At a higher level, each species is defined by a set of parameters. Two organisms of the same species will have the same parameters (apart from small variations

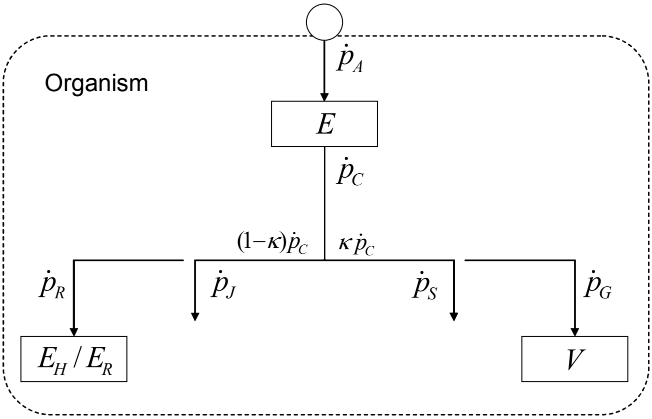


Fig. 1. Diagram of the state variables that define an organism and the fluxes modeled by DEB. A fixed portion of ingested food is assimilated ($\dot{p}_A$) into its reserve (E). Reserve is mobilized for the several tasks the organism needs to perform ($\dot{p}_C$). A fraction κ is first used for somatic maintenance ($\dot{p}_S$) and then for growth ($\dot{p}_G$), which increases its structure (V). The remaining $(1-\kappa)$ fraction is used for maturity maintenance ($\dot{p}_J$) and maturation or reproduction ($\dot{p}_R$), depending on the life stage of the organism. Maturity (E_H) and the reproduction buffer (E_R) are the other two state variables needed to define the organism.

due to genetic variation). Moreover, the framework of DEB theory provides rules for the extrapolation of parameters between species, which we will come back to in section 3.

2.1 State Variables

In the standard DEB model, the organism is mainly defined by two state-variables: **reserve** (E) and **structure** (V). These are two generalized aggregated compounds that have a fixed chemical composition. As the amounts of reserve and structure evolve through time the organism can and will have a varying chemical composition.

As food is assimilated, it is stored on the reserve. On the other hand, reserve is continually being mobilized and used for all the metabolic purposes. Additionally, reserve has no maintenance costs in contradistinction with structure, which needs to be maintained. When relating these two state variables to body size, only structure is taken to be relevant. Physiological processes are either proportional to surface area, e.g. uptake of nutrients, or proportional to volume, e.g. maintenance costs. Thus, the relationship between surface area and volume – body shape – controls the metabolism of an organism.

Complementing the view of the organism's instantaneous fluxes, one can also look at the organism as it evolves through its life cycle. The organism's ontogeny is characterized by stages and respective transitions. For an example, see Fig. 2. DEB theory allows for the modeling of the organism's ontogeny due to the existence of the **maturity** state variable (E_H). A major part of multicellular organisms follow a standard life cycle, which is characterized by three stages: embryo, when the organism does not feed from the environment and thus only uses energy from its initial reserve; juvenile, in which the organism starts feeding, and it continues to allocate energy to maturation and not yet to reproduction; and adult, when the organism stops allocating energy to maturation and diverts the corresponding flux to reproduction. Birth (from embryo to juvenile) and puberty (from juvenile to adult) transitions are defined by maturity values, E_H^b and E_H^p respectively. When the organism reaches the adult phase, it starts to accumulate energy in the **reproduction buffer** (E_R). This buffer has the same chemical composition of the reserve. On the act of reproduction the reproduction buffer is emptied out. Its content is transformed into gametes that will originate new organisms.

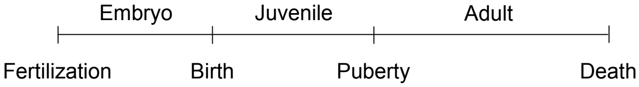


Fig. 2. Diagram of the standard life cycle of a multicellular organism

2.2 Processes

In this section we will characterize the main processes and associated parameters, and see how they impact the organism ontogeny. We will follow closely the diagram in Fig. 1.

For the major part of its life, an organism needs to assimilate food. **Assimilation** is the process of transforming food in to reserve (state variable of the organism). In DEB theory the assimilation flux ($\dot{p}_A$) is dependent on food availability and is proportional to the surface area of the organism. Each organism has a surface-area-specific maximum assimilation rate $\{\dot{p}_{Am}\}$. A bigger value of $\{\dot{p}_{Am}\}$ translates into larger amounts of food being assimilated at a given instant, for the same food level. This would imply that, all other things being equal, an organism with a larger $\{\dot{p}_{Am}\}$ would reach its maximum volume in a shorter period of time.

Reserve is mobilized for the different tasks the organism needs to perform, e.g. growth. In DEB this mobilization is called **catabolism**. Based only on first principles (see [5]) one can compute the reserve dynamics and the catabolic flux ($\dot{p}_C$). For the specification of the reserve dynamics, one needs to introduce the specific-energy conductance $\dot{v}$. This parameter regulates the rate of output from the reserve. The resulting catabolic flux will be used for four purposes: somatic maintenance $\dot{p}_S$, growth $\dot{p}_G$, maturity maintenance $\dot{p}_J$ and maturation (or reproduction) $\dot{p}_R$. DEB makes the assumption that a fraction κ of the catabolic flux will be used for somatic maintenance plus growth, and the remaining $(1-\kappa)$ fraction will be used for maturity maintenance and maturation (or reproduction). κ belongs also to the set of parameters that defines the organism, and the fact that it is a constant implies the decoupling of growth from reproduction, opening the possibility of an organism reproducing while growing. Different values of κ mean different balances between growth and reproduction. A larger κ would delay maturation and reproduction, while hastening growth.

The fraction κ of the catabolic flux is first used for **somatic maintenance**, i.e. to pay the cost of maintaining structure, which includes processes such as the turnover of proteins or the continuous production of hair, feathers or scales. The most important component of somatic maintenance is proportional to the structural volume and controlled by the volume-specific somatic maintenance cost $[\dot{p}_M]$. As $[\dot{p}_M]$ states the cost of maintenance, it impacts on the amount on $\kappa\dot{p}_C$ that remains for growth, and therefore a larger value would shrink the maximum volume of the organism.

The remaining part of $\kappa\dot{p}_C$ that was not used for somatic maintenance is used for **growth**. Reserve is transformed into structure and the inverse of the volume-specific cost of structure $[E_G]$ is the conversion factor. This parameter impacts directly on growth. A greater $[E_G]$ means that the same amount of reserve is transformed in a smaller amount of structure.

A similar system of priorities is used for the fraction $(1-\kappa)$ of the catabolic flux. This flux is first used for **maturity maintenance**. $\dot{p}_J$ is proportional to maturity with a proportionality constant $\dot{k}_J$. As this is the primary use of $(1-\kappa)\dot{p}_C$, a larger $\dot{k}_J$ leaves a smaller amount of reserve to be allocated to **maturation** or, after puberty, to **reproduction**. Reserve that goes to reproduction is first collected in the reproduction buffer E_R. There are handling rules, dependent on the species, that model the transformation of this reserve into gametes (e.g. eggs), that typically leave the body upon formation. At their formation, the gametes can be said to consist exclusively of reserve. The amount of structure is negligibly small and the level of maturity is null.

Finally, DEB can also capture the process of **aging** and its effects on the survival rate, making use of the idea that damage-inducing compounds (changed genes, affected mitochondria), accumulate at a rate that is proportional to the mobilization rate $\dot{p}_C$, and these in turn produce damage at a constant rate. To model this process one needs to keep track of the damage density.

2.3 Equation Set and Parameter Space

After going through the main processes that occur in the organism, one can summarize them in the expressions of the dynamics of the state variables. Reserve and structure are controlled by the following set of coupled differential equations:

$$\frac{dE}{dt} = \frac{\{\dot{p}_{Am}\}f - [E]\dot{v}}{L} \tag{1}$$

$$\frac{dV}{dt} = V\frac{[E]\dot{v}/L + [\dot{p}_M]/\kappa}{[E] + [E_G]/\kappa} \tag{2}$$

Where $[E]$ is the reserve density (E/V), and L is the structural length $(V^{1/3})$. This set of coupled differential equations encapsulates the life history of the organism, given that we know its environment. The functional response f depends on the food level, and gives us the link to the environment.

As for the maturity and the reproduction buffer, their dynamics is dependent on reserve and structure but, as was shown above, does not have any effect on them. Their equations are:

$$\dot{p}_R = (1-\kappa)E\frac{[E_G]\dot{v}/L + [\dot{p}_M]}{\kappa[E]+[E_G]} - \dot{k}_J E_H \tag{3}$$

$$\frac{dE_H}{dt} = \begin{cases} \dot{p}_R \ , \ E_H < E_H^p \\ 0 \ \ , \ E_H \geq E_H^p \end{cases} \qquad \frac{dE_R}{dt} = \begin{cases} 0 & , E_H < E_H^p \\ \kappa_R \dot{p}_R & , E_H \geq E_H^p \end{cases} \tag{4}$$

Where κ_R is the fraction of the reproduction flux that is fixed in the eggs. The branches exist due to the puberty transition, when $\dot{p}_R$ is diverted from maturation to reproduction.

3 Generating Complexity

The theoretical richness of DEB theory allows the automatic generation of complex and novel metabolic behaviours without neglecting realism. The two most straightforward and important automatic generators of complexity are the environment and the parameter set. Below we list several of the features that will help to generate complexity.

Environment. As assimilation depends on food availability, similar organisms exposed to different food levels will have different life histories. But even in the case of organisms in the same food environment, the stochasticity inherent to predation can make two organisms grow in very distinct forms [6].

Parameters. The parameter set of an individual will also have a great impact on its ontogeny. Organisms of the same species will have very similar sets of parameters. The small variation, corresponding to real variations in nature, is implemented stochastically. On the other hand, DEB provides rules for the relation between parameters of different species (which include the usually referred body size scaling relations). These relations provide a way of automatically generate different realistic species.

Life stages. Departing from the life cycle we have described, one can include new life stages, such as larval or pupae, or more unusual stages, e.g negative juvenile, where the organism starts to allocate to reproduction before birth (which is usual in aphids and gives rise to the existence of matryoshka-like reproduction). In fact, there can also be stages characterized by the organism's behaviour, which can have effects on some parameter values.

Body shape. In section 2.1 we stressed the importance of the relation of surface area and structural volume, since there are processes that depend on one or the other. We are most used to the surface area being proportional to $V^{2/3}$, which applies to organism that grow proportionally in all dimensions (isomorphs). But this is not always the case. For instance, there are organisms that grow only in one dimension and in this case surface area is proportional to V. Therefore one needs also to generalize the surface area - volume relation.

Multivariate DEB models. So far we have discussed standard DEB theory, which in fact is fully capable of describing most organisms in typical environmental conditions, but further detailing is possible and useful to model more complex organisms. There are organisms, or details in one organism, that need several reserves or several structures to be well modeled. For instance, heterotrophic organisms, that uptake different substrates (carbon, nitrogen, photons,...) by different routes, have to be modeled with more than one reserve. As for structures, these are usually associated with different organs or body parts. The existence of several structures allows organs or body parts to grow independently of the remaining structures, which can happen to organisms when they are adapting to the environment (e.g. algae reduce chlorophyll in excessive light,). Another example of several structures is the plants, where the root and shoot are modeled as different structures.

4 Conclusion

Bedau [1] stated that AL aimed to develop a coherent theory of life in all its manifestations, while DEB is meant to be a single quantitative framework which comprises all living organisms. We think DEB is the perfect tool for the realization of this goal, firstly on the organism level, but with several implications at the sub- and supra-organism levels.

The fact that DEB is based on simple mechanistic rules and allows for the automatic generation of complexity, without neglecting realism, makes it the perfect instrument to be used as the base of a biology engine. Furthermore, each species is characterized in DEB theory by a set of parameter values and DEB theory predicts how these sets can be extrapolated from one species to another. Apart from the evident economy in coding, the DEB engine provides also an enormous potential for realistic and unexpected complexity.

The scientific and entertainment applications should make use of each other strengths to make progress. In fact applications in which they overlap are envisioned in several fields. For example, massive multiplayer online games allow thousands of players to interact with each other and with AL in search of solutions to problems posed by scientists. This has been advocated by various authors – see [8].

The development of the Life Engine project will take place during the next two to three years, with three main axes of research and implementation. The first axis is on the generalization of DEB theory. With the starting point of the standard DEB model, which can already describe most organisms, the work has been aimed at understanding the rules that describe the interactions between new state variables (reserves and/or structures) and that will allow the automatic generation of new types of organisms. The second axis concerns the organizational levels, and we will follow a bottom-up approach. From the organism level, which is already implemented in C^{++} and returning results, we will implement population through individual-based models (IBM). At this stage we are setting up the the IBM protocols [7] that will guide the simulations. The results from IBM will in turn be used to build a sound theory concerning structured populations of DEB organisms. This will allow us to simulate simultaneously discretized populations and continuous (structured) populations. Finally, the third axis concerns the development of a basic artificial intelligence such that the global behaviour of the system is in accordance to empirical observations.

The biology game engine will supply game designers with a tool to create realistic organisms, populations and ecosystems, each with a wide range of corresponding processes, using a single framework. Some of the various advantages are: (i) less coding is necessary for the creation of organisms - all organisms are instances of the same class, defined by a point in the parameter space; (ii) simpler algorithms for the interaction between objects - different objects are in essence equal, thus there is no need to create explicit functions for each interaction type; and (iii) easier generation of novelty - unorthodox organisms are just points in the parameter space. The application in videogames may span from those focused in the individual, in which processes at the organism level are more relevant, to ecosystem focused games, in which interactions and emerging properties are more relevant.

Acknowledgment. This work was supported by the project Life Engine – 3111 of QREN. The work of Gonçalo M. Marques was financed by FCT through the grant SFRH/BPD/27174/2006.

References

[1] Bedau, A., et al.: Open Problems in Artificial Life. Artificial Life 6, 363–376 (2000)
[2] For a DEB introduction, http://www.bio.vu.nl/thb/deb/
[3] Kooijman, S.A.L.M.: Dynamic Energy Budgets in Biological Systems, 1st edn. Cambridge Univ. Press, Cambridge (1993)
[4] Kooijman, S.A.L.M.: Dynamic Energy and Mass Budgets in Biological Systems, 2nd edn. Cambridge Univ. Press, Cambridge (2000)
[5] Sousa, T., Domingos, T., Kooijman, S.A.L.M.: From empirical patterns to theory: A formal metabolic theory of life. Phil. Trans. R. Soc. B 363, 2453–2464 (2008)
[6] Kooijman, S.A.L.M.: Social interactions can affect feeding behaviour of fish in tanks. To appear on: J. Sea Research 62 (2009)
[7] Grimm, V., et al.: A standard protocol for describing individual-based and agent-based models. Ecological Modelling 198, 115–126 (2006)
[8] Vinge, V.: 2020 Computing: The creativity machine. Nature 440, 411 (2020)

An Analysis of New Expert Knowledge Scaling Methods for Biologically Inspired Computing

Jason M. Gilmore, Casey S. Greene, Peter C. Andrews,
Jeff Kiralis, and Jason H. Moore

Dartmouth College, Lebanon, NH. USA
{Jason.M.Gilmore,Casey.S.Greene,Peter.C.Andrews,
Kiralis,Jason.H.Moore}@dartmouth.edu
http://www.epistasis.org

Abstract. High-throughput genotyping has made genome-wide data on human genetic variation commonly available, however, finding associations between specific variations and common diseases has proven difficult. Individual susceptibility to common diseases likely depends on gene-gene interactions, i.e. epistasis, and not merely on independent genes. Furthermore, genome-wide datasets present an informatic challenge because exhaustive searching within them for even pair-wise interactions is computationally infeasible. Instead, search methods must be used which efficiently and effectively mine these datasets. To meet these challenges, we turn to a biologically inspired ant colony optimization strategy. We have previously developed an ant system which allows the incorporation of expert knowledge as heuristic information. One method of scaling expert knowledge to probabilities usable in the algorithm, an exponential distribution function which respects intervals between raw expert knowledge scores, has been previously examined. Here, we develop and evaluate three additional expert knowledge scaling methods and find parameter sets for each which maximize power.

1 Introduction

Advances in genotyping technology have made it possible for human geneticists to rapidly and accurately measure genetic variations across a large number of genetic markers. Our research focuses on a set of genetic variations known as single nucleotide polymorphisms (SNPs), where single DNA bases differ across individuals. However, determining which polymorphisms are correlated with disease states is a non-trivial exercise. Epistasis, low disease heritability, and informatics constraints present challenges to research efforts. Epistasis, where one gene's contribution to phenotype depends on the genotype of one or more other genes complicates the problem [1]. The study of single genes may therefore not be adequate and more complex models might be required. Heritability is the proportion of phenotypic variance that can be attributed to genotypic (i.e. SNP) factors in a population [2]. When heritability is low, noise obscures relationships between SNPs and disease, making it harder to characterize genetic associations.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 286–293, 2011.

Finally, the size of genomic datasets prevents exhaustive searching of combinations of attributes, necessitating methods which can more efficiently search vast noisy landscapes.

We have used a biologically inspired ant colony strategy to look for pairs of SNPs which are predictive of disease. In the ant system, simulated ants search the dataset for interactions between attributes using a positive feedback approach [3]. The ant system was implemented within the user friendly Multifactor Dimensionality Reduction (MDR) software package [4]. Due to its intuitive design and ready availability, the MDR GUI has been widely used for genetic analyses [5,6,7]. The ant system simulates a specified number of ants over generations where each ant is assigned two SNPs. Initially, each attribute of the dataset is given a selection probability which determines the likelihood that that attribute will be incorporated into an ant. Within MDR, we construct rules for the SNPs assigned to each ant. The rules are assessed for accuracy by the proportion of correct differentiations between case (disease) and control (healthy) subjects. At the end of each generation, the SNP selection probabilities are adjusted to account for the relative success of rules that were tested [8].

The ant system uses expert knowledge and we examine four methods for scaling and incorporating such information. We discuss the strengths and weaknesses of these approaches and we find three of the methods perform well at a genetically relevant heritability, of 0.1, when optimized parameters are used. Biologically inspired approaches, such as the ant colony optimization, allow for the characterization of genetic factors influencing human health in genome-wide datasets where other methods may not be effective.

2 Use and Scaling of Expert Knowledge

The ant system was implemented to use any expert knowledge source in which attributes with higher scores are more likely to be relevant. We use Tuned ReliefF weights as our expert knowledge source. Tuned ReliefF (TuRF) is an algorithm which is capable of detecting how well SNPs, in the context of other SNPs, are able to differentiate individuals with disease from those without [9]. TuRF has been shown to perform significantly better than other algorithms in a human genetics context, where large numbers of noisy attributes confound analyses.

We explore three new scaling methods, along with the one used in our previously developed ant system [8]. Two of these four methods use an exponential distribution function to transform expert knowledge scores into probabilities. Of these two, one uses a fitness based approach which respects the interval between expert knowledge scores [8], while the other method uses only the ranking of the SNPs. We will refer to these two methods as exponential-fitness and exponential-rank. Both are implemented with a user-adjustable parameter, θ, which can vary from 0 to 1. The lower the value of θ, the more likely that SNPs with high expert knowledge scores are selected over those with low scores.

The other two scaling methods use a linear distribution function. As with the exponential case, there are linear-fitness and linear-rank methods. For both linear methods, a window of allowable selection probabilities can be assigned to the SNP with the highest expert knowledge score and a user-adjustable parameter, *maxProb*, specifies this assignment. As *maxProb* varies through its range from 0% to 100%, the selection probability varies linearly from its minimum to its maximum value.

3 Linear and Exponential Scaling

In all four scaling methods, we order the attributes $A_1, A_2, \ldots, A_N$ so that their expert knowledge scores are increasing, and let s_i be the expert knowledge score of the i^{th} attribute A_i. We often work with t_i, the expert knowledge scores normalized so that they range from 0 to 1. Specifically,

$$t_i = \frac{s_i - s_1}{s_N - s_1} \quad \text{for} \quad i = 1, 2, \ldots, N.$$

In the linear-fitness method, we use

$$\frac{N \cdot P(A_N \text{ is selected}) - 1}{N - \sum_{i=1}^{N} t_i} (t_i - 1) + P(A_N \text{ is selected}) \tag{1}$$

for $P(A_i$ is selected), the selection probability of attribute A_i. Here $P(A_N$ is selected) is the probability that the attribute with the highest expert knowledge score is selected. To assure that

$$0 < P(A_1 \text{ is selected}) < P(A_N \text{ is selected}) < 1, \tag{2}$$

it must satisfy

$$\frac{1}{N} < P(A_N \text{ is selected}) < \frac{1}{\sum_{i=1}^{N} t_i}.$$

The parameter *maxProb*, which is entered as a percent, equals

$$\frac{P(A_N \text{ is selected}) - \frac{1}{N}}{\frac{1}{\sum_{i=1}^{n} t_i} - \frac{1}{N}} \cdot 100.$$

For the linear-rank method, we set

$$t_i = \frac{i-1}{N-1} \tag{3}$$

so that the distance between all adjacent t_i is the same, namely $\frac{1}{N-1}$. Then expression (1) becomes

$$P(A_i \text{ is selected}) = \left(2 \cdot P(A_N \text{ is selected}) - \frac{2}{N}\right) \cdot t_i + \frac{2}{N} - P(A_N \text{ is selected}).$$

This gives the probability that attribute A_i is selected, where, if the constraint (2) is to hold, we must have $\frac{1}{N} < P(A_N \text{ is selected}) < \frac{2}{N}$. Again, the value of $P(A_N$ is selected) can be specified by the parameter $maxProb$, which is the percentage the interval $\left(\frac{1}{N}, P(A_N \text{ is selected})\right)$ comprises of the interval $\left(\frac{1}{N}, \frac{2}{N}\right)$.

For the exponential-fitness method we use

$$P(A_i \text{ is selected}) = \frac{1}{\sum_{i=1}^{N} (\theta^{N-1})^{-t_i}} \left(\frac{1}{\theta^{N-1}}\right)^{t_i} . \tag{4}$$

For the rank based exponential method, we use equation (4), with t_i as in equation (3), to get

$$P(A_i \text{ is selected}) = \frac{\theta^{-1} - 1}{\theta^{-N} - 1} \left(\frac{1}{\theta}\right)^{i-1} . \tag{5}$$

In both cases, θ is a user-adjustable parameter which must satisfy $0 < \theta < 1$.

4 Data Simulation and Analysis

The ant colony based search algorithm used by the MDR software package depends on several user-adjustable parameters and allows for the incorporation of expert knowledge scores. Scaled expert knowledge scores are used both to set initial selection probabilities for each SNP and in the update rule for pheromone deposition. To find the optimal parameter set for each of the four expert knowledge scaling methods, we performed a full factorial parameter sweep across $maxProb/\theta$, β, the retention factor, and the number of ants and updates. The parameters $maxProb/\theta$, β, and the retention factor were allowed to vary independently but the number of ants and updates were held such that the total number of ants tested across all generations was 5000. This constraint ensures that at most 1% of the exhaustive search space for the dataset was explored. For linear methods, $maxProb$ was allowed to vary from 0-100% on 10% intervals. For the exponential methods, θ was varied from .8 to 1. Outside of this interval, the probability (using the rank based method) that at least one of the attributes A_N or $A_N - 1$ is paired with any given ant is $> 1/2$. The retention constant was tested at 0.1, 0.5 and 0.9, and the number of ants/updates was tested at 10/500, 100/50, 250/25, and 500/10.

This analysis was performed on previously used datasets [8]. These contained two relevant epistatic SNPs out of 1000. We focused on the subset of these where sample size was 1600 and heritability was 0.1. At this heritability, 10% of the variability in phenotype is due to genotype. We tested five models using each of the expert knowledge scaling methods across all parameters. Each parameter set was run on 100 independent datasets for each model and the power of each parameter set was determined by the proportion of successful trials. We divided our results into four subsets, one corresponding to each expert knowledge scaling method, and performed logistic regression. The effects on power for all single parameters and all two-way interactions between the parameters was determined. For the logistic regression, we used the Design package [10] of the R programming language [11]. Parameters whose effects on power had p-values $\leq$ 0.05 were considered to be statistically significant. Using these criteria, we would expect significant p-values to occur only one in twenty times by chance alone. Optimal settings for significant parameters were determined in order of the magnitudes of their coefficients found by logistic regression.

5 Results

The exponential-fitness method depended, with p-values ≤ 0.05, on the value of θ, the number of ants/updates, and an interaction effect between these two. The parameters β and retention were not shown to be significant, indicating that the update rule for ant pheromone deposition is less influential on the success of the algorithm for this scaling method. The selection probabilities for exponential-fitness scaling are the most extreme of any method we tested, meaning that they assign a greater proportion of the selection probability to the SNPs with higher initial expert knowledge scores. All parameters, with the exception of retention and interaction effects which included retention, were found to have significant effects on the power of the exponential-rank method. The strongest effects were observed from θ, β and an interaction effect between these two. The power of the linear methods was significantly dependent on all parameters and in both cases the strongest effects were observed to be the pheromone retention parameter, β and the interaction between these two parameters.

Using the information from the logistic regression analysis, we propose a set of parameters that maximizes power for each of the expert knowledge scaling methods. We find that for the exponential-fitness method $\theta = 0.99$ and ants = 500 yielded the highest power [8]. As previously stated, β and retention were not

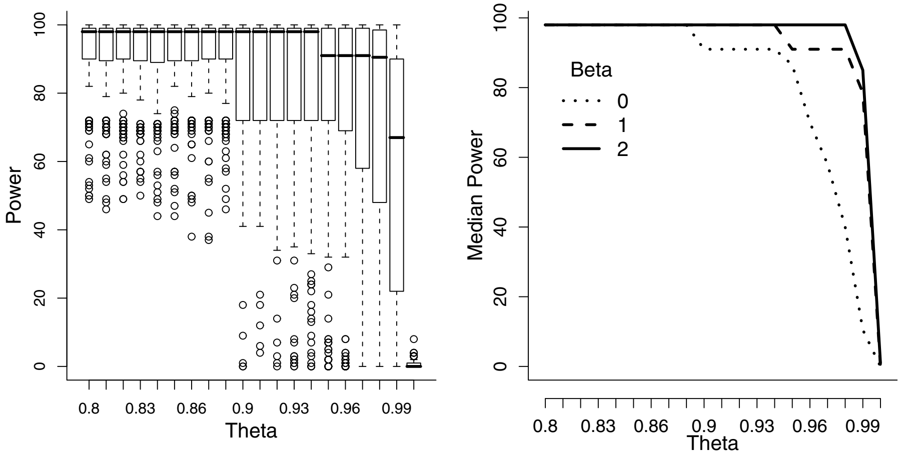
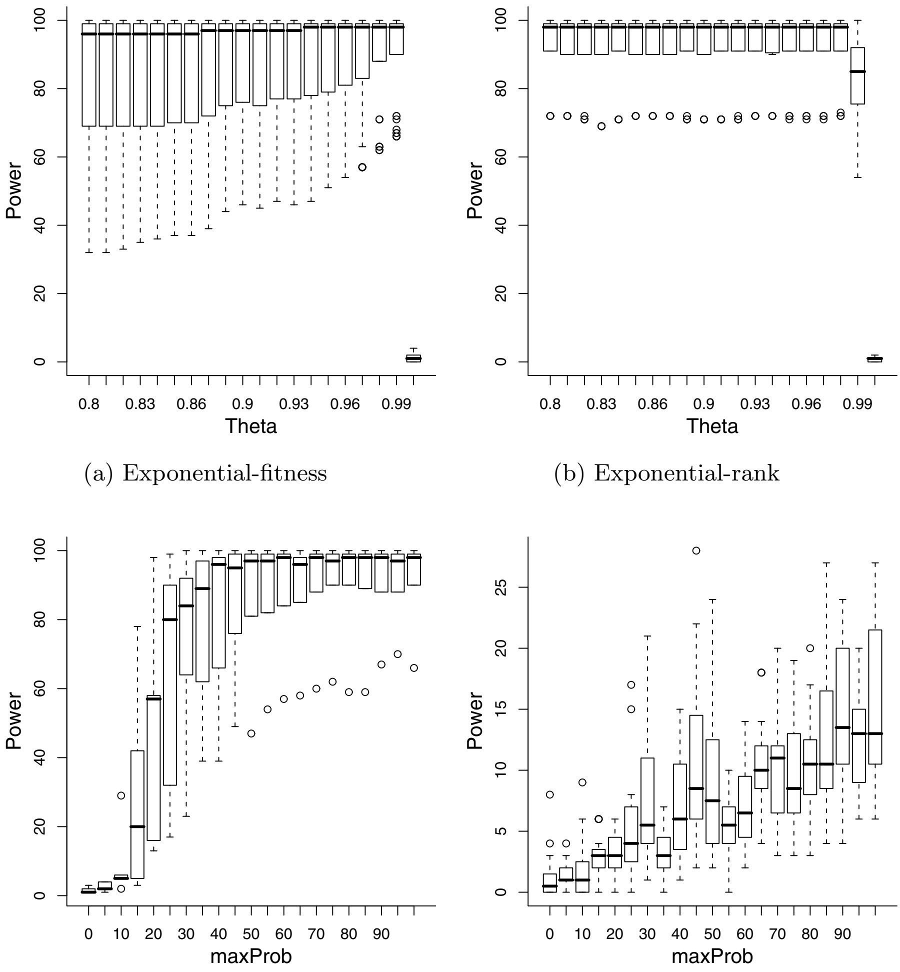


Fig. 1. These plots show the effects of θ and β on power for the exponential-rank method. θ values between 0.8 and 0.9 maximize the power of the algorithm and $\beta = 2$ is the most robust parameter setting across θ values.



(a) Exponential-fitness

(b) Exponential-rank

(c) Linear-fitness

(d) Linear-rank

Fig. 2. The power of each scaling method is shown with their respective optimized parameters. All scaling methods, with the exception of linear-rank, perform well on our simulated data under optimal parameter settings.

shown to be significant for this scaling method. For the exponential-rank method, we find the highest power when $\beta = 2$ and ants/updates = 500. The retention parameter was not shown to be significant and therefore the update rule for ants must depend more on the effects of β as it applies to the original scaling of the expert knowledge [8]. The exponential scaling methods are both able to achieve high power, $> 90\%$, using optimal parameter settings, though we find that the exponential-fitness method is more robust to changes in parameters other than θ. This is because the success of the exponential-fitness method depends highly on the initial scaling of the expert knowledge. Therefore, in cases where expert knowledge is less reliable, the exponential-rank method will likely outperform the exponential-fitness method.

The power of the linear-fitness method, using optimal parameters, was comparable to the powers of the exponential methods. However, the linear-fitness method was influenced more by retention and β than by its scaling parameter, $maxProb$. This indicates that the linear-fitness method may be less dependent on expert knowledge values than the exponential methods. The optimal parameter settings for the linear-fitness method were retention = 0.9, $\beta = 1$, and ants = 500. The optimal parameters for the linear-rank method were the same for those as the linear-fitness method, except that for the linear-rank method the optimal number of ants was 250. The linear-rank method performed dramatically worse than any other method and we hypothesize that a uniform assignment across the linear distribution function does not differentiate strongly enough between expert knowledge scores. As a result, the algorithm rarely explores areas of the search space likely to contain the relevant SNPs. For all methods, except linear-rank, the power was higher for greater numbers of ants.

6 Discussion

High-throughput genotyping presents an informatics challenges to human geneticist who are searching for genetic markers associated with common disease. Exhaustive combinatorial analysis of these genome-wide datasets is not feasible. Biologically inspired methods such as ant colony optimization provide computationally efficient approaches to searching for epistatic pairs of genes which are predictive of disease. By including expert knowledge information into our ant system we have shown that such a technique can be effective. Using the parameter set derived from our logistic regression analysis it is feasible to apply powerful methods such as MDR to the analysis of epistasis in large-scale high-throughput datasets. Biologically inspired computing are powerful software tools for the discovery of genetic interactions and genetic associations with disease, with the potential for widespread use in the field of human genetics.

7 Implementation and Method Availability

Every method we use here, including the machine learning algorithm, TuRF, which we use as expert knowledge, is available in the open source MDR software package from a user friendly GUI or command line interface. The ant system was implemented in the Java programming language and is distributed as a stochastic search method in the MDR software package [4]. Java and MDR are both open-source and freely available at sourceforge.net. The TuRF algorithm is available in the MDR framework as a dataset filtering method. Finally, the logistic regression analysis was performed using the Design package of the R statistical programming language. R [11] and the Design package [10] are both open source and freely available at cran.org. Powerful and easy to use implementations of biologically inspired computing methods provide effective tools for researchers interested in disease susceptibility and the elucidation of epistatic genes.

Acknowledgement

This work was supported by NIH grants LM009012, AI59694, and ES007373.

References

1. Bateson, W.: Facts limiting the theory of heredity. Science 26(672), 649–660 (1907)
2. Jacquard, A.: Heritability: One word, three concepts. Biometrics, 465–477 (1983)
3. Greene, C.S., White, B.C., Moore, J.H.: Ant colony optimization for genome-wide genetic analysis. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217, pp. 37–47. Springer, Heidelberg (2008)
4. Moore, J.H., Gilbert, J.C., Tsai, C.T., Chiang, F.T., Holden, T., Barney, N., White, B.C.: A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. Journal of Theoretical Biology 241(2), 252–261 (2006)
5. Cho, Y.M., Ritchie, M.D., Moore, J.H., Park, J.Y., Lee, K.U., Shin, H.D., Lee, H.K., Park, K.S.: Multifactor-dimensionality reduction shows a two-locus interaction associated with type 2 diabetes mellitus. Diabetologia 47, 549–554 (2004)
6. Brassat, D., Motsinger, A., Caillier, S., Erlich, H., Walker, K., Steiner, L., Cree, B., Barcellos, L., Pericak-Vance, M., Schmidt, S., et al.: Multifactor dimensionality reduction reveals gene–gene interactions associated with multiple sclerosis susceptibility in African Americans. Genes and Immunity 7(4), 310–315 (2006)
7. Manuguerra, M., Matullo, G., Veglia, F., Autrup, H., Dunning, A., Garte, S., Gormally, E., Malaveille, C., Guarrera, S., Polidoro, S., et al.: Multi-factor dimensionality reduction applied to a large prospective investigation on gene-gene and gene-environment interactions. Carcinogenesis 28(2), 414 (2007)
8. Greene, C.S., Gilmore, J.M., Kiralis, J., Andrews, P.C., Moore, J.H.: Optimal use of expert knowledge in ant colony optimization for the analysis of epistasis in human disease. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) EvoBIO 2009. LNCS, vol. 5483, pp. 92–103. Springer, Heidelberg (2009)
9. Moore, J.H., White, B.C.: Tuning reliefF for genome-wide genetic analysis. In: Marchiori, E., Moore, J.H., Rajapakse, J.C. (eds.) EvoBIO 2007. LNCS, vol. 4447, pp. 166–175. Springer, Heidelberg (2007)
10. Harrell Jr, F.E.: Design: Design Package (2007)
11. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008)

Impoverished Empowerment: 'Meaningful' Action Sequence Generation through Bandwidth Limitation

Tom Anthony, Daniel Polani, and Chrystopher L. Nehaniv

Adaptive Systems Research Group, University of Hertfordshire, UK

Abstract. *Empowerment* is a promising concept to begin explaining how some biological organisms may assign *a priori* value expectations to states in taskless scenarios. Standard empowerment samples the full richness of an environment and assumes it can be fully explored. This may be too aggressive an assumption; here we explore impoverished versions achieved by limiting the bandwidth of the empowerment generating action sequences. It turns out that limited richness of actions concentrate on the "most important" ones with the additional benefit that the empowerment horizon can be extended drastically into the future. This indicates a path towards and intrinsic preselection for preferred behaviour sequences and helps to suggest more biologically plausible approaches.

1 Introduction

Methods to provide an agent embodied in an environment with strategies to behave intelligently when given no specific goals or tasks are of great interest in Artificial Life. However, to do this embodied agents require some method by which they can differentiate available actions and states in order to decide on how to proceed. In the absence of no specific tasks or goals it can be difficult to decide what is and is not important to an agent.

One set of approaches examines processing and optimising the Shannon information an agent receives from its environment (Atick, 1992; Attneave, 1954; Barlow, 1959, 2001), following the hypothesis that embodied agents benefit from an adaptive and evolutionary advantage by informationally optimising their sensory and neural configurations for their environment.

Information-based predictions could provide organisms/agents with intrinsic motivation based on *predictive information*(Ay et al., 2008; Bialek et al., 2001; Prokopenko et al., 2006). In this paper we will concentrate on *empowerment* (Klyubin et al., 2005a,b), an information theoretic measure for the external efficiency of a *perception-action loop*.

One shortcoming of empowerment is that whilst it provides behaviours and results which seem to align it with processes that may have resulted from evolution the algorithms used to calculated it tend not to operate using an equally plausible process. It implicitly requires a notion of the richness and full size of the space it searches whatever algorithm is used to determine it. In this paper we thus introduce the assumption of a limit on the richness of the action repertoire.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 294–301, 2011.

1.1 Information Theory

First we give a very brief introduction to information theory, introduced by Shannon (1948). The first measure is *entropy*, a measure of uncertainty given by $H(X) = -\sum_x p(x)\log p(x)$ where X is a discrete random variable with values x from a finite set $\mathcal{X}$ and $p(x)$ is the probability that X has the value x. We use base 2 logarithm and measure entropy in *bits*.

If Y is another random variable jointly distributed with X the *conditional entropy* is $H(Y|X) = -\sum_x p(x)\sum_y p(y|x)\log p(y|x)$. This measures the remaining uncertainty about the value of Y if we know the value of X. Finally, this also allows us to measure the *mutual information* between two random variables: $I(X;Y) = H(Y) - H(Y|X)$.

Mutual information can be thought of as the reduction in uncertainty about the variable X or Y, given that we know the value of the other.

1.2 Empowerment

Essentially empowerment measures the channel capacity for the external component of a perception-action loop to identify states that are advantageous for an agent embodied within an environment. It assumes that situations with a high efficiency of the perception-action loop should be favoured by an agent. Based entirely on the sensors and actuators of an agent, empowerment intrinsically encapsulates an evolutionary perspective; namely that evolution has selected which sensors and actuators a successful agent should have, which in turn implies which states are most advantageous for the agent to visit.

Empowerment is based on the information theoretic perception-action loop formalism introduced by Klyubin et al. (2004, 2005a,b), as a way to model embodied agents and their environments. The model views the world as a communication channel; when the agent performs an action, it is injecting Shannon information into the environment, which may or may not be modified, and subsequently the agent re-acquires part of this information from the environment via its sensors.

In Fig. 1 we can see the perception-action loop represented by a Bayesian network, where the random variable R_t represents the state of the environment, S_t the state of the sensors, and A_t the actuation selected by the agent at time t. It can be seen that R_{t+1} depends only on the state of the environment at time t, and the action just carried out by the agent.

Empowerment measures the maximum *potential* information flow, this can be modelled by the channel capacity (Shannon, 1948) for a discrete memoryless channel: $C(p(s|a)) = \max_{p(a)} I(A;S)$.

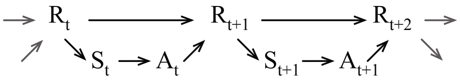


Fig. 1. Bayesian network representation of the perception-action loop

The random variable A represents the distribution of messages being sent over the channel, and S the distribution of received signals. The channel capacity is measured as the maximum mutual information taken over all possible input distributions, $p(a)$, and depends only on $p(s|a)$, which is fixed. One algorithm to find this maximum is the iterative Blahut-Arimoto algorithm (Blahut, 1972).

Empowerment can be intuitively thought of as a measure of how many observable modifications an embodied agent can make to his environment, either immediately, or in the case of n-step empowerment, over a given period of time.

In the case of n-step empowerment, we first construct a compound random variable of the last n actuations, labelled A_t^n. We now need to maximise the mutual information between this variable and the sensor readings at time $t+n$, represented by S_{t+n}. Here we consider empowerment as the channel capacity between these: $\mathfrak{E} = C(p(s_{t+n}|a_t^n)) = \max_{p(a_t^n)} I(A_t^n; S_{t+n})$.

An agent that maximises its empowerment will position itself in the environment in a way as to maximise its options for influencing the environment (Klyubin et al., 2005a).

2 Empowerment with Limited Action Bandwidth

2.1 Goal

We wanted to introduce a bandwidth constraint into empowerment, specifically n-step empowerment where an agent must look ahead at possible outcomes for sequences of actions, and even with a small set of actions these sequences can become very numerous.

An agent's empowerment is bounded by that agent's memory; empowerment measures the agent's ability to exert influence over it's environment and an agent that can perform only 4 distinct actions can have no more than 2 bits of empowerment per step. However, there are two factors which normally prevent empowerment from reaching this bound:

- Noise - A noisy / non-deterministic / stochastic environment means that from a given state an action has a stochastic mapping to the next state. This reduces an agent's control and thus its empowerment.
- Redundancy - Often there are multiple action (or sequences) available which map from a given state to the same resultant state. This is especially true when considering multi-step empowerment: e.g Moving North then West, or moving West then North.

Due to redundancy there are many cases where bandwidth for action sequences can be reduced with little or no impact on achievable information flow. Beyond this there may be scenarios with a favourable trade off between a large reduction in action bandwidth only resulting in a small reduction in empowerment (or utility).

2.2 Scenario

To run tests we constructed a simple scenario; an embodied agent is situated within a 2-dimensional infinite gridworld and has 4 possible actions in any single time step. The actions the agent can execute are North, South, East and West each moving the agent one space into the corresponding cell, provided it is not occupied by a wall. In the scenario the state of the world is solely the position of the agent, which is all that is detected by the agent's sensors.

2.3 Algorithm

The agent to examines all possible sequences for n-step empowerment for small values of n (typically $n < 6$) and then selects a subgroup of the available sequences to be retained.

To do this we use the information bottleneck method (Tishby et al., 1999). Having calculated the empowement we have two distributions: $p(a_t^n)$ is the capacity achieving distribution of action sequences and $p(s_{t+n}|a_t^n)$ is the channel that represents the results of an agent's iteractions with the environment.

We now look for a new "compact" distribution $p(g|a_t^n)$, where g are groups of 'alike' action sequences with $g \in G$ where $|G| \leq |A_t^n|$ and the cardinality of G corresponds to our desired bandwidth limit. A colloquial, though not entirely accurate, way to think of this is as grouping together action sequences that have similar outcomes (or represent similar 'strategies'). The information bottleneck works by first choosing a cardinality for G and then maximising $I(G; S_{t+n})$ (the empowerment of the reduced action set) using S_{t+n} as a relevance variable.

This results in a conditional distribution $p(g|a_t^n)$, from which we must derive a new distribution of our action sequences (with an entropy within the specified bandwidth limit). In order to end up with a subset of our original action sequences to form this new action policy for the agent, we must use an algorithm to 'decompose' the conditional distribution into a new distribution $p(\hat{a}_t^n)$ which has an entropy within the specified bandwidth limit (and usually contains only a subset of the original action sequences).

In the spirit of empowerment, for each g we want to select the action sequences which are most likely to map to that g (i.e the highest value of $p(g|a_t^n)$ for the given g) and provide the most towards our empowerment (i.e the highest value of $I(a_t^n; S_{t+n})$). This results in collapsing strategies to their dominant action sequence and maximises an agent's ability to select between strategies.

2.4 Results

Fig. 2 shows three typical outcomes of this algorithm; in this example we have a bandwidth constraint of 2 bits, operating on sequences of 6 actions. The walls are represented by patterned grey, the starting position of the agent is the light center square, and the selected trajectories by the dark lines with a black cell marking the end location of the sequence. The result that emerges is of interest;

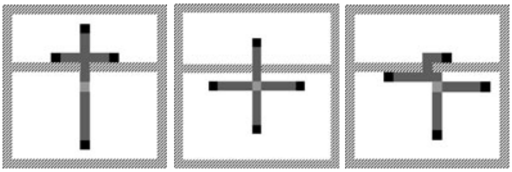


Fig. 2. Typical behaviours where 4 action sequences were selected from 4^6 possibilities

the sequences chosen can immediately be seen to be non-trivial and a brief examination reveals that the end points of each sequence each have only a single sequence (of the available 4,096) that reaches them.

In section 2.1 we discussed redundancy as one factor which should be eliminated first in order to maintain empowerment whilst reducing bandwidth. If we extrapolate this process of eliminating trajectories to 'easier to reach' states then it follows that, exactly as in Fig. 2, the last states the agent will retain are the entirely unique states that have only a single sequence that reaches them.

It appears that choosing to retain a limited number of explored sequences and this tendency for the agent to value 'unique' sequences indicates a first step towards a solution for extending the sequences beyond what was computationally possible before and may point to a plausible process for a biological organism to undertake. We discuss this in section 3.

2.5 Noise Induced Behaviour Modifications

Figures 3 A & B, a 4-step scenario with a bandwidth constraint of 2 bits corresponding to 4 action sequences, show there is not always a neat division of the world into what we would probably recognise as the 4 main 'strategies' (one trajectory into each of the 4 rooms). However, there is no pressure for the agent to do this or to consider the geographical distinctions between states, only for it to select unique end points.

However, with the introduction of noise this changes. Figures 3 C & D show two more randomly selected behaviours from the same scenario but with the introduction of noise, where each action in the sequence has a 5% probability of

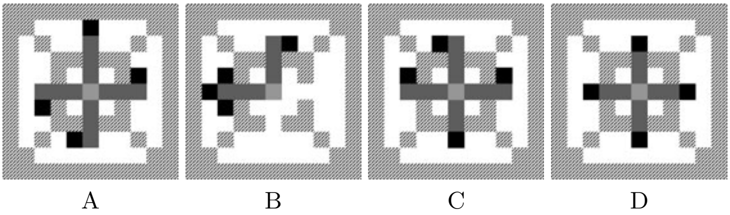


Fig. 3. Randomly selected behaviours; 4 steps with a 2 bit bandwidth constraint. A & B have no noise, C & D have 5% noise per step.

being replaced with a random action. In order to maintain as much empowerment as possible, the agent must ensure that in attempting one strategy it does not accidentally employ another, and in this environment that translates to being 'blown off course' and adds a drive for a geographical distinction between end states.

Note that some of the sequences appear to be only 3 steps long. This is a strategy employed by the agent, and what is actually happening is the agent uses an action to push against the wall while passing through the doorway, possibly as a way to minimise the effect of noise.

3 Building Long Action Sequences

The current formulation for n-step empowerment utilises an exhaustive search of the action space for $n - steps$. It can be seen that this is a highly unlikely approach for biological organisms to employ, especially for large values of n and in rich environments.

We hypothesise, given that for short sequences of actions it is manageable to cheaply examine all sequences, that we could approach an agent's bandwidth divided into two parts. In section 2.3 we evaluated all possible short sequences in a 'working' memory, then retained only a subset for the agent's 'long term' memory according to our bandwidth constraint.

Following the result above from bandwidth-limited empowerment it became apparent that retaining only a small subset of investigated action sequences lends itself to the idea of then searching further from the final states of such sequences.

This is obvious when applied to the cases where the bandwidth has been constrained just enough to retain empowerment but eliminate all redundancy. It is essentially realising the Markovian nature of such sequence based exploration: when arriving at a state to explore, how you arrived is not of consequence to further exploration. The results, however, seem to suggest that even *beyond* this point of retained empowerment, where the bandwidth severely limits the achievable empowerment and selection of sequences, the iterative approach still produces noteworthy behaviours.

The approach was to set a target length for a sequence, for example 15-step empowerment, then the problem is broken down in to i iterations of n-step empowerment where $n \cdot i = 15$. Standard n-step empowerment is performed, and then the above presented bandwidth-reduction algorithm is run to reduced the action set to a small subset. Each of these action sequences is then extended with n additional steps. These are then again passed through the bandwidth-reduction algorithm and this repeated a total of i times. If we select $n = 5$, $i = 3$ and a bandwidth limit of 4 bits (16 action sequences) then the total sequences evaluated in our gridworld scenario is reduced from 4^{15} to $33,792$, which is a search space more than $3 \cdot 10^4$ times smaller.

Figure 4 shows the results of such a scenario with the selected action sequences and there are several important aspects to note. Firstly, the agent continues to

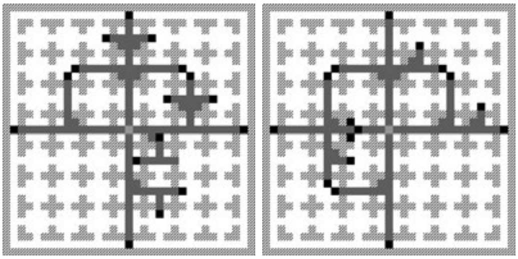


Fig. 4. Iteratively built sequences of 15 steps, with a bandwidth constraint of 4 bits

reach certain states that are of obvious consequence, most notably the 4 cardinal directions, but also over half of the furthest reachable corner points. Furthermore the pattern of trajectories has a somehwat 'fractal' nature and appears to divide the search space up systematically. These results are of interest because these states and behaviours are far beyond the horizon of a single iteration of standard n-step empowerment. Space does not permit us to give details but initial results also indicate that interesting locations of the environment, such as door and bridges, are also handled by such iterative sequence building.

4 Discussion

We have identified several challenges to the recently introduced concept of empowerment which endows an agent's environmental niche with a concept distinguishing desirable from less desirable states. Empowerment essentially measures the range in environmental change imprinted by possible action sequences whose number grows exponentially with the length of the sequence. It is virtually impossible to compute it algorithmically for longer sequences, and, likewise, it is implausible that any adaptive or evolutionary natural process would be able to indirectly map this whole range.

Therefore, here we have, consistently with the information-theoretic spirit of our study, applied informational limits on the richness of the action sequences that generate the empowerment. In doing so, we found that: 1. the information bottleneck reduces redundant sequences; 2. in conjunction with the complexity reduction through the collapse of action sequences, particularly "meaningful" action sequences that explore important features of the environment, e.g. principal directions, doors and bridges, are retained, and finally, that *significantly* longer action sequences than before can be feasibly handled. This in itself already suggests insights for understanding the possible emergence of useful long-term behavioural patterns. Note that in this study we have relinquished the computation of empowerment as measure for the desirability of states in favour of filtering out desirable action patterns.

References

Atick, J.J.: Could information theory provide an ecological theory of sensory processing. Network: Computation in Neural Systems 3(2), 213–251 (1992)

Attneave, F.: Some informational aspects of visual perception. Psychological Review 61(3), 183–193 (1954)

Ay, N., Bertschinger, N., Der, R., Guettler, F., Olbrich, E.: Predictive information and explorative behavior of autonomous robots. European Physical Journal B (2008) (accepted)

Barlow, H.B.: Possible principles underlying the transformations of sensory messages. In: Rosenblith, W.A. (ed.) Sensory Communication: Contributions to the Symposium on Principles of Sensory Communication, pp. 217–234. The M.I.T. Press, Cambridge (1959)

Barlow, H.B.: Redundancy reduction revisited. Network: Computation in Neural Systems 12(3), 241–253 (2001)

Bialek, W., Nemenman, I., Tishby, N.: Predictability, complexity, and learning. Neural Comp. 13(11), 2409–2463 (2001)

Blahut, R.: Computation of channel capacity and rate distortion functions. IEEE Transactions on Information Theory 18(4), 460–473 (1972)

Klyubin, A.S., Polani, D., Nehaniv, C.L.: Organization of the information flow in the perception-action loop of evolved agents. In: Zebulum, R.S., Gwaltney, D., Hornby, G., Keymeulen, D., Lohn, J., Stoica, A. (eds.) Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware, pp. 177–180. IEEE Computer Society Press, Los Alamitos (2004)

Klyubin, A.S., Polani, D., Nehaniv, C.L.: All else being equal be empowered. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 744–753. Springer, Heidelberg (2005a)

Klyubin, A.S., Polani, D., Nehaniv, C.L.: Empowerment: A universal agent-centric measure of control. In: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, vol. 1, pp. 128–135. IEEE Press, Los Alamitos (2005b)

Prokopenko, M., Gerasimov, V., Tanev, I.: Evolving spatiotemporal coordination in a modular robotic system. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J., Marocco, D., Meyer, J.-A., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 558–569. Springer, Heidelberg (2006)

Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423 (1948)

Tishby, N., Pereira, F., Bialek, W.: The information bottleneck method. In: Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing, pp. 368–377 (1999)

Influence of Promoter Length on Network Convergence in GRN-Based Evolutionary Algorithms

Paul Tonelli, Jean-Baptiste Mouret, and Stéphane Doncieux

Institut des Systèmes Intelligents et Robotiques UPMC-Paris 6,
CNRS UMR 7222 4 place Jussieu, 75252 Paris Cedex

Abstract. Genetic Regulation Networks (GRNs) are a model of the mechanisms by which a cell regulates the expression of its different genes depending on its state and the surrounding environment. These mechanisms are thought to greatly improve the capacity of the evolutionary process through the regulation loop they create.

Some Evolutionary Algorithms have been designed to offer improved performance by taking advantage of the GRN mechanisms. A recent hypothesis suggests a correlation between the length of promoters for a gene and the complexity of its activation behavior in a given genome. This hypothesis is used to identify the links in in-vivo GRNs in a recent paper and is also interesting for evolutionary algorithms. In this work, we first confirm the correlation between the length of a promoter (binding site) and the complexity of the interactions involved on a simplified model. We then show that an operator modifying the length of the promoter during evolution is useful to converge on complex specific network topologies. We used the Analog Genetic Encoding (AGE) model in order to test our hypothesis.

1 Introduction

Evolutionary Algorithms (EA) are nowadays able to offer improved solutions for many problems and sometimes outperform engineering methods. Though, we are unable to obtain solutions as complex as what in-vivo evolution has produced. Some of the evolutionary mechanisms behind biological evolution are still unexplained. It is believed EA could benefit from understanding these mechanisms [1]. There are many of these mechanisms, as for example splicing, through which a single gene can encode multiple proteins [2, 3]. Genetic Regulation Networks (GRNs) are a model of other of theses mechanisms by which a cell regulates the expression of its different genes depending on its state[4]. A strong hypothesis is that the complexity of the interactions obtained thanks to the GRNs is, at least, partly responsible for the diversity increase of many living organisms [5], as well as improvements in the genome evolvability and robustness [6]. Though, the complexity of these mechanisms is still not fully understood [7].

We believe that understanding which in-vivo characteristics improve the performance of biological evolution is a key to designing efficient EA. In order to do

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 302–309, 2011.

so, it is necessary to identify which elements of these mechanisms are necessary and must be reproduced to improve performance. In this work, we investigated if one of these mechanisms is relevant: a mutation operator adding or removing one base in string based evolutionary algorithms. This operator is usually considered minor, as in-vivo RNA is read three bases at a time (codon) during protein synthesis, and adding or deleting a base in DNA shifts the sequence, creating a protein totally different from the original. Though, in the case of non-coding DNA, this operator may be more useful. In the following work, we asked ourselves two questions. First, does the add/remove operator provide a sufficient mechanism to obtain a correlation between the length of cis-regulatory sequences as stated in [8]. And more important, does this operator improve the performance of the algorithm in any way ?

2 Related Works

2.1 Gene Regulation Networks

GRNs rely on multiple mechanisms. One of them is the possibility for a protein, called a transcription factor to bind itself to a sequence of DNA located before or after a given gene. When a relevant protein binds itself to the site, the expression of the gene will either be enhanced (enhancer) or blocked (inhibitor). Most genes present such cis-regulatory sequences, which contain multiple binding sites for various proteins. These mechanisms can first be seen as a mean to create "programs", enhancing the cell capacity to order the synthesis of proteins or reactions to adapt to specific stimuli [9].

GRNs are also believed to speed up evolution. A single mutation in a cis-regulatory region can have varying impact on an organism without modifying the gene itself, for example by removing or creating a new interaction between the regulated gene cis-regulatory sequence and another transcription factor. Duplication of a transcription factor gene or binding site also creates new interactions in a genome [10]. Therefore interactions provided by the GRNs provide another level impacted by evolution.

2.2 Existing Methods

The objective to find ways to harness the properties of these GRNs to improve the performance of evolutionary algorithms is stated in [1]. More precisely, the goal is to understand which GRN properties improve the evolvability of living organisms. Some properties have already been highlighted in several articles [11–13]. Algorithms have been created to take advantage of them as Artificial Ontogeny [14] or lately PBGA [15]. We tried to find a model closely related to the biological mechanisms while avoiding the overhead of more complex biology based models like HeRoN [16].

2.3 Research of Relevant Properties of GRNs

The first step to take advantage of the GRN capabilities is to understand their properties and the implications for evolvability. Examples of these properties are found in [11] which tries to understand how varying goals coupled to specific evolution mechanisms can change the evolvability of a genome to speed up convergence on specific problems. It highlights the fact that the specificity of transcription factors to multiple binding sites is, in itself, a way to encode evolutionary information. The mechanisms behind GRNs are quite complex and multiple parameters are still unknown. Here, we restrict our study to algorithms using both a string based encoding while keeping simple enough matching mechanisms. AGE is the only existing example we could find to fulfill these conditions. A work similar to this one was done by C. Mattiussi [17], who studied the impact of a mutation operator allowing the duplication of sequences in a genome to obtain convergence of the algorithm. Here, we do a similar work with the possibility to incrementally modify the length of the cis-regulatory sequences.

2.4 AGE

Analog Genetic Encoding is an indirect encoding mechanism which was designed to use some mechanisms of the GRNs [17] to generate networks by evolving a string based EA. It has recently shown impressive results for the reverse engineering of existing in-vivo GRNs[18]. AGE features complex generation mechanisms, many of which are not relevant to our problem, therefore, we chose to focus only on part of these mechanisms which will be described here. Our goal, as in [11] is to assess the capacity of a set of nodes to converge to a specific network topology. In AGE, a network is composed by a set of genes. Each gene is equivalent to a node in a network (see figure 1). Each gene is composed by output and input sequences, each of which is located in the gene and located between a start and an end sequence (which could be compared to start and stop codons). The links in the network are defined by comparing the input and output sequences of different nodes. Similar sequences will be considered as a strong link between two nodes while two totally different sequences will be considered as an absence of link. As expressed in [17], this is quite similar to the process by which micro RNA can repress the expression of other genes by blocking their DNA or RNA expression [19]. For this model, the strength of the link is computed using a local alignment score [20].

3 Experiments

For our experiment, we considered a fixed set of nodes (three to five, depending on the experiment). Each node is composed of two sequences of nucleotides. One input sequence which models the promoter sequence, and one output sequence, loosely modelling the transcription factor / microRNA. Our algorithm mutates theses two sequences by using three possible operators. The first (second) one is

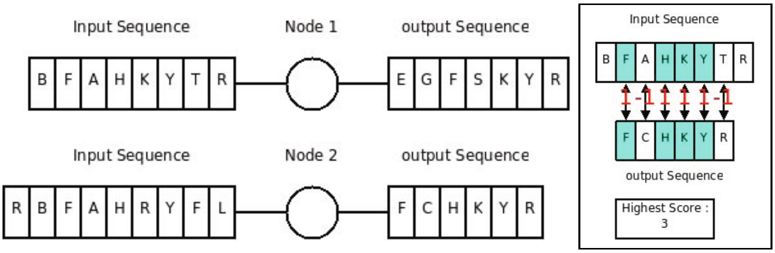


Fig. 1. Left: Two sample nodes in our simplified implementation of AGE; Right: the sequence comparison algorithm for definition of the link between the output sequence of the bottom node and the input sequence of the top node. Only contiguous bases are considered, the longest subsequence for this comparison is therefore of 3 bases even if both also have the F in common.

the addition (deletion) of a nucleotide at some point in the sequence. Another point is that we do not extend our algorithm to differentiating inhibitory and enhancing links.

The fitness of a network is obtained by comparing the network topology to a reference network. An optimal link strength (1.0) is defined by two sequence containing a common subsequence of three bases. The absence of link is defined as two sequences with a longest common subsequence between two sequences of one base. The intermediary is considered as a link with a strength of 0.5. We made two different experiments related to the length of the sequence of the binding sites. The first one is used to test the correlation between the length of a sequence and the number of links this sequence has in the network. The second experiment tests the effect of switching on and off the add/remove operator on the convergence speed for a "complex" network. All the runs were done using the simplified version of AGE described previously where the only additional mutation operators possible are the exchange of a base in the sequence for another and the addition / deletion operators. In order to improve the performance, the size of the alphabet was set at 7 (this gives the best results for our networks). There is no order relationship between the bases in the alphabet (the replacing mutation operator switches randomly from one base to the other). The selection algorithm used for all the experiments was NSGA 2 [21], a commonly used tournament based multi objective selection algorithm as further experiments required multiple objectives. Each run was repeated at least 10 times. Figure 2 sums up all the parameters used for the experiments.

3.1 Convergence of Sequence Length

The first experiment was done in two steps. In the first step, we tried to evolve two simple networks of 3 nodes with homogeneous (2 output links and 2 input links per node) or heterogeneous links (2 outputs for each node but 1 to 3 inputs per node) as shown in figure 3 and we analysed the length of all the sequences of the first individual to reach the optimal fitness in each run.

Parameters			
alphabet size	7	population	200
maximum number of generation	10000	probability to delete base	0.01
probability to add base	0.02	probability to mutate base	0.1
max sequence length	20		

Fig. 2. Summary of the parameters

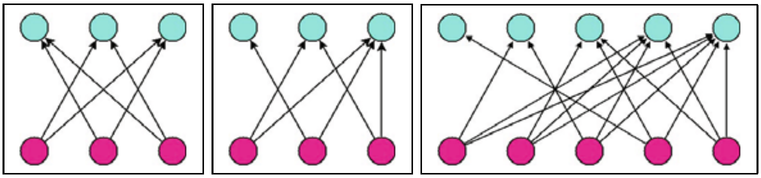


Fig. 3. Left: homogeneous network of three vertex (the top circles show the outputs, the bottom ones the inputs); Center: heterogeneous (different number of inputs and same numbrer of outputs) network with 3 nodes; Right: heterogeneous network with 5 nodes

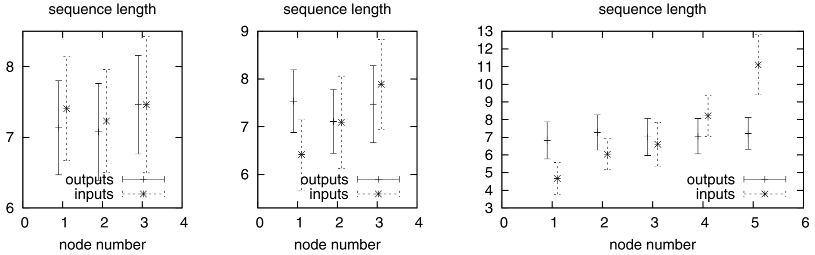


Fig. 4. Left: sequence length for an homogeneous network containing 3 nodes; Center: sequence length for an heterogeneous network containing 3 nodes; Right: sequence length for an heterogeneous network containing 5 nodes

The results of these experiments show a convergence to a size of 7 for each sequence in the homogeneous network. In the experiment trying to converge on a 3 node heterogeneous network (center box of figure 3),we obtained a correlation between the length of the sequence and the number of links connected to the node. As the differences were not significant, we made a similar experiment with a network containing 5 nodes and heterogeneous connections (nodes had 3 outputs each and respectively 1, 2, 3, 4 and 5 inputs). All the results (mean length of the sequences and standard deviation) are shown in figure 4.

For the results on the 5 nodes network, we have a significant difference (using Wilcoxon T-test) as the probability of the two sequences being from the same data is less than 1% between all the nodes having different numbers of inputs apart from between sequences 2 and 3 where this probability is 7.5%. These results confirm our first hypothesis, which is that the length of the sequences illustrates

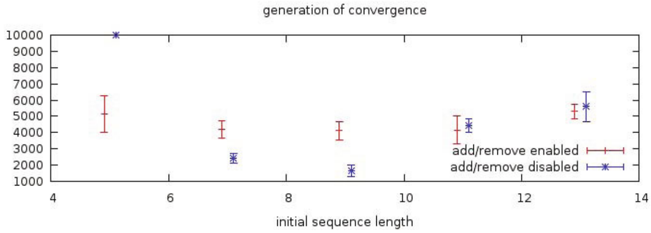


Fig. 5. Performance of the EA on a complex network for different bootstrap lengths

bootstrap length	5	6	7	8	9	10	11	12	13
add/remove enabled	1	1	1	1	1	1	1	1	0.8
add/remove disabled	0	1	1	1	1	1	1	1	0.4

Fig. 6. Convergence rate of experiments depending on bootstrap length and specified operators

the complexity of the interactions the node is involved in. The add/remove mutation operator is therefore sufficient to obtain these results. This is also a confirmation of the results stated in [8].

3.2 How the Operator Affects Performance of the EA

The first experiment showed that the length of the cis-regulatory sequence depends on the complexity of the interactions the node is involved in. Our next step was to test if a network correctly initialized could converge without length modification during the run or if the modification of the length improves performance. In evident cases (simple 3 nodes networks), the add/remove operator doesn't have a significant impact on the results unless initial parameters are initialized to abnormal values. Therefore, we experimented on a network which the algorithm has difficulties to converge to. In order to place ourselves in the worst possible scenario for our hypothesis (necessity of the add remove operator), we took an homogeneous symmetric target network where each node has 4 inputs and 4 outputs and tried to compare the performance. To do so, we first randomly initialized a 5 node network and made it converge to the target network with the addition / deletion operator enabled. It converged in all of the runs and showed a mean length of 8 bases per sequence and a standard deviation around 1. We then ran the same experiments with length modification operators disabled and a fixed initial length of 8 corresponding to the “optimal” length and compared it to a similar network without the length modification enabled.

The results were that the runs without sequence length modification were significantly faster than the runs with the operator enabled. However, this is a case where we specified the optimal length before running the algorithm, which is an unusual situation. Therefore, we made several additional runs of both algorithms by changing only the initial sequence length to compare their performance. The results are given in figure 5 and show that, if the initial length is not optimal, the add/remove operator is a good way to avoid seeing the algorithm get stuck because of insufficient initial complexity. It also helps the convergence rate in non optimal bootstrap cases as can be seen in figure 6. Therefore, we believe that, as the optimal situation of both an homogeneous network and predefined optimal sequence length is unusual, it is usually a good idea to enable the sequence length modification. The alternative being to define the optimal length by another mean before running the algorithm.

4 Conclusion and Discussion

In this work we have first confirmed that there is a correlation between cis-regulatory sequence length and the complexity of interactions the following gene is involved in, but also that the dynamic modification of sequence length is a useful operator (sometimes to allow convergence of the EA, or to facilitate convergence in complex situations). We have also illustrated that a simple evolution mechanism is able to take advantage of these operators, at least in certain cases.

The results shown in the first experiment converge around a length of 7 for both the 3 nodes network while the 5 nodes network converge on a length of 8. Therefore, it could be argued that the optimal length is usually around 8 and that, as this length is sufficient, the add/remove operator can be disabled to improve performance. This is partly true for these sandbox networks (in experiment 2, the best performance is achieved for a bootstrap length of 9). The goal of AGE and other EA is eventually to solve complex problems, with potentially many more nodes and links. In these situations, using a default a length of 8 is raises a risk that the network might not offer enough complexity to converge, as was the case with the 5 bases long runs of the third experiment and therefore be unable to converge.

References

1. Banzhaf, W., Beslon, G., Christensen, S., Foster, J.A., Képès, F., Lefort, V., Miller, J.F., Radman, M., Ramsden, J.J.: GuidelinesFrom artificial evolution to computational evolution: a research agenda. Nature Reviews Genetics 7(9), 729–735 (2006)
2. Tischer, E., Mitchell, R., Hartman, T., Silva, M., Gospodarowicz, D., Fiddes, J.C., Abraham, J.A.: The human gene for vascular endothelial growth factor. Multiple protein forms are encoded through alternative exon splicing. Journal of Biological Chemistry 266(18), 11947–11954 (1991)
3. Modrek, B., Lee, C.: A genomic view of alternative splicing. Nature Genetics 30(1), 13–19 (2002)
4. Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Joseph, B.Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., et al.: Transcriptional regulatory networks in Saccharomyces cerevisiae. Science 298(5594), 799–804 (2002)
5. Levine, M., Tjian, R.: Transcription regulation and animal diversity. Nature 424, 147–151 (2003)

6. Wagner, A.: Robustness and evolvability in living systems. Princeton University Press, Princeton (2005)
7. Wittkopp, P.J.: Evolution of cis-regulatory sequence and function in Diptera. Heredity 97(3), 139–147 (2006)
8. Kristiansson, E., Thorsen, M., Tamas, M.J., Nerman, O.: Evolutionary forces act on promoter length: identification of enriched cis-regulatory elements. Molecular Biology and Evolution (2009)
9. Alon, U.: An introduction to systems biology: design principles of biological circuits. Chapman & Hall/CRC (2007)
10. Teichmann, S.A., Babu, M.M.: Gene regulatory network growth by duplication. Nature Genetics 36(5), 492–496 (2004)
11. Izquierdo, E.J., Fernando, C.T.: The Evolution of Evolvability in Gene Transcription Networks. Artificial Life 11, 265 (2008)
12. Tanay, A., Regev, A., Shamir, R.: Conservation and evolvability in regulatory networks: The evolution of ribosomal regulation in yeast. Proceedings of the National Academy of Sciences 102(20), 7203–7208 (2005)
13. Chen, K., Rajewsky, N.: The evolution of gene regulation by transcription factors and microRNAs. Nature Reviews Genetics 8(2), 93–103 (2007)
14. Bongard, J.C.: Evolving modular genetic regulatory networks. In: Proceedings of The IEEE 2002 Congress on Evolutionary Computation (CEC 2002), pp. 1872–1877 (2002)
15. Bellas, F., Becerra, J.A., Duro, R.J.: Using promoters and functional introns in genetic algorithms for neuroevolutionary learning in non-stationary problems. Neurocomputing (2008)
16. Gonçalves, A., Costa, E.: A Computational Model of Gene Regulatory Networks and its Topological Properties. Artificial Life 11, 204 (2008)
17. Mattiussi, C.: Evolutionary synthesis of analog networks. PhD thesis, Università degli Studi di Trieste (2005)
18. Marbach, D., Mattiussi, C., Floreano, D.: Replaying the evolutionary tape: Biomimetic reverse engineering of gene networks. Annals of the New York Academy of Sciences (2008)
19. Ruvkun, G.: Molecular biology: glimpses of a tiny RNA world. Science's STKE 294(5543), 797 (2001)
20. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of Molecular Biology 147, 195–197 (1981)
21. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)

Cellular Automata Evolution of Leader Election

Peter Banda

Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics, Comenius University
Mlynska Dolina, 842 48 Bratislava, Slovakia
banda@fmph.uniba.sk

Abstract. The leader election problem is a crucial problem in the theory of distributed algorithms, multi-agent systems as well as in sociobiology. In this paper we investigate one-dimensional binary state cellular automata with an intention to track self-organizational mechanisms that finally enable a global leader to be elected. Since our model is anonymous and uniform we also have to deal with a problem of symmetry that in great majority of cases is broken by inhomogeneity of arbitrary initial configurations. Our approach to the problem is based on the evolution of cellular automata by genetic algorithms and the methodology of computational mechanics. The presented new solution of the leader election reaches remarkably high performance of $94 - 99\%$. The analysis shows a sophisticated collective computation demonstrated by so called particles and their interactions. Due to the simplicity of our model, presented approach is general and universal enough to be applicable even at the level of primitive biological or artificial societies.

1 Introduction

In the last decades scientists from natural as well as social science are increasingly facing problems related to the principal holistic concepts of complexity and self-organization [1]. It is often a surprising finding that dynamics of numerous structurally simple uniform systems, both natural and artificial might be considered as complex. Elementary distributed system of cellular automaton (CA) [2] introduced by the pioneer of computational age, von Neumann, has been extensively used since its very origin for studying various aspects of artificial life and dynamical systems. In this paper we focus on a well known problem of leader election [3], nontrivial for the anonymous and uniform architecture of CA. Our methodology includes evolution of CAs [4,5] employing genetic algorithms [6] and consecutively an analysis of CA dynamics by the framework of computational mechanics [7]. These approaches have been applied at the Santa Fe Institute with the motivation to understand natural systems and also to engineer decentralized artificial systems which can give rise to emergent computation. Emergent dynamics of best CAs we present are characterized by propagating space-time structures called particles. Our findings have important implications for the theory of distributed algorithms, sociobiology and development biology.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 310–317, 2011.

2 Leader Election Problem

Leader election has an important role for a global coordination, decision making and spatial orientation of social and biological systems. Animal groups have to collectively decide about communal movements, activities, nesting sites and cooperative hunting. Sociobiology distinguishes two decision making procedures - shared consensus, in which the group does what the majority votes for, and unshared consensus also referred to despotic decision, in which one individual (leader) makes the decision, which the rest of the group follows. For example, penguins *Columbia livia* decide mostly by shared consensus [8], whereas despotic decision of one individual can be found in the population of bottlenose dolphins [9]. Consensus decision in the group of *Macaca tonkeana* involves nearly all group members, on the other side just a few dominant and old individuals take a prominent role in *Macaca mulatta* [10]. In animal societies leader is usually elected or determined by attributes like age, knowledge and/or dominance, however variable leadership with no correlation to dominance was observed as well. Leader election is also implemented in biologically-inspired computational models of cell differentiation [11]. Within the homogenous regions, some cells have to be elected to take on special roles. Nice example of such process is the developing of a wing during morphogenesis of the fruit fly *Drosophila* [12]. Evolution of leader election in the system of self-replicating digital organisms (AVIDA) is described in [13]. The theory of distributed algorithms deals with the leader election (queen bee problem) [3] at more general level. Given a net of processors it is required to design the processors such that they are able to choose a leader (single processor) starting from an initial configuration, where all the processors are in the same state. It is an important prerequisite of other distributed algorithms as minimal clique, graph explore, broadcast etc. Angluin [14] showed that no deterministic algorithm can find a leader of a ring of anonymous processors due to the full symmetry of system. Nevertheless, various solutions considering less strict instance of the problem were proposed. Basically the most common approach presumes distinguishable unique processors where the search for a processor with particular minimal or maximal id is applied [15].

The leader election has its indisputable purpose, but what processes hidden behind the scene are responsible for that. Let us make fundamental abstraction and consider fully uniform society with anonymous agents without any memory. Would it be possible to elect leader also in such a case? In this work we will demonstrate that model required for leader election does not have to be complicated at all and no comparable attributes of individuals as size, age is needed. In particular, we present the minimalist biologically inspired distributed system of cellular automaton [2] with aspiration to explain or at least give some insights to leader election at the elementary level of cells, biological and artificial societies.

3 Cellular Automaton and Computational Mechanics

John von Neumman introduced the concept of a cellular automaton with motivation to explore logical requirements for machine self-replication and information processing in nature. CA consists of a lattice of components called cells with cycled boundaries (toroid topology). Let us denote the number of cells by letter N and the state set by Σ ($k = |\Sigma|$). The state of a cell with index i at a time t is labelled as $s_i^t \in \{0, \ldots, k-1\}$. The configuration is then sequence of cell states

$$\mathbf{s}^t = (s_0^t, s_1^t, \ldots, s_{N-1}^t) \tag{1}$$

In this paper we are focused exclusively on one-dimensional binary state CA. The neighborhood function $\eta : N \to \Sigma^n$ for one-dimensional CA is determined by radius r, thus the number of neighbors $n = 2r + 1$ and

$$\eta_i = (s_{i-r}, \ldots, s_i, \ldots, s_{i+r}) \tag{2}$$

The same transition rule $\phi : \Sigma^n \to \Sigma$ represented either by the transition table or finite state transducer, is applied synchronously to each cell resulting in an update of cell state $s_i^{t+1} = \phi(\eta_i^t)$ starting from some initial configuration (IC). By extending the scope of transition function, we can define the global transition rule $\Phi : \Sigma^N \to \Sigma^N$ operating on configurations and furthermore the ensemble operator $\mathbf{\Phi} : 2^N \to 2^N$ mapping a set of configurations. Dynamics of one-dimensional CA is often illustrated using a space-time diagram (Figure 1), where lattice of cells is displayed horizontally using black for active cell (state 1) and white for inactive cell (state 0). Time goes vertically from top to the bottom. CAs were successfully used in a variety of research fields and applications, such as artificial life, physical modelling, social and biological simulations etc.

Since CAs are completely discrete, it was quite difficult to analyze their behaviors with instruments known from the theory of conventional dynamical systems. This gap was bridged by the methodology of computational mechanics [7] using concepts from both, computation and dynamical system theories. The global, collective dynamics of CA can be therefore understood and described in terms of space-time structures - domains forming the regular background of computation, particles acting as carriers of information and particle interactions that are analogous to information processing.

Formally, a regular domain Λ^j is a process (regular) language consisting of a set of spatial configurations. This process language fulfils the properties of temporal invariance ($\mathbf{\Phi}^p(\Lambda^j) = \Lambda^j$) and spatial homogeneity - the graph of Λ^j process language is strongly connected. Domains can be discovered either by visual inspection of space-time diagrams or automatically via the ϵ-machine reconstruction and verifying given two conditions [16]. Domains might be filtered out from space-time diagram by so called domain filter [7] revealing certain space-time structures not belonging to domains. Part of these structures are regular, propagating objects that are known as particles. A particle usually marked by a letter of Greek alphabet is spatially localized and temporally periodic structure at the boundary of two domains with limited width (Fig. 1). Temporal periodicity of particle α is denoted as p_α and set of particles as $\mathbf{P} = \{\alpha, \beta, \ldots\}$. The displacement d_α of a particle α is defined as the number of cells, that particle is shifted during one period (the left displacement is negative, the right one is positive). Velocity is then calculated as $v_\alpha = d_\alpha / p_\alpha$. Different velocities of particles lead to particle interactions denoted $\alpha + \beta \to \gamma$ (Fig. 1). The result of interaction is determined by the actual phase of colliding particles and might lead to a production of new particle(s) or to an annihilation of colliding particles (result $\emptyset$). The set containing all possible interactions is denoted as **I**. Particle catalog consisting of domains, particles and particle interactions $\{\mathbf{\Lambda}, \mathbf{P}, \mathbf{I}\}$ gives us a strong descriptive tool to understand the processes underlying CA dynamics.

We use the model of CA ($r = 3$) that is anonymous, synchronous and deterministic with no information about its size. We would like to emphasize that it is a most difficult instance of leader election which is principally insolvable in its purist form. Basically only approach to break a symmetry in such a system is to presume self-stabilizing version in which ICs are not uniform, but arbitrary (random). All state of art models however require besides arbitrary ICs some additional prerequisites. Our CA operates in linear time ($2N$) and just constant (binary state) memory and is even to some minor limitations in terms of ICs, most basic system capable of leader election reaching performance of $94 - 99\%$.

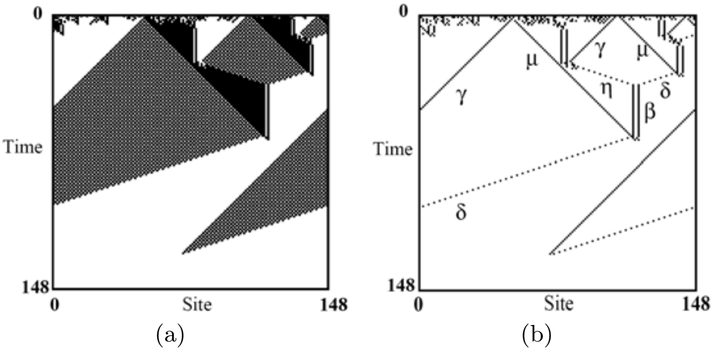


Fig. 1. (a) Space-time diagram and (b) filtered version with particle identification [5]

4 Evolution of Leader Election

Now we define leader election computational task reflecting the relationship between an input (IC) and desired output (final configuration(s)) as

$$T : \{0,1\}^N \to \{\mathbf{s} \in \{0,1\}^N |_{\#_1}\mathbf{s} = 1, T(\mathbf{s}) = \mathbf{s}\} \tag{3}$$

The function $\#_1$ denotes the number of 1s in a configuration. The goal is to transform each IC to some stable point configuration containing exactly one cell in state 1 and the rest in state 0. Due to the huge number of possible configurations ($|\Sigma|^N$) performance $P_N^I(\phi)$ of CA with transition rule ϕ and lattice size N considers not all, but just the reasonable number of randomly generated ICs I. Performance is then a fraction of the number of correctly computed ICs to the total number of test ICs I. Additionally, if CA does not reach a correct answer within the time T_{MAX}, the output is considered as incorrect. The CAs (transition rules) capable of leader election were found by employing well known genetic algorithms (GAs) [6] inspired by classical Darwin's theory of natural selection. We performed CA evolutions in a similar way as EvCA research group at SFI for CA problems of density classification and synchronization [4,5].

Chromosome is a binary vector of length 2^{2r+1} coding transition table outputs $\phi(\eta_i)$. Since the size of chromosome space ($k^{k^{2r+1}}$) is in our case 2^{128}, full search can not be applied. The GA model implemented in our evolutions uses a one-point cross-over and elitist selection. Implementation details can be found in [5]. Fitness $F_N^{I_1}(\phi)$ is basically a performance on $I_1 = 100$ density-uniform ICs. After each evolution we calculated performance $P_N^{I_2}(\phi)$ of selected chromosomes more accurately for both, density-uniform (each IC density has the same probability) and uniform distribution (each IC has the same probability), $N \in \{149, 599, 999, 1001, 1301\}$ and $I_2 = 10^4$. Acceptable solutions were achieved only for small part of the total of 202 evolutionary runs. Due to the local scope of individual cells ($r << N$), best CAs or strategies are pushed to employ the global information exchange. The evolutionary dynamics of leader election is not quite smooth and shows some very dramatic leaps or innovations. From the computational perspective the most important one lies between the fitness 0.4 and 0.8, since it separates localistic from the global particle-based strategies. Evolutionary process uncovered various strategies which were further examined and analyzed. They can be roughly categorized according to their characteristic computational aspects into - strategy of mandatory function, density reduction, divide and eliminate strategy, first particle-based strategy and strategy of mirror particles. First three strategies are localistic and are based on the statistical parameters of generated ICs. There is no information exchange on the long distance, therefore leader is elected just on short subsequences of cells, what results to sharp decrease of performance with regard to N. The last two strategies are based on particle models of computational mechanics exhibiting coordination of cells at the global scale. We will focus on the evolutionary most successful strategy of mirror particles in more details.

5 Strategy of Mirror Particles

Main characteristics of this strategy is the occurrence of pair-like particles moving the same velocity but in opposite directions, therefore we call them mirror particles (see Fig. 2). Mirror particles α and β, γ and δ, ε and ζ lie at the border of domain 0^* and the zig-zag domain of $(01)^*$. The main rule responsible for leader (particle ω) election is the interaction of $\alpha + \beta \rightarrow \gamma + \delta$ and sequentially $\gamma + \delta \rightarrow \omega$ (Fig. 3(a)). Collision of α and β indicates that in the place of their interaction a potential leader might emerge. Consequently particles γ and δ are emitted to verify if there are any particles left. They shift around the whole configuration with very high opposite velocities. In case they do not collide on their routes, they meet in the middle and they finally create a global leader ω.

Particles **P**		
Label	Boundary	Velocity
α	$\Lambda^1\Lambda^0$	1
β	$\Lambda^0\Lambda^1$	-1
γ	$\Lambda^0\Lambda^1$	3
δ	$\Lambda^1\Lambda^0$	-3
ε	$\Lambda^0\Lambda^1$	3
ζ	$\Lambda^1\Lambda^0$	-3
ω	$\Lambda^0\Lambda^0$	0

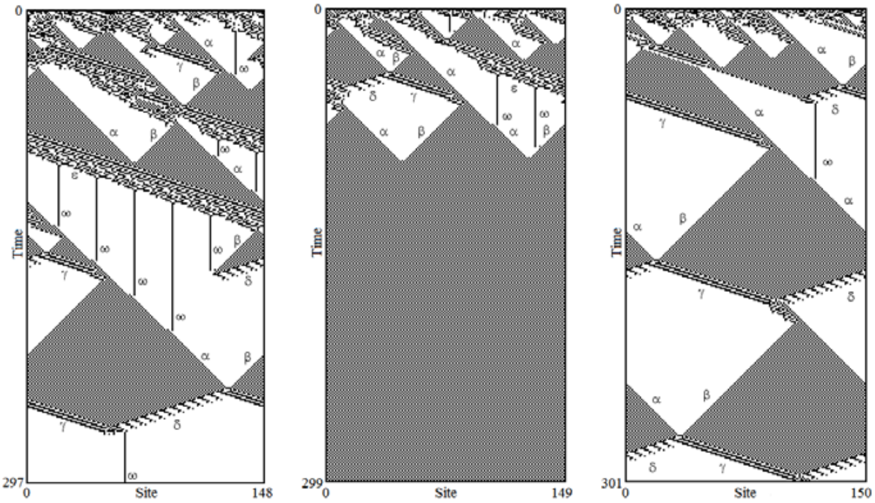
Domains $\mathbf{\Lambda}$	
Label	Regular language
Λ^0	0^*
Λ^1	$(01)^*$

Interactions **I**	
$\alpha+\beta \rightarrow \gamma+\delta \,\vert\, \emptyset(\Lambda^1)$	$\alpha+\omega \rightarrow \alpha$
$\gamma+\delta \rightarrow \omega \,\vert\, \alpha+\beta$	$\beta+\omega \rightarrow \beta$
$\alpha+\gamma \rightarrow \alpha+\beta$	$\varepsilon \rightarrow \varepsilon+\omega$
$\beta+\delta \rightarrow \omega$	$\zeta \rightarrow \zeta+\alpha+\beta$
	$\varepsilon+\zeta \rightarrow \alpha+\beta+\varepsilon$

Fig. 2. The particle catalog of the strategy of mirror-particles

The unique attribute of long-periodical particles of ε and ζ is a spontaneous emission of particles during each period. The ε generates a leader particle ω, ζ generates mirror particles α and β. Another observation is that α and β clean ω particles on their ways, what in case of $\alpha+\omega \rightarrow \alpha$ causes the phase shift of α.

There is the equal number of particles with positive and negative velocities. Particles are quite fast and the differences of colliding particles velocities are high that has a positive impact on the overall performance. The fitness of this final evolutionary strategy is 0.99. Performance reaches for $N = 149$ values of 0.944, resp. 0.992 depending on distribution type of ICs and remains high for much bigger N (see Fig. 3(d)). However, the performance for $N = 999$ cells is extremely low - just about 0.01. This finding can be maybe surprisingly explained by crucial leader electing interactions $\alpha+\beta \rightarrow \gamma+\delta$ and $\gamma+\delta \rightarrow \omega$, which actually might lead to different results according to the phases of colliding particles. Since α and β are two phase particles, there are two possible results of their interactions $\alpha+\beta \rightarrow \gamma+\delta \,\vert\, \emptyset$. Analogically γ and δ are three phase particles, hence the interactions $\gamma+\delta \rightarrow \omega \,\vert\, \alpha+\beta \,\vert\, \alpha+\beta$ can be obtained. The result of second and third interaction of $\gamma+\delta$ is the same, however the interaction process is slightly different. In the last phase of the leader election, when $\gamma+\delta$ and partially $\alpha+\beta$ have to shift around the whole configuration, the number of cells N becomes essential in determining their phases in a moment of interactions. We identified typical results of $\alpha+\beta$ and $\gamma+\delta$ interactions in relation to the modulo classes of N (Table 3(e)). Our goal is to produce only one leader (particle) ω in a final configuration. That can be achieved by the interaction of $\gamma+\delta \rightarrow \omega$ occurring for $N \equiv x \bmod 6$, $x \in \{2, 5\}$. Further, particles $\gamma+\delta$ needed for this interaction are produced for $N \equiv x \bmod 6$, $x \in \{1, 3, 5\}$. As a result, the only acceptable number of cells allowing the leader election (with satisfactory result) is 5 mod 6 also proved by experiments ($N \geq 23$). CA dynamics for the modulo classes 0, 2 and 4 leads to the global zig-zag domain Λ^1 (Fig. 3(b)), remainder 5 to the stable point of the leader particle ω (Fig. 3(a)) and 1 and 3 to the cycled behavior of $\alpha+\beta \rightarrow \gamma+\delta \rightarrow \ldots$ (Fig. 3(c)). We would like to stress that the predecessor of the strategy of mirror particles called the first-particle based strategy is not $N \equiv 5 \bmod 6$ restricted and for 999 cells reaches performance of about 0.7.



(a) (b) (c)

N	$P^{10^4}_{dens\,N}(\phi)$	$P^{10^4}_{unif\,N}(\phi)$
149	0.944	0.992
599	0.932	0.954
999	0.011	0.009
1001	0.971	0.989
1301	0.972	0.987

(d)

$N\,mod\,6$	$\alpha+\beta\rightarrow$	$\gamma+\delta\rightarrow$
0	$\emptyset$	$\alpha+\beta$
1	$\gamma+\delta$	$\alpha+\beta*$
2	$\emptyset$	ω
3	$\gamma+\delta$	$\alpha+\beta$
4	$\emptyset$	$\alpha+\beta*$
5	$\gamma+\delta$	ω

(e)

Fig. 3. Strategy of mirror particles: space-time diagrams with particle identification showing examples of (a) successful ($N = 149$) and (b)(c) unsuccessful ($N = 150$, $N = 151$) leader election; (d) performance with respect to N; (e) typical results of crucial leader electing interactions in relation to N modulo classes

6 Conclusion

In this paper we successfully analyzed and solved the problem of leader election for CA. We showed that even mere distributed system of CA consisting of indistinguishable and uniform cells operating just with a binary state is capable of emergent and complex dynamics. Evolutionary process [5] pushed CAs towards a creation of remarkable collective patterns known as domains, particles and interactions [16]. Our final evolutionary strategy of mirror particles reached performance of $94-99\%$ that was however limited by the number of cells $N \equiv 5$ mod 6. The general limitations of 1D CAs on symmetric and lossy-coupled ICs will be addressed in another paper. Our approach substantially reduced architectural requirements on the model capable of leader election. Furthermore, the presented results are generally applicable in the theory of distributed algorithms and also at the elementary level of biology and social science. Last, but not least we hope that this paper illustrated how powerful self-organization is and what processes are responsible for complex computation implemented by nature.

Acknowledgement. This work was supported by VEGA1/4053/07 and 1/0804/08.

References

1. Érdi, P.: Complexity explained. Springer, Heidelberg (2007)
2. Neumann, J.V.: Theory of self-reproducing automata - edited and completed by Burks. Illinois Press, Urbana (1966)
3. Smith, A.R.I.: Two-dimensional formal languages and pattern recognition by cellular automata. In: Proceeding of the 22nd Annual IEEE Symp. of Foundations of Computer Science (FOCS), pp. 144–152. IEEE Press, Los Alamitos (1971)
4. Mitchell, M., Crutchfield, J.P., Das, R.: Computer science application: Evolving cellular automata to perform computations. In: Bck, T., Fogel, D., Michalewicz, Z. (eds.) HandBook of Evolutionary Computation. Oxford University Press, Oxford (1997)
5. Crutchfield, J.P., Mitchell, M., Das, R.: Evolutionary design of collective computation in cellular automata. In: Evolutionary Dynamics: Exploring the Interplay of Selection, Accident, Neutrality, and Function, Oxford, pp. 361–411 (2003)
6. Mitchell, M.: An introduction to genetic algorithms. MIT Press, Cambridge (1996)
7. Crutchfield, J.P., Hanson, J.E.: Turbulent pattern bases for cellular automata. Physica D 69(3/4), 279–301 (1993)
8. Conradt, L., Roper, T.J.: Consensus decision making in animals. Trends in Ecology & Evolution 20(8), 449–456 (2005)
9. Lusseau, D., Conradt, L.: The emergence of unshared consensus decisions in bottlenose dolphins. Behavioral Ecology and Sociobiology (2009)
10. Sueura, C., Petit, O.: Shared or unshared consensus decision in macaques? Behavioural Processes 78(1), 84–92 (2008)
11. Nagpal, R.: A catalog of biologically-inspired primitives for engineering self-organization. In: Di Marzo Serugendo, G., Karageorgos, A., Rana, O.F., Zambonelli, F. (eds.) ESOA 2003. LNCS (LNAI), vol. 2977, pp. 53–62. Springer, Heidelberg (2004)
12. Lawrence, P.A.: The Making of a Fly: The Genetics of Animal Design. Wiley-Blackwell, Chichester (1992)
13. Knoester, D.B., McKinley, P.K., Ofria, C.A.: Using group selection to evolve leadership in populations of self-replicating digital organisms. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 293–300. ACM, New York (2007)
14. Angluin, D.: Local and global properties in networks of processors. In: STOC 1980: Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing, pp. 82–93. ACM, New York (1980)
15. Frederickson, G.N., Lynch, N.A.: Electing a leader in a synchronous ring. JACM 34(1), 98–115 (1987)
16. Crutchfield, J.P.: The calculi of emergence: Computation, dynamics and induction. Physica D 75, 11–53 (1994)

Update Dynamics, Strategy Exchanges and the Evolution of Cooperation in the Snowdrift Game

Carlos Grilo[1,2] and Luís Correia[2]

[1] Dep. Eng. Informática, Escola Superior de Tecnologia e Gestão,
Instituto Politécnico de Leiria Morro do Lena, Portugal
[2] LabMag, Dep. Informática, Faculdade Ciências da
Universidade de Lisboa, Portugal
grilo@estg.ipleiria.pt, luis.correia@di.fc.ul.pt

Abstract. We verify through numerical simulations that the influence of the update dynamics on the evolution of cooperation in the Snowdrift game is closely related to the number of strategy exchanges between agents. The results show that strategy exchanges contribute to the destruction of compact clusters favorable to cooperator agents. In general, strategy exchanges decrease as the synchrony rate decreases. This explains why smaller synchrony rates are beneficial to cooperators in situations where a large number of exchanges occur with synchronous updating. On the other hand, this is coherent with the fact that the Snowdrift game is completely insensitive to the synchrony rate when the replicator dynamics transition rule is used: there are almost no strategy exchanges when this rule is used.

Keywords: evolution of cooperation, update dynamics, asynchronism.

1 Introduction

The existence of cooperation in nature has been challenging to explain since, from an evolutionary point of view, this type of behavior is apparently less advantageous than a selfish one [12]. This problem is also of central importance in social sciences [1] and especially on the development and maintenance of artificial societies [11], where it is relevant to study how cooperation may be promoted and sustained. Evolutionary games are models used to study these phenomena. In these models, a population of agents interacts during several time steps through a given *game* which is used as a metaphor for the type of interaction that is being studied. The underlying structure that defines who interacts with whom is called the *interaction topology*. After each interaction session, some or all the agents, depending on the *update method* used, have the possibility of changing their strategies. The *strategy update process* is modeled using a so called *transition rule* that emulates the fact that agents tend to adapt their behavior to the context in which they live by imitating the most successful agents they know. It can also be interpreted as the selection step of an evolutionary process in which the least successful strategies tend to be replaced by the most successful ones.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 318–325, 2011.

In the research areas of dynamical systems and evolution of cooperation, *synchronous updating* has been the most used update method: at each time step, all the elements of the system are updated at exactly the same time. This practice has been widely questioned, the argument being that perfect synchronism is absent from the real world [8][9]. The most common alternative to synchronous updating is *sequential updating*, which is an extreme case of asynchronism: at each time step, exactly one element is updated. It has been shown that the level of cooperation achieved and the dynamics of these models, can be significantly affected if such an asynchronous updating is used. Previous studies on the Prisoner's Dilemma game, played on regular lattices under the *best-neighbor* transition rule, reported that synchronous updating supports more cooperators than sequential updating [8][9]. The results are the same for the Snowdrift game played under the same conditions [13]. When this game is played using the *proportional rule* (see Section 2) it was found that sequential updating favors cooperation [13]. In our work with both games [2][3][5] we confirmed the results of previous works but also found that asynchronous updating is detrimental for cooperation for very small noise values only, especially for regular networks. We also showed that the influence of the update dynamics depends mainly on the noise present in the strategy update process [4]: asynchronism becomes increasingly beneficial to cooperators as the noise level grows up to a certain value. Finally, it was found that both games are insensitive to the update dynamics when the *replicator dynamics* rule is used [6][13]. However, we showed that this rule becomes sensitive to the synchrony rate if agents are allowed to imitate less successful agents, which is equivalent to raising the noise level [4].

Here, we verify the idea suggested by Tomassini et al. [13] that strategy exchanges between agents are the reason why sequential updating supports more cooperators than synchronous updating when a proportional transition rule is used. In Section 2, the model used in the simulations is described. In Section 3, we show and discuss the results of the simulations in order to verify this idea. Finally, in Section 4, some conclusions are drawn and future work is advanced.

2 The Model and Simulations Setup

We use a model very similar to [13]. The Snowdrift (SD) game, also known as Hawk-Dove, is a two-player game where there is a task which takes a cost c to be completed and which pays a benefit $b > c$ to each player, regardless of their participation in the task completion. If both cooperate (C), they divide the cost, which results in a payoff of $b - \frac{c}{2}$ to each; If only one cooperates, it receives a payoff of $b - c$, while the defecting agent (D) receives b. If none cooperates, both receive nothing. Given these conditions, it follows that the best action depends on the opponent's decision: the best thing to do is to take the opposite action the opponent takes. As is common practice, we set $c = 1$ which leads to a cost-to-benefit ratio of mutual cooperation $r = 1/(2b - 1), 0 \leq r \leq 1$.

At each time step, agents first play a one round game with all their neighbors. After this, each agent updates its strategy with probability α using a *transition* *rule* (see below). The update is done synchronously by all the agents selected to engage in the update process. The α parameter represents the *synchrony rate* and is the same for all agents. It allows us to cover all the space between synchronous and sequential updating: $\alpha = 1$ models synchronism; as $\alpha \to \frac{1}{n}$, where n is the population size, the model approaches sequential updating.

Small-world networks (SWNs) [14] are used as interaction topologies as in [13]: first a toroidal regular 2D grid is built so that each node is linked to its 8 surrounding neighbors; then, with probability ϕ, each link is replaced by another one linking two randomly selected nodes. Self, repeated links or disconnected graphs are not allowed. These networks have the property that, even for very small ϕ values, the average path length is much smaller than in a regular network, maintaining a high clustering coefficient. Both these properties are commonly observed in real social systems. As $\phi \to 1$, we get random networks with both small average path lengths and clustering coefficients.

Two different transition rules are used to model the strategy update process: the *generalized proportional* (GP) [10] and the *replicator dynamics* (RD) [7]. Let G_x be the average payoff earned by agent x, N_x be the set of neighbors of x and c_x be equal to 1 if x's strategy is C and 0 otherwise. According with the GP rule, the probability that an agent x adopts C as its next strategy is

$$p_C(x, K) = \frac{\sum_{i \in N_x \cup x} c_i (G_i)^{\frac{1}{K}}}{\sum_{i \in N_x \cup x} (G_i)^{\frac{1}{K}}}, \tag{1}$$

where $K \in]0, +\infty[$ is the noise present in the strategy update process. Noise is the possibility that an agent imitates strategies other than the one used by its most successful neighbor. $K \to 0$ corresponds to the *best-neighbor* rule in which x always adopts its best neighbor's strategy. With $K = 1$ we have a linear *proportional* rule. Finally, for $K \to +\infty$ we have random drift where payoffs play no role in the decision process. Usually, the interval $K \in]0, 1]$ is used.

According to the RD rule, the updating agent x imitates a randomly chosen neighbor y with probability $\frac{G_y - G_x}{b}$ if $G_y - G_x > 0$. Here, b is the largest possible payoff difference between two players in a one shot game. Notice that in this rule agents do not imitate neighbors with lower payoffs.

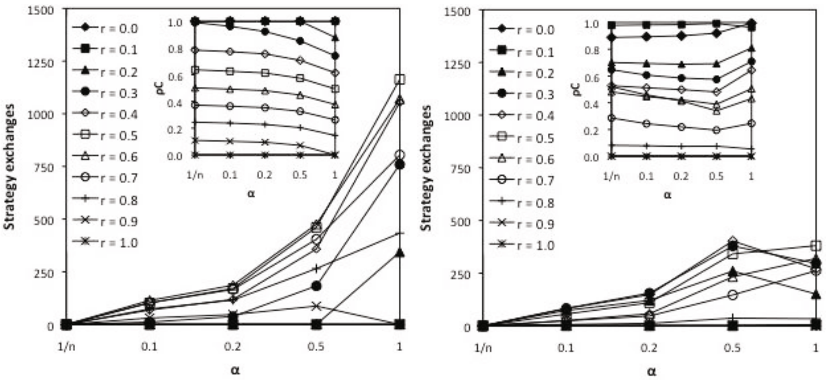
All the simulations were performed with populations of $50 \times 50 = 2500$ agents, randomly initialized with 50% of Cs and 50% of Ds. When the system is running synchronously, i.e., when $\alpha = 1$, we let it first run during a period of 900 iterations which, we confirmed, is enough to pass the transient period of the evolutionary process. After this, we let the system run for 100 more iterations and, at the end, we take as output the average proportion of cooperators, ρ_C, and the average number of strategy exchanges during this period. Simulations where $\alpha \neq 1$ are setup so that the number of individual updates is approximately the same as in the $\alpha = 1$ case. Each run is a combination of r (SD game), ϕ (SWNs), K (only for the GP rule) and α. For each tested combination, 30 runs were made and the average of these runs is taken as the output.

3 Strategy Exchanges and Cooperation

Tomassini et al. [13] suggested that, when the proportional rule ($K = 1$) is used, synchronous updating leads to less cooperation than sequential updating because in the former case agents may exchange their strategies, which is not possible in the last case. However, the authors did not verify this idea, which is the purpose of this paper. This idea raises some questions that are not answered by the authors. The first one is concerned with the transition rule: it is true that both the best-neighbor ($K = 0$) and the replicator dynamics rules do not allow direct strategy exchanges between two connected agents as the proportional rule does. That is, two agents x and y cannot infect each other simultaneously. However, it is possible that they exchange strategies indirectly: x can be infected by an agent a having the same strategy as y while y is infected by another agent b having the same strategy as x. It is not obvious also that strategy exchanges are disadvantageous for cooperators. After a strategy exchange, be it direct or not, the number of cooperators and defectors remains the same. This means that we must verify if there is a relation between the number of strategy exchanges and the final proportion of cooperators and, if it exists, we need to verify if there is a cause-effect relation between the two aspects. Finally, if we conclude that strategy exchanges negatively affect cooperation, one must try to explain why it is so.

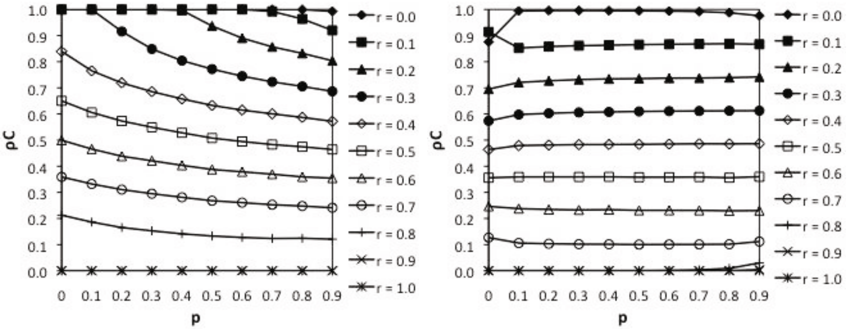
We first measure the average number of strategy exchanges, as a function of α. This average is taken over periods of $\frac{1}{\alpha}$ time steps so that the number of individual updates considered is approximately the same as for the synchronous case ($\alpha = 1$). The observed result is a strategy exchange decrease as α decreases (Fig. 1(a) shows an example). Exceptions to this result happen almost exclusively for small noise levels ($K = 0$ and $K = \frac{1}{100}$) when the GP rule is used: for some r values the number of strategy exchanges for $\alpha = 0.5$ is larger than for $\alpha = 1$ (Fig. 1(b)). In these situations, the smaller number of strategy exchanges for $\alpha = 1$ is due to cyclic dynamics in the asymptotic phase, resulting from the deterministic nature of the model (synchronous and best-neighbor rule). Even so, for the most part of these situations ρ_C is larger for $\alpha = 1$ than for $\alpha = 0.5$. For larger K values, the number of strategy exchanges is larger for $\alpha = 0.5$ than for $\alpha = 1$ only in rare cases. This happens when, for $\alpha = 1$, the system converges to uniform populations of Ds, where no exchanges can occur (ex: $r = 0.9$ in Fig. 1(a)).

Figs. 1(a) and (b) also exemplify how the number of strategy exchanges vary with K for the GP rule (K values used: $0, \frac{1}{100}, \frac{1}{10}, \frac{1}{8}, \frac{1}{6}, \frac{1}{4}, \frac{1}{2}, 1$): strategy exchanges grow with K, specially for $\alpha = 1$. Only results for $\phi = 0.05$ are shown but this pattern arises for all the tested topologies (ϕ values used: 0, 0.01, 0.05, 0.1, 1). On the other hand, we verified that there are almost no strategy exchanges when the RD rule is used: the maximum number of strategy exchanges never exceeds 4 during each $\frac{1}{\alpha}$ period no matter the α value used. We recall that, when the GP rule is used, smaller α values are more beneficial for cooperators as K grows (see Section 1). We also recall that the game is completely insensitive to α when the RD rule is used. This suggests that there is indeed a close



(a) (b)

Fig. 1. Number of strategy exchanges and ρ_C (insert) as a function of α, when the game is played on SWNs ($\phi = 0.05$) using the GP rule. (a) $K = 1$, (b) $K = 0$.



(a) (b)

Fig. 2. Proportion of cooperators, ρ_C, as a function of p when the game is played on regular networks ($\phi = 0$) using (a) the GP rule ($K = 1$) and (b) the RD rule

relation between the number of strategy exchanges and how the model reacts to α changes. In other words, the larger the difference between the number of strategy exchanges occurring under synchronous and sequential updating, the larger is the difference between the level of cooperation achieved with synchronous updating (less cooperation) and sequential updating (more cooperation). However, this result does not allow us to establish a cause-effect relation between the number of strategy exchanges and the proportion of Cs since strategy exchanges are just a consequence of the input parameters.

In order to verify the effect of strategy exchanges on the level of cooperation, we did the following experiment: on each time step, either a randomly chosen agent is updated with probability $1-p$ using the transition rule or two randomly chosen neighbor agents exchange their strategies with probability p. We note that no strategy exchanges can occur as a result of the utilization of the transition

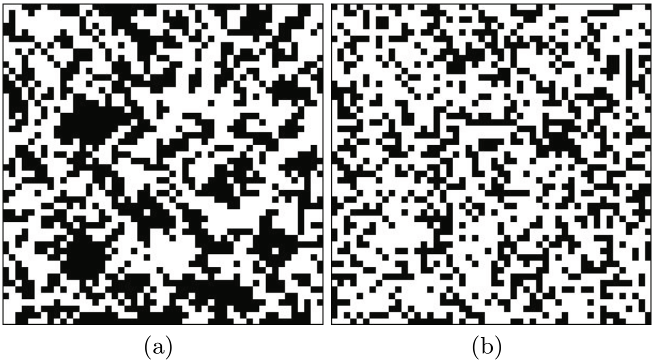


Fig. 3. Typical asymptotic patterns when the game ($r = 0.6$) is played on regular networks ($\phi = 0$) using the GP rule ($K = 1$). (a) $p = 0$, $\rho_C = 0.4692$; (b) $p = 0.9$, $\rho_C = 0.3636$. Colors: black for cooperators and white for defectors.

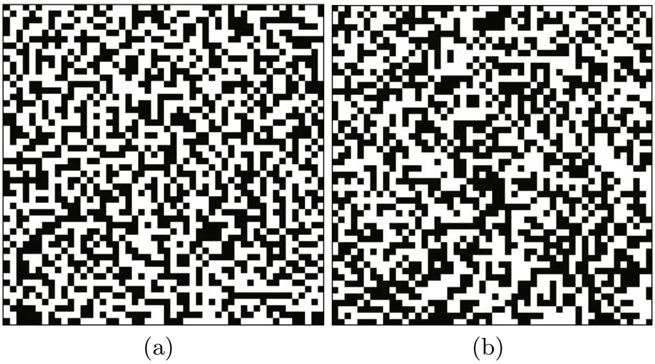


Fig. 4. As in Fig. 3 but for the RD rule and $r = 0.4$. (a) $\rho_C = 0.4668$; (b) $\rho_C = 0.4788$.

rule since only one agent is selected. This means that, when $p = 0$, this is a sequential system. Fig. 2(a) and (b) exemplify the effect of strategy exchanges on the level of cooperation, respectively for the GP and RD rules, when the game is played on regular networks. For reasons of space only these two examples are shown but the results are similar no matter the noise level (only GP rule) and the interaction topology used. Results for $p = 1$ are not shown also because in this regime we have random drift, where payoffs do not influence the dynamics, and the cooperation level always converges around 0.5, which is the initial proportion of cooperators. The results are the following: when the game is played under the GP rule, the proportion, ρ_C, of Cs decreases as the probability of strategy exchanges increases. On the other hand, when the RD rule is used, the game is almost insensitive to p.

The difference between these two behaviors can be understood if we look at the spatial patterns formed by the agents during the evolutionary process. Fig. 3 shows two examples of asymptotic spatial patterns achieved with the GP rule, for $p = 0$ and $p = 0.9$. When no strategy exchanges are allowed, C agents organize into more compact clusters. This is a well known phenomena: structured populations allow C agents to form clusters so that they interact mainly with each other, thus protecting themselves from exploration by D agents. This is important to understand how strategy exchanges influence the level of cooperation. A strategy exchange between a C and a D in the fringe of a cluster pushes the C away from the other Cs. At the same time, it introduces a D inside the cluster or, at least, it contributes to more irregular cluster frontiers, which is also detrimental for cooperators [10]. This can be seen in Fig. 3(b) where there are more isolated C agents and more filament-like clusters. The situation is different when the RD rule is used: C agents organize into filament-like clusters and this pattern does not change when strategy exchanges are introduced (Fig. 4). That is, when the RD rule is used, agents do not organize into compact clusters even when there are no strategy exchanges. This means that when strategy exchanges are introduced, there are no compact clusters to destroy and that is the reason why both the spatial patterns and ρ_C are not affected.

4 Conclusion and Future Work

We verified the idea from Tomassini et al. [13] that, when the Snowdrift game is played under the proportional transition rule ($K = 1$), sequential updating supports more cooperators than synchronous updating because in the last case strategy exchanges may occur, which is not possible in the former case. The results of the simulations, put together, are a strong evidence that this idea is correct. We saw that strategy exchanges are detrimental to the evolution of cooperation because they destroy compact clusters of agents when these exist, which is disadvantageous for cooperators. The results show that, when the *generalized proportional* transition rule is used, the number of strategy exchanges grows considerably as the noise level gets larger, mainly for synchronous updating. This explains why smaller synchrony rates increasingly favor cooperation as the noise level grows. On the other hand, when the *replicator dynamics* rule is used, there are almost no strategy exchanges, which explains why this rule is completely insensitive to the synchrony rate of the system. The way how evolutionary games depend on the update dynamics has already been reported in other works, including ours, but this work, complementing the work by Tomassini et al., offers an explanation for how they react to changes in this parameter. The results presented in this paper are also in line with the one achieved in our previous work that the sensitivity of evolutionary games depends mainly on the noise level present in the strategy update process [4]. This is important because it means that, in order to build less sensitive artificial societies, special care should be taken in the design of strategy update processes and agents' perception skills, namely by trying to avoid less successful agents to be imitated.

In future developments of this work we will verify this result with the Prisoner's Dilemma game. The way this game reacts to synchrony rate changes is similar to Snowdrift. It is, however, more sensitive than the Snowdrift when the GP rule is used, which turns it a good candidate for the task of verifying how general this result is.

Acknowledgements

This work is partially supported by FCT/MCTES grant SFRH/BD/37650/2007.

References

1. Axelrod, R.: The Evolution of Cooperation. Penguin Books (1984)
2. Grilo, C., Correia, L.: Asynchronous Stochastic Dynamics and the Spatial Prisoner's Dilemma Game. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) EPIA 2007. LNCS (LNAI), vol. 4874, pp. 235–246. Springer, Heidelberg (2007)
3. Grilo, C., Correia, L.: The Influence of Asynchronous Dynamics in the Spatial Prisoner's Dilemma Game. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 362–371. Springer, Heidelberg (2008)
4. Grilo, C., Correia, L.: What makes the spatial prisoner's dilemma game sensitive to asynchronism? In: Proceedings of the 11th Int. Conf. on the Simulation and Synthesis of Living Systems, Alife XI, pp. 212–219. MIT Press, Cambridge (2008)
5. Grilo, C., Correia, L.: The influence of the update dynamics on the evolution of cooperation. International Journal of Computational Intelligence Systems, Special Issue on Computational Intelligence Issues in Systems Experiencing Nonlinear Dynamics and Synchronization 2(2), 104–114 (2009)
6. Hauert, C., Doebeli, M.: Spatial structure often inhibits the evolution of cooperation in the snowdrift game. Nature 428, 643–646 (2004)
7. Hofbauer, J., Sigmund, K.: Evolutionary Games and Population Dynamics. Cambridge University Press, Cambridge (1998)
8. Huberman, B., Glance, N.: Evolutionary games and computer simulations. Proceedings of the National Academy of Sciences 90, 7716–7718 (1993)
9. Newth, D., Cornforth, D.: Asynchronous spatial evolutionary games: spatial patterns, diversity and chaos. In: Proceedings of the 2007 IEEE Congress on Evolutionary Computation, pp. 2463–2470 (2007)
10. Nowak, M., Bonhoeffer, S., May, R.M.: More spatial games. International Journal of Bifurcation and Chaos 4(1), 33–56 (1994)
11. Oh, J.C.: Cooperating search agents explore more than defecting search agents in the internet information access. In: Proceedings of the 2001 Congress on Evolutionary Computation, CEC 2001, pp. 1261–1268. IEEE Press, Los Alamitos (2001)
12. Smith, J.M.: Evolution and the Theory of Games. Cambridge University Press, Cambridge (1982)
13. Tomassini, M., Luthi, L., Giacobini, M.: Hawks and doves on small-world networks. Physical Review E 73(1), 016132 (2006)
14. Watts, D., Strogatz, S.H.: Collective dynamics of small-world networks. Nature 393, 440–442 (1998)

Learning in Minority Games with Multiple Resources

David Catteeuw and Bernard Manderick

Computational Modeling Lab
Vrije Universiteit Brussel, Pleinlaan 2,
1050 Brussels, Belgium

Abstract. We study learning in Minority Games (MG) with multiple resources. The MG is a repeated conflicting interest game involving a large number of agents. So far, the learning mechanisms studied were rather naive and involved only exploitation of the best strategy at the expense of exploring new strategies. Instead, we use a reinforcement learning method called Q-learning and show how it improves the results on MG extensions of increasing difficulty.

1 Introduction

J. von Neumann, inventor of game theory, showed that economic behavior (i.e. maximizing your share of economic goods) is in fact the same as players interacting in a game [1]. Game theory assumes that all players act rationally, they deduce their optimal action for each interaction. Later, J. Maynard Smith explained how game theory applies in biology under natural selection [2]. In an evolutionary scenario, each individual interacts many times with many other individuals. The more successful the individual, the higher the probability it passes its strategy to the next generation. Natural selection on populations replaces rationality used in traditional game theory. In 1994, the economist W. Brian Arthur criticized the deductive reasoning assumption and introduced the El Farol Bar problem [3]. In real-world situations, players cannot cope with the complexity of the situation or competitors simply choose not to play rationally. In such problems, players choose their actions according to what they think other players will do. These actions in turn are precedents for other players which then in turn adjust their beliefs.

The Minority Game (MG) is a mathematical formulation of the El Farol Bar problem by statistical physicists Challet and Zhang [4]. It can be seen as an abstraction of many real-life situations that are governed by a so-called minority rule. The transportation system is one such example. It is a complex system where many people use a common network of roads to get to their destination – preferably as fast as possible. However, each road has limited capacity, i.e. only a fixed number of people can use the same road at the same time. When too many people use the same route (at the same time), they cause traffic jam and their travel time increases considerably. The Internet is another example of

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 326–333, 2011.

a complex network with an important minority rule. The data packets travel faster through less-used routers. A third example with a totally different type of resource: animals try to find areas where food is abundant but where competitors are few.

In each of these cases we have a complex system where individuals (travelers, ISPs and animals respectively) have access to a limited amount of resources (network of roads, bandwidth and food respectively). The individuals do not care about the welfare of each other (or the whole population). They are self-interested and only want to fulfill their own needs. However, it is clear that the actions/decisions of one particular individual affect the success of the others. And vice versa: the actions of the others affect his own welfare.

The common assumption in game theory mentioned above is that agents always behave rational and solve problems deductively. The El Farol Bar problem [3] demonstrates a limitation of deductive reasoning. In this problem, many agents have to decide autonomously whether or not they will attend Thursday's night live concert based on the attendances of the previous weeks. Every agent prefers to attend as long as it is not too crowded. Note that there is no rational best strategy. When the best action would be to attend, then everybody would go and they would all end up in the majority.

Inductive reasoning, i.e. learning, was proposed as an alternative. Agents make decisions based on what they think the other agents will do and these beliefs are influenced by what the other agents have done in the past. Each new decision becomes in its turn a precedent that will influence the future decisions of the others. In other words, it creates a feedback loop from precedents to beliefs to decisions and back to precedents. Inductive reasoning assumes that this feedback loop eventually leads to a steady state which corresponds to the best possible outcome for the agents involved.

In the standard MG [4], the first mathematical formulation of the El Farol Bar problem, a large but odd number of agents have to make independently and repeatedly one of two choices. The agents who end up on the minority side win and are rewarded while the others, i.e. the majority, are punished. The only information available to the agents is the distribution over the two outcomes in the previous time-steps. Of course, every agent wants to be in the minority but this is simply impossible and the optimal outcome for the agent society as a whole is the largest possible minority given the total number of agents.

The standard MG can be interpreted as a simple resource allocation problem: there is a single resource with a fixed optimal capacity level η and agents can decide whether to use it or not. The resource can only be used efficiently if the fraction of agents using it is less than η. Extensions of the MG include (i) allowing the optimal capacity level $\eta(t)$ to change over time and see whether agents can adapt to the new capacity level, and (ii) introducing multiple resources and see whether they can balance the load over the different resources.

Since its introduction, the MG has received a lot of attention in the statistical physics community because of its similarities with complex disordered systems. For an overview of MG-related research in statistical physics we refer to [5,6].

It has been shown that in the standard MG a very simple learning rule results in an optimal steady state [4] but it fails when applied to the extensions mentioned above [7].

In previous work [8] we applied Q-learning, a reinforcement learning technique [9]. We found good results both on the standard MG and the time-dependent MG. Here, we apply Q-learning on MGs with multiple resources, the second extension mentioned above. In the following we first give a formal description of the standard MG and the MG with multiple resources. Then we discuss reinforcement learning in general and Q-learning in particular. Finally, we present our experiments, results and conclusions.

2 The Standard Minority Game

The standard MG is defined as follows [4]:

- The MG is played with a large but odd number of agents N, i.e. $N = 2k + 1$ for some integer k. All agents use the same simple learning rule.
- Each agent i can choose between two possible *actions* a_i: either to use the resource – represented by 1 – or not to use it – represented by 0.
- The *optimal resource capacity* $\eta = 50\%$, this means the minority group is at most k and the majority group is at least $k + 1$.
- The payoff is +1 if the agent is in the minority and −1 if it is in the majority.
- The agents have a memory of m bits to store the last m winning actions, i.e. the action chosen by the agents in the minority. The memory is updated after every stage of the game. This information is used by each agent to decide which action to take at the next time-step.
- Agents also have a set of (at random chosen) strategies S. A *strategy* $s \in S$ maps each possible combination of m winning actions to the action a_i to be taken next by agent i whenever that combination occurs. Table 1 shows two strategies.
- Each strategy has a *score* reflecting how successful it was in the past. After each stage, the agents increase the score of all their strategies that predicted the minority action. The next time, they use the prediction of their best strategy. This is the simple learning rule.

Table 1. An example of two strategies for $m = 3$, i.e. agents remember the last 3 winning actions. In that case, there are 8 possible combinations. If the winning actions at time-steps $T - 3$, $T - 2$ and $T - 1$ were 0, 0 and 1 respectively, then agent i will apply action 1 if it uses s_1 and action 0 if it uses s_2. The agent uses the strategy with the highest score. Ties are broken randomly.

	000	*001*	010	011	100	101	110	111
s_1	0	*1*	0	1	1	0	0	1
s_2	0	*0*	0	1	1	0	1	0

3 Minority Games with Multiple Resources

In MGs with multiple resources, every agent $j = 1, \ldots, N$ from a large population needs to choose repeatedly one out of R resources. At any time t, the resources $i = 1, \ldots, R$ can only satisfy a limited number of agents ($\eta_i(t)N$, with $0 \leq \eta_i(t) \leq 1$). We denote the number of agents choosing resource i at time t by $\#_i(t)$. Only resources that are not overused, i.e. $\#_i(t) \leq \eta_i(t)N$, will satisfy their users. The agents are rewarded with a point (+1) while others are punished (−1). Users do not know each other's intentions (i.e. the resource they will select next time). Also, they do not know the capacities of the resources.

Here, we consider the case with just enough resources to please the whole agent population: $\sum_{i=1}^{R} \eta_i(t) = 1$.

As in the standard MG, the ideal outcome is where as many agents as possible are satisfied each time. We do not want that some agents are excluded all the time. In terms of resource demand this means all resources should be used optimally. I.e., the number of agents $\#_i(t)$ selecting resource i should be as close as possible to the capacity of that resource ($\eta_i(t)N$). Or, the overuse of resource i, measured by $A_i = \#_i(t) - \eta_i(t)N$ should approach $A_i \rightarrow 0$.

Next, we discuss the learning scheme we apply and show its advantages over the simple learning rule often used in MGs.

4 Reinforcement Learning

Reinforcement Learning (RL) agents solve problems using trial and error. Unlike in supervised learning, agents are not told which actions or decisions are best. Instead, agents receive a reward[1] after each action taken. They also have (some) information about the state of their environment. The goal of the agent is to find an optimal policy. A policy is a mapping from each state of the environment to a probability distribution over the possible actions in that state. The agent maximizes its total expected reward if it takes all actions according to an optimal policy.

Since agents are not told which action to take, they should balance exploration and exploitation. While the agent wants to exploit its knowledge of the environment most of the time in order to maximize its rewards, once in a while the agent should take an exploratory action in order to discover actions that are better than the best one found so far.

The environment is allowed to be non-deterministic: there are probabilities for each transition from one state to a successor state and for the rewards that the agent receives. An important assumption is that these probabilities remain fixed over time, i.e. the environment is stationary. See [9] for an overview of RL.

4.1 Q-Learning

One well-known RL-algorithm is Q-learning. It maps each state-action pair (s, a) to the total expected reward if the agent applies action a in state s. One can

[1] The reward is a scalar, possibly zero or negative. When negative, it is also called punishment.

proof that Q-learning will find the optimal policy, i.e. the Q-values converge to the true total expected reward provided that the environment is stationary [10]. Unfortunately, in a multi-agent setting the environment is non-stationary due to the presence of other agents that also learn. Whereas in a stationary environment exploration can be totally ignored when enough information has been collected, in a non-stationary environment the agent has to continue exploring in order to track changes in the environment.

Q-values are initialized with an arbitrary value, e.g. 0. After each action a, the agent updates $Q(s, a)$ according to the rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \tag{1}$$

where a is the action taken in state s, r is the immediate reward that follows this action, s' is the new state of the environment and a' is the best action currently known to the agent when in state s'. The learning rate $0 < \alpha \leq 1$ controls the update speed of the Q-values. The discount factor $0 \leq \gamma < 1$ allows to weight the importance of future rewards. In the extreme case of $\gamma = 0$ only the immediate reward is taken into account.

Q-learning only tells the agent how to exploit but not how to explore. Therefore we need an exploration strategy or action-selection rule like ϵ-greedy or softmax. In the ϵ-greedy action-selection strategy (where ϵ is small), the agent selects with probability $1 - \epsilon$ the action with the highest Q-value and with probability ϵ it selects an action at random.

5 Experiments

In this section we first discuss related work on the MG with multiple resources and we put forward an alternative hypothesis. Next, we explain how we apply Q-learning to the MG followed by a discussion of the experimental setup. Finally, we present and discuss our results.

5.1 Related Work

Galstyan and colleagues [7] have shown that agents using the simple learning rule in the time-dependent MG cannot track changes in the optimal resource capacity $\eta(t)$. To solve this problem, they have restricted the information available to the agents. Before, all agents knew the last m winning actions. Now, agents only know what their immediate neighbors did in the previous stage. They use this information to decide their next action. In other words, a neighborhood structure is imposed on the agents and they have only local information available as opposed to global information in the standard MG. An agent population with local information manages to track changes in optimal resource capacity $\eta(t)$. The neighborhood structure used for the agents are the random NK-networks: N is the number of agents, i.e. nodes in the network, and K is the number of neighbors of each agent in the network. These K neighbors are chosen at random when the network is constructed.

In previous work we illustrated an alternative hypothesis. The simple learning rule used in the MG is naive from a reinforcement point of view since there is only exploitation, i.e. the current best action is always applied. We have shown that a proper balance between exploitation and exploration can track changes in the resource capacity. We implemented this using Q-learning.

In [7] the authors extended their work to MGs with multiple resources. Here, we also test our hypothesis on these generalized MGs.

5.2 Q-Learning in the Minority Game

In order to apply Q-learning to the MG, the rewards, state- and action-space must be defined. The actions simply refer to one of the resources $i = 1, \ldots, R$. Each agent is punished ($r = -1$) when choosing an overused resource; and is rewarded ($r = +1$) otherwise. The agents receive only local information or, no information regarding the state of the environment.

The first type of agents (Q_0) receive no information from the environment. The second type of agents (NK+Q_2) have to choose their next action based on the last decision of their K neighbors, i.e. the agent society is organized according to a random NK-network.

Rests us to choose the parameter values: the learning rate α is set arbitrarily to 0.1 and the discount factor γ is set to 0 since experimental results have shown that the dynamics of the MG is not affected by the order in which the states are visited [5]. This results in the following update rule:

$$Q(s,a) \leftarrow Q(s,a) + 0.1(r - Q(s,a)) \tag{2}$$

To select an action we use the ϵ-greedy strategy. After some experimentation a reasonable value for most simulations was found to be $\epsilon = 0.01$.

5.3 Experimental Setup

The performance of the Q-learning agents is compared to two other types of agents:

***RND*-agents** choose a resource i with probability equal to the resource's capacity level $\eta_i(t)$,

***NK*-agents** are organized according to an NK-random graph as proposed by Galstyan et al. [7]. They use the last action of their K neighbors to decide which action to take.

One might say that *RND*-agents are cheating. They know the capacity of the resources at each time-step, whereas all other agents have no access to this information. Anyway, *RND*-agents are used as benchmark. Q-learning agents are considered to use the resources efficiently only if the resulting volatility is lower than that for the *RND*-agents.

Table 2. Volatilities (average μ_V and standard deviation σ_V) for the MG with 3 resources. The resources have fixed capacities 0.5, 0.3 and 0.2. The result of NK-agents were found for $P = 0.60$.

agent type	K	ϵ	μ_V	σ_V
RND	0	–	0.208152	0.00694868
NK	2	–	0.0265430	0.0377557
Q_0	0	0.01	0.00784792	0.000489822
$NK+Q_2$	2	0.01	0.0395739	0.00709967

Table 3. Volatilities (average μ_V and standard deviation σ_V) for the MG with 3 resources with fluctuating capacities as mentioned in the text

agent type	K	ϵ	μ_V	σ_V	K	ϵ	μ_V	σ_V
RND	0	–	0.219063	0.00718633	0	–	0.197639	0.00695407
NK	2	–	0.118924	0.157421	–	–	–	–
Q_0	0	0.01	0.0166284	0.00118486	0	0.01	0.0349235	0.00252304
$NK+Q_2$	2	0.01	0.0907612	0.00789532	2	0.01	0.461914	0.0784233

The experiments of which we show the results are all run 100 times. Each simulation lasts for 10,000 time-steps. The volatility V (averaged over all resources) is measured during the last 1,000 time-steps according to the equations below. Furthermore, we used a population of size $N = 101$.

$$A_i(t) = \#_i(t) - \eta_i(t)N \tag{3}$$

$$\sigma_i^2 = \frac{1}{T}\sum_{t=0}^{T} A_i(t)^2 \tag{4}$$

$$V = \frac{\sigma_{tot}^2}{N} = \frac{1}{NR}\sum_{i=1}^{R}\sigma_i^2 \tag{5}$$

5.4 Results for Multiple Resources

In Table 2 results are given for a MG with $R = 3$ resources ($\eta_1 = 0.5$, $\eta_2 = 0.3$ and $\eta_3 = 0.2$). We see the Q_0-agents perform significantly better. The left side of Table 3 is the result of changing capacities according to following sine functions:

$$\eta_1(t) = 1/3 + 1/6\sin(2\pi t/1000) \tag{6}$$

$$\eta_{2,3}(t) = 1/3 - 1/12\sin(2\pi t/1000) \tag{7}$$

Next, we doubled, both the size and frequency of the fluctuations in resource capacities. The results are shown in the right side of Table 3. Note that the

NK-agents are not mentioned. We simply did not found a value for the strategy bias P giving satisfying results[2]. We searched the range $[0, 1]$ in steps of 0.05. Volatility was almost always 10 times worse than for the *RND*-agents.

In general, we see that Q-learning agents Q_0 (i.e. without communication) perform best in these multiple resource MGs. Adding (local) information decreases system performance. The *NK*-agents of Galstyan are difficult to tune. Their results vary largely, possibly due to the relatively small network sizes, $N = 101$ nodes. But more likely, because of the way the strategies are created.

6 Conclusion

In this paper, we showed first that reinforcement learning is a good alternative for the learning schemes used so far in the MG. We showed that the simplified Q-learning agents can handle the MG extensions. Moreover, they can deal with difficult multiple time-dependent resource capacities which the *NK*-agents cannot.

It shows that learning is an important feature of adaptive systems. The ϵ-greedy action selection strategies allows the agents to continuously explore. This creates a situation were they 'coordinate', i.e. they adapt to each other's adaptation. In order to understand this phenomenon better, we would like to compare our current results with Learning Automata and softmax action selection.

References

1. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)
2. Smith, J.M.: The Theory of Evolution. Penguin Books (1958)
3. Arthur, W.B.: Inductive reasoning and bounded rationality (the el farol problem). Am. Econ. Rev. 84, 406–411 (1994)
4. Challet, D., Zhang, Y.C.: Emergence of cooperation and organization in an evolutionary game. Physica A 246, 407 (1997)
5. Moro, E.: The minority game: an introductory guide (2004)
6. Challet, D., Marsili, M., Zhang, Y.C.: Minority Games. Oxford University Press, Oxford (2005)
7. Galstyan, A., Kolar, S., Lerman, K.: Resource allocation games with changing resource capacities. In: Proc. of the Int. Conf. on Autonomous Agents and Multi-Agent Systems, pp. 145–152 (2003)
8. Catteeuw, D., Manderick, B.: Learning in the time-dependent minority game. In: Proc. of the Genetic and Evolutionary Computation Conf., pp. 2011–2016 (2009)
9. Sutton, R., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
10. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8, 279–292 (1992)

[2] See [7] for Galstyan et al.'s hypothesis on the importance of the strategy bias P.

Robustness of Market-Based Task Allocation in a Distributed Satellite System

Johannes van der Horst, Jason Noble, and Adrian Tatnall

University of Southampton, Southampton,
SO17 1BJ, United Kingdom
jgvdh08r@ecs.soton.ac.uk

Abstract. The perceived robustness of multi-agent systems is claimed to be one of the great benefits of distributed control, but centralised control dominates in space applications. We propose the use of market-based control to allocate tasks in a distributed satellite system. The use of an artificial currency allows us to take the capabilities, energy levels and location of individual satellites, as well as significant communication costs into account. Simulation is used to compare this approach to centralised allocation. We find the market-based system is more efficient and more robust to satellite failure, due to the adaptive allocation of tasks.

1 Introduction

The robustness and scalability of multi-agent systems are articles of faith in the artificial life community. Researchers observe that, in social insects, in human organisations and in markets, the behaviour of the whole is not critically dependent on the functioning of any single component. In recent years these types of social and biological networks have inspired a host of technological systems in which robust, cooperative behaviour is achieved across a group of autonomous agents. However, the view that distributed systems should be controlled from the centre still dominates in other engineering fields. A notable example is the engineering of spacecraft where the central control paradigm has been extremely successful. This can be traced back to a history of monolithic spacecraft, incremental development philosophies and high mission costs. In these systems, reliability is usually achieved through redundancy, fault detection and error correction. The centralised and distributed control approaches are rarely compared head to head, because their respective applications are often completely different.

The idea of using multiple coordinated spacecraft to perform the function of a single larger vehicle has recently been proposed[1]. Distributing functionality between the component spacecraft allows larger structures to be constructed in orbit, while also benefiting from the commoditisation of spacecraft, rapid deployment and mission flexibility. The increased complexity requires that the system handles component failure transparently, while also abstracting the management of components in the system. The control of such a system lies at the intersection of the spacecraft engineering and multi-agent systems fields, with

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 334–341, 2011.

neither approach assuredly superior. In this paper we investigate the suitability of a distributed multi-agent system solution by comparing its performance to a centralised control implementation. Using a simplified task allocation model also allows us to verify the widely assumed increased robustness offered by a multi-agent system, which is one of the primary motivations for using this approach.

In [2] a multi-objective evolutionary algorithm is used to find a trade-off between signal delay and transmission power cost for communication in a multi-satellite system. Power consumption is decreased by using multi-hop routing, with the option of long distance transmissions to meet communication time constraints. An artificial potential field is used in [3] to position spacecraft in a lattice formation. By using a bottom-up approach it provides scalability and robustness. Self-organisation is also used in the proposed in-orbit assembly of large structures[4]. None of the above addresses the problem of task distribution in a group of spacecraft.

From a computational point of view, a group of spacecraft can be seen as a multi-robot system, where coordination in a hostile and noisy environment is a challenge. Alternatively, the multi-satellite system can be abstracted as a distributed computing network, where tasks need to be efficiently allocated to different nodes. Natural systems have provided useful metaphors for obtaining desired emergent behaviours in multi-robot contexts; for example, aggregations of robots through mimicking cockroaches[5], and divisions of labour[6] and foraging strategies [7] inspired by social insects. From the distributed computing perspective marketplaces have been popular mechanisms for allocating tasks in a distributed fashion. Typically, agents bid based on their suitability to perform a task and an auctioneer selects the best agent based on the size of the bid. The use of an artificial currency allows agents to make adaptive local decisions in a system where global information is incomplete and out of date. When viewed from a system-level perspective, these market-driven systems display collective social adaptive behaviour[8].

In [9] a market is used to allocate computing tasks in a heterogeneous network. The system self-organises to distribute loads fairly, by mapping idle resources into currency. The trade-off between sensing and data routing in wireless sensor networks (WSNs), is managed using a market mechanism in [10]. Nodes decide which role to fulfil, based on the payment they receive. By maximising their own revenue, the system performs close to optimal, dealing with changing sensor numbers and extending network lifetime. [11] combines the market-based allocation with robotics to efficiently explore a unknown territory. Robots compare their own cost of visiting a waypoint with trading it with a potentially better-situated vehicle. This maximises information retrieved while minimizing the system cost. Target allocation in unmanned miniature aerial vehicles (MAVs) using a market is presented in [12]. A distributed auction scheme takes the kinematic and sensing constraints of MAVs account.

The above examples demonstrate the utility of market-based control (MBC) in task allocation in distributed systems. The limited power and high communication costs in a distributed satellite mission are, however, not fully addressed.

While WSNs share these constraints, their computation and communication demands are quite different. We will use a distributed market-based control approach that encapsulates the energy of individual nodes, their capabilities and their location in the network to achieve robust, adaptive allocation.

Although several multi-spacecraft missions have been proposed, no actual implementations have been flown yet. We therefore propose the following reference mission: A group of small, low-cost satellites, numbering in the tens to hundreds are positioned in close proximity to each other in low earth orbit. Due to power constraints, communication is limited to only take place between neighbours, forming a network of autonomous, yet highly interdependent agents. The agents are specialised, with different classes displaying different skills: some are equipped to communicate with the ground station, while others carry remote observation cameras.The different payloads can operate independently, or be merged to provide synthesized data. For example, cameras can either be operated independently for low-resolution coverage of a wide area, or be combined to provide high-resolution images of a particular area of interest. Ideally, the group of satellites should be addressable as a single virtual spacecraft, with the detailed management of individuals left to the system. This reference mission is used to construct a simulation model that can be used to compare the robustness and efficiency of the centralised and distributed control methodologies directly.

2 Model

The desired multi-satellite system has no single point of failure and the goal is to allocate tasks in a manner that maximises the total amount of work done. We represent the system as a network of agents, with connections reflecting reliable wireless links between satellites. We currently have two types of agents: uplink nodes that communicate with the ground station; and worker nodes that perform tasks. A batch of tasks is uploaded to the uplink nodes at regular intervals, similar to a satellite having periodic contact with the ground station. Considered from the point of view of an uplink node, uploaded tasks are sequentially allocated to worker nodes using a sealed-bid reverse auction.

The uplink node acts as auctioneer and announces the auction by flooding a request describing the task through the system. If a node has enough energy and the required infrastructure to complete a job, it places a bid. The request message is repeated to its neighbours, who perform the same process. The value of the bid (B) is dependent on the ratio of maximum (e_{max}) to remaining energy (e_{rem}) of the node, the size s of the task and a scaling factor α that reflects the actual energy cost of performing the type of task: $B = \alpha s \frac{e_{max}}{e_{rem}}$. Returning bids are routed along the path of the original request message, with intermediate nodes adding a constant percentage commission to the bid. The uplink node selects the lowest bidder to assign the task to. The winning agent takes responsibility for the task, decreases its remaining energy (e_{rem}) by the cost of performing the task (αs) and receives payment. All the nodes in the communication path also receive their commission. As commission is multiplicative, it encourages the

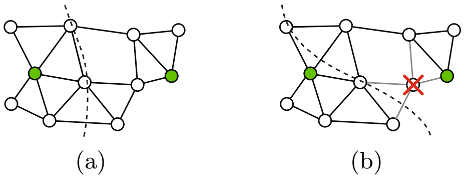


Fig. 1. Task allocation shifts to reflect changes in the network. In (a) the dotted line shows the area in which the majority of tasks sold by each auctioneer (shaded node) is allocated to. The nodes intersected by the dotted line receives tasks from both auctioneers. If a node fails, we lose its capacity to complete jobs and its routing functionality. The relative distances (measured in number of hops between nodes) and loading of nodes will change, resulting in a new distribution of labour in the network (b).

local allocation of tasks, because distant nodes appear more expensive to the auctioneer. This bidding mechanism results in inexpensive bids from nodes that are under-utilised (large e_{rem}), while nodes that receive more allocations (lower e_{rem}) will increase their bids, making them less likely to be allocated a task. Agents do not try to win tasks by underbidding others — the quoted price is an honest reflection of the internal state of the node. Communication uses a significant portion of the total energy budget, in contrast to many multi-robot systems where it can safely be treated as negligible when compared to other energy uses. On every transmission e_{rem} is decreased by a value corresponding to the packet size. The commission parameter is responsible for minimising this communication expenditure.

When this process is executed concurrently across the network, the balance between the distributive effect of energy dependent bids and the localising effect of commission results in an allocation policy that is sensitive to levels of utilisation and changes in topology. Note that any node can act as an auctioneer, but for the simplified task structure above only uplink nodes allocate tasks.

Node failure is implemented as a uniformly distributed probability of failure per node per time step. Failed nodes cannot perform any work and have no communication links, thereby altering the network topology, as shown in Fig. 1.

This model has only one type of task and a relatively simple allocation problem. Although it is a simplified scenario, we believe it still captures enough of the dynamics of the system to allow a fair evalution of control approaches. The design does, however, allow for future expansion: multiple task types can be used and recursive auctions can take place, which means any node can subcontract another node to perform part of a task. While the bid value is currently largely determined by the remaining energy in a node, it can easily be extended to include other system resources, such as bandwidth or memory.

Note the similarities between this abstraction and networked computing systems, where the best node for a task is determined by available CPU cycles, memory and disk space[9]. It also applies to wireless sensor networks, where node utility is dependent on remaining energy.

3 Results

The market-based control allocation scheme is used in a network with 100 nodes arranged in a 10 by 10 square lattice formation. Nine of the nodes act as uplink nodes; the remaining 91 are worker nodes. The duration of a day is 100 time steps. Our focus is on task allocation, so we assume formation flight is managed by a system similar to that presented in [3]. Tasks are introduced to the system at a constant rate (9 tasks per day), while nodes have failure probability of 0.001 per day per node. Approximately half the nodes usually fail after 800 days. The energy costs of tasks are generated from a Weibull distribution with shape $k = 2$ and scale $\lambda = 2$. The recharging of batteries from solar panels is implemented by increasing nodes' energy by 0.15 units per day, up to a maximum of 10 units per node. Transmission cost is set to 0.001 units per packet for negotiation and 0.1 units for allocation packets. As we currently only have a single task type, α is set to 1 for all nodes. Commission is set at 20%.

We compare our system to three other idealised control strategies to quantify its performance and robustness. Note that the following systems are unrealisable in a real world, but they do provide useful measures for comparison.

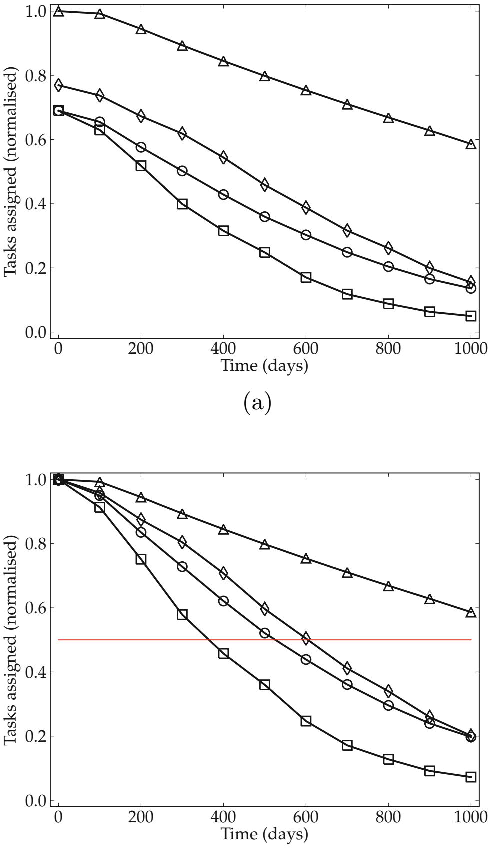
The *ideal* case represents the best possible performance. This assumes the controller has perfect knowledge of the network and can communicate cost-free with any node, without being constrained by network topology. Allocation is treated as a bin packing problem: for every task, the controller finds the worker node with the most remaining energy and assigns the task to it. The controller is considered immune against failure.

In the *centralised* approach, we have an intelligent mother ship that controls a network of simpler worker spacecraft. The level of realism is increased by reintroducing the network topology and transmission cost. The single controller node is positioned in the centre of the same lattice used by the distributed controller. The remaining 99 nodes are workers, as opposed to 91 workers for the MBC case. The controller has perfect information about the energy levels of nodes in the network, as well as the topology of the network — we ignore the cost of maintaining this information. Tasks are again assigned as in the ideal case, with the additional constraint that to allocate a job to a node, a valid path must exist between the central controller and the selected worker node. A path is valid if all nodes along it are active and have enough energy for transmission.

As the controller node in the centralised approach is a single point of failure, we assume that in a real mission scenario, it would incorporate redundancy to decrease its vulnerability. We therefore model this node as being immune to failure in the *centralised with immunity* (CI) case. All other variables are the same as used in the centralised approach.

3.1 Performance

The system was allowed to settle into steady state behaviour, before enabling the failure of nodes. The number of tasks successfully assigned over a period of



(a)

(b)

Fig. 2. Performance and robustness of different allocation strategies. In (a), network performance is measured by the total number of tasks allocated, normalised with respect to the steady-state performance of the ideal system. The amount of work stays constant, while nodes fail with a uniform probability. The ideal case, marked with triangles, provides an upper bound on the performance because it does not take communication cost or network topology into account. The market-based control allocation scheme (diamonds) deteriorates faster than the ideal case, but performs more efficiently than the centralised approaches (circles and squares). Robustness is defined as the ability to maintain performance despite satellite failures. In (b), the performance data is normalised with respect to the steady-state performance of the respective allocation strategies. The ideal case (triangles) shows the theoretical maximum obtainable, if topology and transmission cost have no influence. Market-based allocation (diamonds) shows a more gradual deterioration than either of the centralised approaches. The vulnerability of the network due to failure of the controller node is clearly visible when comparing centralised case (squares) to centralised with a controller that is immune to failure (circles). The solid horizontal line indicates 50% of the initial throughput.

100 days was measured and normalised with respect to the steady-state performance of the ideal allocator. This was repeated 100 times to obtain an average behaviour; the resulting performance is shown in Fig. 2(a).

The ideal system can be seen to form an upper bound on the allocation success. It slowly deteriorates over time as the number of failed nodes increases and the system's capacity to complete jobs decreases. The MBC approach displays lower initial performance: due to the energy cost of communication it accommodates only 76.9% of the tasks the ideal case does. Steady-state performance drops to 68.9% for both centralised control schemes, because of the larger portion of the energy budget spent on communication (the average path length when allocation tasks is greater than with MBC). Progressive node failure decreases the total capacity of the network: allocation paths becomes longer and use more energy, and the network is fragmented when all routes to functioning nodes are cut. This is reflected in the steep slope of the centralised, CI and MBC data. The sensitivity of the network to failure of the central controller is significant, as can be seen when comparing the centralised approach to CI.

Additional experiments confirmed that the behaviour of the system is robust to variations in parameter values. The qualitative observations still hold, although some quantitive changes occur. For smaller networks, the centralised and MBC results converge. If the ratio of transmission cost to task size changes, the performance will increase (for smaller packets) or decrease (for larger packets) accordingly.

3.2 Robustness

We define robustness as the ability of the system to maintain steady-state performance despite satellite failures. To compare the robustness of the different systems, the results from Sect. 3.1 are normalised with regards to their respective steady-state values (Fig. 2(b)).

The ideal case again provides an upper bound. The centralised case deteriorates rapidly, largely due to the whole network collapsing if the controller node fails. While the CI approach performs better, it is still subject to network fragmentation. The MBC approach is more robust than both centralised systems. To express these results in terms of mission reliability, we define a mission as operational while it delivers more than 50% of its initial throughput. The centralised system reaches this limit at 360 days, the immune-centralised at 525 days and the MBC approach at 605, making it the most reliable of the three. In spite of having having fewer worker nodes, the performance of the MBC system is superior. This is not only related to efficiency, but also to robustness. In particular, this is a result of having multiple uplink nodes which are able to adapt their allocation to changes in the network topology and node utilisation.

4 Discussion

We have shown that a market-based task allocation system completes more jobs and is more robust than a centralised approach, irrespective of whether the central controller is subject to failure. The improved performance is a result of lower system-level communication costs when assigning jobs and improved robustness, due to the distributed and adaptive nature of the control system. These results are promising for distributed space applications. Launch mass will always be the dominant factor in total mission cost and, assuming a given launch mass and spacecraft of equal size, our results show that more work can be done more robustly using an MBC approach.

The results are also applicable to robustly controlling systems with similar constraints, such as WSNs and distributed computing systems, by using emergent behaviour. The work presented here is the first step towards a task-allocation mechanism for a distributed satellite system. Future work will look at composite tasks, requiring cooperation; adding temporal constraints to tasks; optimising the energy cost of transmissions to match communication distances; optimal composition of the satellite types in the group; and enhanced physics to provide more realism.

References

1. Brown, O., Eremenko, P.: Fractionated space architectures: A vision for responsive space. In: Responsive Space 4 (2006)
2. Wu, X., Vladimirova, T., Sidibeh, K.: Signal routing in a satellite sensor network using optimisation algorithms. In: IEEE Aerospace Conference, pp. 1–9 (2008)
3. Pinciroli, C., Birattari, M., Tuci, E., Dorigo, M., Marco, Vinko, T., Izzo, D.: Self-organizing and scalable shape formation for a swarm of pico satellites. In: NASA/ESA Conference on Adaptive Hardware and Systems, pp. 57–61 (2008)
4. Izzo, D., Pettazzi, L., Ayre, M.: Mission concept for autonomous on orbit assembly of a large reflector in space. In: 56th International Astronautical Congress (2005)
5. Garnier, S., Jost, C., Jeanson, R., Gautrais, J., Asadpour, M., Caprari, G., Theraulaz, G.: Aggregation behaviour as a source of collective decision in a group of cockroach-like-robots. In: Advances in Artificial Life, pp. 169–178 (2005)
6. Groß, R., Nouyan, S., Bonani, M., Mondada, F., Dorigo, M.: Division of labour in self-organised groups. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 426–436. Springer, Heidelberg (2008)
7. Campo, A., Dorigo, M.: Efficient multi-foraging in swarm robotics. In: Advances in Artificial Life, pp. 696–705 (2007)
8. Cliff, D., Bruten, J.: Animat market — trading interactions as collective social adaptive behavior. Adaptive Behavior 7(3-4), 385–414 (1999)
9. Waldspurger, C.A., Hogg, T., Huberman, B.A., Kephart, J.O., Stornetta, S.W.: Spawn: A distributed computational economy. IEEE Transactions on Software Engineering 18(2), 103–117 (1992)
10. Rogers, A., David, E., Jennings, N.R.: Self-organized routing for wireless microsensor networks. IEEE Transactions on Systems, Man and Cybernetics, Part A 35(3), 349–359 (2005)
11. Zlot, R., Stentz, A., Dias, M.B., Thayer, S.: Multi-robot exploration controlled by a market economy. In: IEEE International Conference on Robotics and Automation, vol. 3, pp. 3016–3023 (2002)
12. Sujit, P.B., Beard, R.: Multiple MAV task allocation using distributed auctions. In: AIAA Guidance, Navigation and Control Conference and Exhibit (August 2007)

Hierarchical Behaviours: Getting the Most Bang for Your Bit

Sander G. van Dijk, Daniel Polani, and Chrystopher L. Nehaniv

Adaptive Systems Research Group
University of Hertfordshire, Hatfield, UK

Abstract. Hierarchical structuring of behaviour is prevalent in natural and artificial agents and can be shown to be useful for learning and performing tasks. To progress systematic understanding of these benefits we study the effect of hierarchical architectures on the required information processing capability of an optimally acting agent. We show that an information-theoretical approach provides important insights into why factored and layered behaviour structures are beneficial.

1 Introduction

Animals sometimes make performing complex tasks seem almost trivial. For instance, a praying mantis can show a wide arrange of complicated behaviours such as feeding, grooming and sexual behaviour, with very limited brain power. Understanding this is only possible when realising that their behaviour is well structured, partially in a hierarchical manner [1]. Nature supplies a large amount of examples of this kind of hierarchical behaviour, e.g. in vocal behaviour in singing birds [2] and ordering tasks in capuchin monkeys [3]. Unsurprisingly, there has for long been a call in the field of ethology to not neglect this structure [4].

Computational models of such structures are well established. Traditionally as static systems [5], but the latest advances in reinforcement learning show that adaptive hierarchical behaviour structures can significantly speed up learning [6], even when an agent has to build up its behavioural hierarchy autonomously in parallel with learning a new task [7, 8]. Recent research has investigated the relationship of biological structures to computational models of layered control [9] and modern adaptive hierarchical architectures [10].

The benefits of hierarchical organization of behaviour are intuitive: it reduces complexity, eases learning and facilitates skill transfer [6]. What is missing however is a *systematic* rather than *heuristic* understanding of the reasons for success of this kind of structures in nature and in artificial agents. Additionally, quantitative methods for comparing the growing number of computational models of behaviour hierarchies are needed. Currently these are partly, if not wholly, based on assumptions and external knowledge of their designer, of which the necessity is difficult to judge. What we seek is a minimal model that extracts and unifies commonalities between specific fields such as biology, robot control, decision theory and engineering.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 342–349, 2011.

Information theory has proven to be a useful tool in supplying such models and in understanding fundamental, global properties and limits of agent-environment systems [11–14]. Our hypothesis is that the advantages of hierarchical behaviour structures in animals and artificial agents are grounded in their effect on the way an agent processes information. Firstly, they divide the burden of information processing over different layers. Secondly, information about the current context is retained in more abstract behaviours at higher levels, relieving lower levels from performing redundant information processing. In this paper we will first research and quantify these effects separately. Subsequently we will investigate their combination.

2 Relevant Information

An agent with sensors and actuators combined with its environment forms an action-perception loop, in which at each time step t the agent perceives the state of the environment $s_t \in \mathcal{S}$ and decides on an action $a_t \in \mathcal{A}$. The dynamics of the total system are determined by the *state transition probability distribution* $\mathcal{P}^{a}_{s_t,s_{t+1}} = p(s_{t+1}|s_t, a_t)$ and the agent's *policy* $\pi(s_t, a_t) = p(a_t|s_t)$.

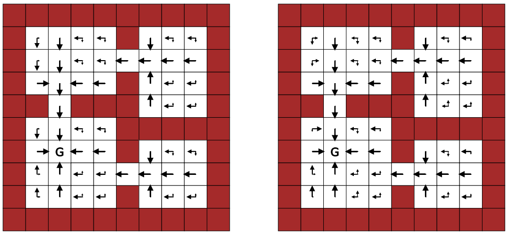
We are interested in agents operating in an environment that rewards certain behaviour, according to an *immediate reward function* $r_{t+1} = \mathcal{R}^{a_t}_{s_t,s_{t+1}} \in \mathbb{R}$. Given this reward we can determine two functions: the *state value function* $V^{\pi}(s)$, which is defined as the expected future reward an agent will receive when starting in state s and following policy π, and the *state-action value function* (or *utility function*) $U^{\pi}(s, a)$ which gives the expected future reward of taking action a when in state s and consequently following policy π [15]:

$$V^{\pi}(s) = \sum_{a} p(a|s) \sum_{s'} p(s'|s, a) \left[r(s, a, s') + \gamma V^{\pi}(s')\right] \tag{1}$$

$$U^{\pi}(s, a) = \sum_{s'} p(s'|s, a) \left[r(s, a, s') + \gamma V^{\pi}(s')\right] , \tag{2}$$

where $\gamma \in (0, 1)$ is a discount factor to model preference of short term over long term reward.

An optimal policy for an agent in such an environment is one that maximizes the expected utility $E_{\pi}\left[U(S, A)\right] = \sum_{s,a} p(s, a)U(s, a) = \sum_{s,a} p(a|s)p(s)U(s, a)$. To be able to execute this policy, the agent needs to take in and process information from the environment through its sensors. However, not all available information is needed. The concept of *relevant information* provides a concrete minimum of the amount of information an agent needs to acquire to be able to follow an optimal policy, measured by the *mutual information* between states and optimal actions [11]: $I(S; A^*) = \min_{p(a|s):p(a|s)p(s)>0 \Rightarrow a \text{ optimal for } s} I(S; A)$. This measure is an important, fundamental property of the agent-environment dynamics and the given reward function. It does not depend on the actual method of finding an optimal policy. Mutual information is measured in bits and is defined by the change in entropy (uncertainty) about A when S is being observed: $I(S; A) = H(A) - H(A|S)$.



(a) Flat (b) Memory

Fig. 1. Grid-world environment used in experiments with the policy of the best solution found using (a) a flat, memory-less policy and (b) when using memory. The size of the arrows is proportional to the probability of choosing the action to move in that direction in a given state ($\pi(s, a) = p(a|s)$).

3 Grid-World Navigation Task

3.1 Environment

The environment used in this paper is a 2-dimensional grid-world as shown in Fig. 1. The set of possible states $\mathcal{S}$ comprises all unoccupied cells. The set of available actions $\mathcal{A}$ consists of four actions that move the agent one cell north, east, south or west. State transitions are fully deterministic: $\mathcal{P}^{a_t}_{s_t,s_{t+1}} \in \{0, 1\}$. When performing an action that would bring the agent to an occupied grid cell, the agent remains in the same state.

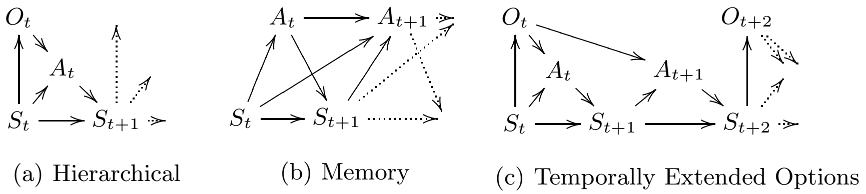
A single cell g is designated as the goal state and a reward is given when the agent arrives in this state: $\mathcal{R}^{a_t}_{s_t,s_{t+1}} = 1$ if $s_t \neq g, s_{t+1} = g$, and 0 otherwise.

3.2 Policy Evolution

We are interested in the minimum amount of state information an agent, equipped with different behaviour structures, takes in when following an optimal policy. We will denote the amount of memory intake as $\mathfrak{I}$, the minimum amount as $\mathfrak{I}^*$. When using a flat policy, the minimum is the relevant information as discussed in Sect. 2: $\mathfrak{I}^* = I(S; A^*)$, which can be determined computationally, e.g. by generalising the classical Blahut-Arimoto algorithm for rate-distortion[11, 16].

However, it is not trivial to adapt this algorithm to the other behaviour structures that we will introduce further on in this paper. Therefore, we will use an evolutionary approach to find policies that minimize $\mathfrak{I}$. We start with a population of random individuals, defined by the conditional probability distributions that make up their policies. At each epoch, individuals are iteratively selected from the previous population and combined by crossing their genotypes to create descendants, which advance to the next generation, possibly undergoing mutation.

The selection is done through tournament selection. A number of individuals is chosen at random and a fitness function $F(\pi)$ is used to rate their policy.



(a) Hierarchical (b) Memory (c) Temporally Extended Options

Fig. 2. Causal Bayesian network of the action-perception loop using (a) a hierarchical policy, (b) memory or (c) temporally extended options, unrolled in time

The winner is selected for breeding and produces a single offspring, of which a conditional probability distribution that is part of its policy can be crossed with that of the child of another tournament winner. The probability of occurrence of this crossing is determined by a cross-over rate $p_c \in [0, 1]$. Cross-over mixes distributions of the children such that $p_{child1}(A|B) \leftarrow p_{child2}(A|B)$ for $B \geq b$ where b is chosen at random. Mutation is applied by permuting a distribution $P(A = a|B = b) \leftarrow P(A = a|B = b) + \epsilon$ for each b with probability p_m, where a and ϵ are chosen at random. After mutation $P(A|B = b)$ is renormalized.

In all experiments described in this paper we have used a population size of 100 individuals and a tournament size of 3 individuals. The cross-over and mutation rates have been set to $p_c = p_m = 0.1$. The fitness function is defined as:

$$F = \mathfrak{I} + \beta \sum_{s,a} p(a|s)p(s)U(s,a) , \tag{3}$$

where $\mathfrak{I}$ is varied according to the experiments as described in the next sections. Furthermore, we set $\beta = 10^5$ to enforce near-optimal policies. Evolution is run until the change of fitness of the best individual between generations becomes sufficiently small. To establish a baseline we have applied the evolutionary algorithm using memory-less flat policies where $\mathfrak{I}_{flat} = I(S;A)$. The algorithm consistently converged to a solution giving $\mathfrak{I}^*_{flat} \approx 1.27$ bits, which is the same amount as found with the traditional Blahut-Arimoto algorithm. The policy of this solution is shown in Fig. 1(a).

4 Hierarchical Policies

In this section we will investigate the first intuition about the advantage of using a hierarchical policy: it splits up the task into simpler parts so it can be performed by an agent with simpler components. The minimum information intake $\mathfrak{I}^*$ gives a natural quantitative measure to determine difficulty of a (sub)task. Our hypothesis is that using a hierarchy reduces the amount of information required, and thus the necessary information processing capabilities, at each layer.

Here we will use the simplest hierarchical model with two levels. At each time step, a general decision is made at the higher level, based on the current state, which we will call an *option*, in accordance to the terminology of Sutton et al [15].

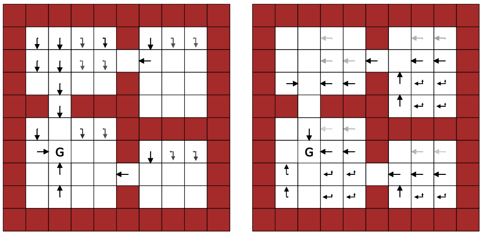


Fig. 3. Policy of the best solution found using a hierarchical, memory-less policy per option. The size of the arrows is proportional to the probability of choosing the action to move in that direction in a given state and option ($p(a|o,s)$). The opacity of the arrows in a cell are proportional to the probability of selecting the respective option in that state ($p(o|s)$).

At the lower level the actual action that is to be performed is selected based on this option o_t and some extra information about the current state. The policies at the higher and lower level are determined by the conditional probability distributions $p(o_t|s_t)$ and $p(a_t|s_t, o_t)$. The corresponding causal Bayesian network of this hierarchical policy is shown in Fig. 2(a).

The total information intake is the sum of the information intake at each level:

$$\mathfrak{I} = I(S_t; O_t) + I(S_t; A_t|O_t) = I(S_t; A_t) + I(S_t; O_t|A_t)\,. \tag{4}$$

As can be seen, this sum is greater than or equal to the mutual information between states and actions, with equality if either the current state or the chosen option is completely determined by the selected action. Therefore, the total information intake will never drop below the minimum achieved with a flat policy. It is no surprise then that we find that both the evolutionary approach as an adapted version of the Blahut-Arimoto algorithm find solutions where $I(S_t; O_t) = 0$ and $I(S_t; A_t|O_t) = I(S_t; A_t)$, effectively optimizing away the hierarchy.

Our goal is to show that using a hierarchy can let one get away with combining simple components. To do this we not only minimize the total sum, but also the information intake of the most complex part by extending the fitness function:

$$F = \mathfrak{I} + \max\left[I(S_t; O_t), I(S_t; A_t|O_t)\right] + \beta \sum_{s,a} p(a|s)p(s)U(s,a)\,. \tag{5}$$

In our experiments we set the number of options to 2. The best solution found by the evolutionary algorithm resulted in average information intakes of $I(S_t; O_t^*) \approx 0.83$ bits, $I(S_t; A_t^*|O_t^*) \approx 0.98$ bits, totalling to $\mathfrak{I}^*_{hier} \approx 1.81$ bits. Although the total system needs more information, the most complex level (the lower level) can get away with handling almost 25% less information as compared to a flat policy. The policy of this solution is shown in Fig. 3.

5 Memory

As mentioned in the introduction, the second intuition about hierarchical behaviour structures is that behaviours on higher levels function as a form of memory and retain information about the environment and the current task being performed. In this section we will investigate the effect of memory on necessary information intake of a policy. When looking at trajectories of states and actions, an agent can clearly get away with taking in less information than the sum of the single-step intakes discussed in the previous sections. Knowing about the past already gives you information about the current state without looking, since inherent structure of the world restricts the number of possible state transitions and the selection of an action is fed back into the distribution of possible states.

To handle channels that show this kind of feedback, the concept of *directed information* $I(S^T \rightarrow A^T)$ [17] was developed, which measures the actual information flow from an input sequence (S^T) to an output sequence (A^T) of length T. The directed information intuitively gives an indication of the amount of information an agent has to take in when having access to information about the past. In our experiments with memory we use this measure, scaled by trajectory length, as the average per step intake:

$$\mathfrak{I} = \frac{1}{T} I(S^T \rightarrow A^T) = \frac{1}{T} \sum_{t=0}^{T} I(S^t; A_t | A^{t-1})\,. \tag{6}$$

We use the simplest model, where $T = 2$ and thus memory only goes back a single time step, resulting in the network shown in Fig. 2(b). The best solution found in these experiments resulted in an average information intake of $\mathfrak{I}^*_{mem} = \frac{1}{2}\left(I(S_t; A^*_t) + I(S_t, S_{t+1}; A^*_{t+1}|A^*_t)\right) \approx \frac{1}{2}(1.32 + 1.08)\,\text{bits} = 1.2\,\text{bits}$. The policy of this solution is shown in Fig. 1(b). As expected, this average is lower than in the memory-less case. However, the average intake at the first step of a two-step trajectory ($I(S_t; A^*_t) \approx 1.32\,\text{bits}$) was consistently higher than $\mathfrak{I}^*_{flat}$, which means that the agent still needs a more complex information processing system.

6 Temporally Extended Actions

The separate properties of hierarchical behaviour discussed in the previous sections show important effects on the processing capabilities of an agent, however each still introduce more processing costs in some aspects. In this section we study the combination of both properties into a structure that corresponds better with the examples of nature given in the introduction and the notion of Sutton's options.

In these experiments we model a hierarchical policy that uses Temporally Extended (TE), abstract, actions: at higher levels in the hierarchy such an action (or option) is chosen as in Sect. 4, with the difference that this option is applied for several subsequent time steps. An action is chosen at the lowest level in the same way as in the memory-less case. The Bayesian network corresponding to the model used in our experiments, with a two-step cycle, is shown in Fig. 2(c). Note that this model is more limited than that of the previous section, since the choice for an option can only retain direct information about the previous state, information about the previous action is only implicit through the lower level policy ($p(a|o, s)$).

In this model the total per-step information intake is the average over the intake during both steps of a cycle:

$$\mathfrak{I} = \frac{1}{2}(I(S_t; O_t) + I(S_t; A_t|O_t) + I(S_{t+1}; A_{t+1}|O_t)) \,. \tag{7}$$

Again, we aim to minimize the costs of both the complete system and its seperate parts. Therefore, as in Eq. 5, we add the maximum intake of all parts $I(S_t; O_t)$, $I(S_t; A_t|O_t)$ and $I(S_{t+1}; A_{t+1}|O_t)$ to the fitness function.

Experiments with this structure in the grid-world described in the previous sections is still ongoing work at time of writing. Results obtained with a smaller, 2-room environment, however, show that temporally extended actions combine the advantages quantified in the previous two sections. One solution achieves the minimum necessary information intakes $I(S_t; O_t^*) \approx 0.62$ bits, $I(S_t; A_t^*|O_t^*) \approx 0.97$ bits, $I(S_{t+1}; A_{t+1}^*|O_t^*) \approx 0.36$ bits, giving $\mathfrak{I}_{TE}^* \approx 0.97$ bits, where with a flat, memory-less policy one has $\mathfrak{I}_{flat}^* \approx 1.03$ bits. This indicates that an agent can achieve an optimal policy with simpler information processing components (as in Sect. 4) and still on average take in less information in total (as in Sect. 5). Preliminary results suggest that these advantages generalize to more complex environments.

7 Discussion and Future Work

The results presented in the previous sections show quantitatively how hierarchical behaviour structures can reduce the necessary amount of information an agent needs to be able to process to perform a task optimally. We have also shown that simpler behaviours placed in a hierarchy can, with the same or even smaller amount of information, perform as well as a more complex, flat behaviour. This offers a novel, quantifiable, environment and architecture-independent argument for the use of these structures.

The methods put forward in this paper promise applications in the study of hierarchical behaviour in ethology, possibly resulting in a systematic understanding of the prevalence of this phenomenon in nature. Additionally, they can be applied to a wide variety of computational models and their specific implementations of behaviour structuring to determine the necessity of designer imposed assumptions and heuristics.

Future work will consist of studying how these results can guide optimal behaviour organization in organisms and artificial agents and to determine the optimality of existing models of organization. Additionally, we will study the effect of including more general notions of behaviour structuring, different models of memory and information flow [18] and further limitations and noise.

References

1. Prete, F.R., Wells, H., Wells, P.H., Hurd, L.E.E.: The Praying Mantids. Johns Hopkins University Press, Baltimore (1999)
2. Yu, A.C., Margoliash, D.: Temporal Hierarchical Control of Singing in Birds. Science 273(5283), 1871–1875 (1996)
3. McGonigle, B.O., Chalmers, M., Dickinson, A.: Concurrent disjoint and reciprocal classification by *Cebus apella* in serial ordering tasks: evidence for hierarchical organization. Animal Cognition 6, 185–197 (2003)
4. Dawkins, R.: Hierarchical organisation: A candidate principle for ethology. In: Growing Points in Ethology, pp. 7–54. Cambridge University Press, Cambridge (1976)
5. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (2007)
6. Barto, A.G., Mahadevan, S.: Recent Advances in Hierarchical Reinforcement Learning. Discrete Event Dynamic Systems 13(4), 341–379 (2003)
7. McGovern, A., Barto, A.G.: Automatic Discovery of Subgoals in Reinforcement Learning using Diverse Density. In: Proc. 18th International Conf. on Machine Learning, pp. 361–368. Morgan Kaufmann, San Francisco (2001)
8. Şimşek, O., Barto, A.G.: Using relative novelty to identify useful temporal abstractions in reinforcement learning. In: ICML 2004: Proceedings of the Twenty-First International Conference on Machine Learning, ACM, New York (2004)
9. Prescott, T.J., Redgrave, P., Gurney, K.: Layered Control Architectures in Robots and Vertebrates. Adaptive Behavior 7, 99–127 (1999)
10. Botvinick, M.M.: Hierarchical models of behavior and prefrontal function. Trends in Cognitive Science 12(5), 201–208 (2008)
11. Polani, D., Nehaniv, C., Martinetz, T., Kim, J.T.: Relevant Information in Optimized Persistence vs. Progeny Strategies. In: Proceedings of Artificial Life X. MIT Press, Cambridge (2006)
12. Capdepuy, P., Polani, D., Nehaniv, C.L.: Constructing the Basic Umwelt of Artificial Agents: An Information-Theoretic Approach. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 375–383. Springer, Heidelberg (2007)
13. Klyubin, A.S., Polani, D., Nehaniv, C.L.: All Else Being Equal Be Empowered. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 744–753. Springer, Heidelberg (2005)
14. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. In: Proc. 37th Annual Allerton Conference on Communication, Control and Computing, Illinois, Urbana-Champaign (1999)
15. Sutton, R.S., Precup, D., Singh, S.: Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artificial Intelligence 112(1-2), 181–211 (1999)
16. Blahut, R.E.: Computation of Channel Capacity and Rate-Distortion Functions. IEEE Transactions on Information Theory 18, 460–473 (1972)
17. Massey, J.L.: Causality, Feedback and Directed Information. In: Proceedings of the International Symposium on Information Theory and Its Applications (ISITA 1990), pp. 303–305 (1990)
18. Ay, N., Polani, D.: Information Flows in Causal Networks. Advances in Complex Systems 11(1), 17–41 (2008)

The Common Stomach as a Center of Information Sharing for Nest Construction

Istvan Karsai and Andrew Runciman

Department of Biological Sciences, East Tennessee State University,
Box 70703 Johnson City, TN 37614, USA
karsai@etsu.edu

Abstract. Construction of wasp nests is a self organized process that requires building materials, pulp and water foragers, and builders to cooperate. In this paper we study how the society of agents use a social crop, or common stomach, to store water that also provides a mechanism for worker connectivity, which in turn regulates building. Our model predicts that via the common stomach usage, medium sized colonies enjoy the benefit of having highly effective foragers and this in turn means that the colonies need only endanger a few foragers to ensure steady construction. When pulp foraging becomes more costly than water foraging, the colonies adjust via recruiting more pulp foragers and less water foragers, but keep high numbers of common stomach wasps on the nest. The common stomach provides an adaptable platform for indirect worker connectivity and a buffer for water storage.

Keywords: communication, swarm, social insect, superorganism, agent.

1 Introduction

Insect societies can be conceived as superorganisms in which inter-individual conflict for reproductive privilege is largely reduced and the worker caste is selected to maximize colony efficiency [1]. The ability of social insects to divide the colony's work via specialization, polyethism, and task partitioning has fascinated scientists for centuries. Many of these studies are commonly concerned with the integration of the individual worker behavior into colony-level task organization and with the question of how regulation of division of labor may contribute to colony efficiency. For example, bee colonies collect food from flowers, whose abundance varies strongly in space and time. Finding these resources requires considerable search effort [2] and effective foraging by the colony depends on an appropriate allocation of bee workers to exploration versus exploitation (collection of food from known resources) [3].

Colony-level flexibility in response to environmental changes and internal perturbations is an essential feature of division of labor [4, 5]. The actual number of workers allocated to different tasks is a result of the individual decisions made by these workers, which in turn are based on information available to these individuals through direct and indirect communication with their nest-mates and the environment. The collective colony-level pattern is therefore self-organized, without a central leader or

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 350–357, 2011.

template directing individuals to particular resources. Colony size correlates with productivity, body size, behavioral flexibility and colony organization [6].

Studies on a diverse array of social insect taxa show that interactions among workers (called worker connectivity) often play important roles in structuring division of labor [7]. O'Donnell and Bulova [8] propose that relying on shared and connected information can be beneficial: 1. Connectivity permits sharing of information among more workers and across greater distances than direct perception of stimuli. 2. Connectivity can foster task switching or can overcome task inertia. 3. "Catalytic individuals" with better or more information can propagate the information through the connected colony faster. The possible mechanisms of worker connectivity range from simple pair-wise encounters [9, 10] to specialized communicative displays [11].

While pheromones and dances are well-developed communication systems, even bees and ants use a wide variety of other types of communication to regulate or fine tune their division of labor [12]. For example, Cassill and Tschinkel [13] found that the division of labor in *S. invicta* ants depends on worker size and age and is fine tuned by the states of their crop volume and content. In honeybees, food reserves not only ensure homeostasis, but also regulate division of labor [14]. In social wasps, construction behavior is regulated indirectly by the temporally stored water in the crop of the insects [15, 16].

In this paper our goal is not to construct a detailed model the behavior of wasp societies, but rather to investigate in a more abstract way some important features of the common stomach. While our model is inspired by the colony regulation of wasps, the presented model is minimalistic; our agents are much simpler than the wasps, and we focus on the function of the common stomach rather than on the dynamically evolving agents. Specifically, we investigate how colony efficiency changes as a function of colony size and the constitution of task force. We also demonstrate how colony efficiency changes with the increase of the time cost of pulp foraging.

2 The Model

We constructed a multi-agent model (implemented in Java) inspired by the nest construction of Metapolybia wasps [15]. Nests are built by builder wasps using wooden pulp. For collecting pulp, the colony needs water and pulp foragers, and for the water the colony needs water foragers. For simplicity, we assumed that each agent belongs to a predetermined task group and her job does not change. Our goal was to focus on short term efficiency and to study the efficiency of different workforce combinations. For a more complex approach of the same phenomena see [16].

Collection of water and pulp take place outside the nest at the water and pulp source respectively (Fig. 1). The time required for collecting these materials is parameterized with Tw (water) and Tp (pulp) which represent the time in which the forager wasp is outside the nest collecting materials. We assumed that there is no variation in collection times or the amount of water and pulp collected. Collection of pulp generally takes longer [15] and we explicitly studied the effect of collecting time and the effectiveness of different colony compositions. Construction of the nest is simplified into a sink and the shape and the size of the nest do not change or play any role in this model. The pulp is considered to build into the nest by builder wasps that are working outside the interaction platform (therefore they were not modeled explicitly). Upon arrival the pulp forager unloads the pulp (into a sink that represents nest building) and starts to collect water on the interaction platform in the next turn. This simplification is in agreement with the usual operation of wasp colonies where individuals that are willing to build are generally abundant and they ready to accept the pulp promptly from the pulp forager [15].

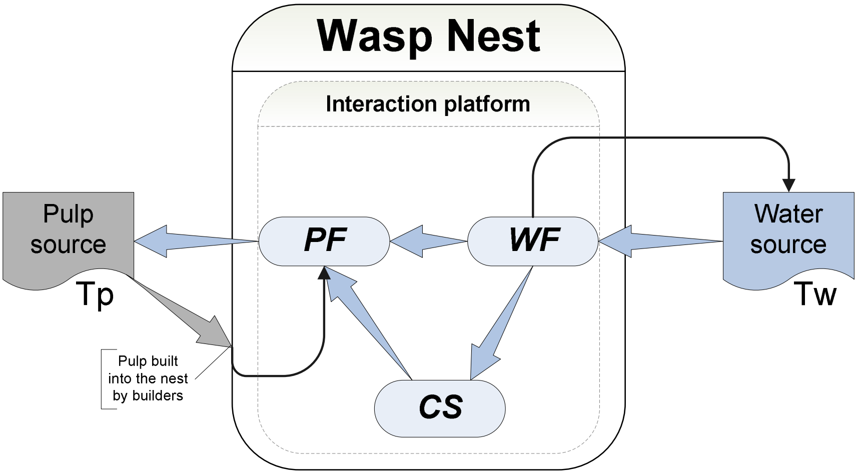


Fig. 1. Schematic representation of the wasp nest. Wasp types: water forager (*WF*); pulp forager (*PF*); common stomach wasp (*CS*). Common stomach wasps are able to receive water when empty and able to give water when they are full. Flow of the water (*blue arrows*), pulp transported from pulp source to the nest (*gray arrow*). Pulp is given to builders (not modeled explicitly). Transition of behavior of unloaded foragers for the next time step (*solid arrows*).

The simplifications above permitted us to focus on the water exchange among individuals as the main focus of this study. There are three types of agents on the interaction platform (Fig. 2):

- Common stomach wasps are either empty or filled with water. When they are empty they accept water from a water forager, and when they are full they give water to a pulp forager. However, they do not exchange water with each other.
- Pulp foragers attempt to collect water from a common stomach wasp or from a water forager. After this happens, they leave the nest for Tp time for collecting pulp. The pulp foragers use up all their water for the pulp collection (i.e., they leave with water and return only with pulp).
- Water foragers attempt to download their water load into a pulp forager or a common stomach wasp. After this happens, they leave the nest for Tw time for collecting water.

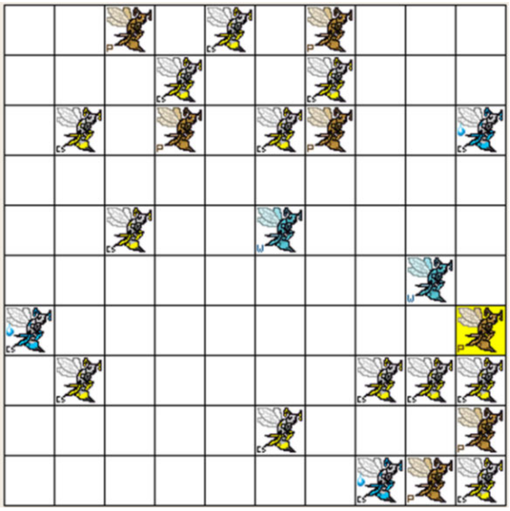


Fig. 2. Interactions of agents on the interaction platform. Wasps: water forager (*W: blue color*); pulp forager (*P: orange color*); common stomach wasp (*CS: blue with blue dot: holding water; yellow: empty*). Currently active pulp forager (*yellow background*) are able to receive water from a W agent in its Moore neighborhood. The two CS agent in its neighborhood are empty and cannot interact with this pulp forager in this turn.

In each time step, the agents attempt to land on the interaction platform randomly and interact with a single wasp in a Moore neighborhood (Fig. 2). The agent in focus examines how many potential cooperative agents are in the neighbor cells and randomly chooses one to interact with. Both foragers and common stomach wasps are considered a potential partner in the same way. For example, both a water forager and a common stomach wasp that hold water can give water to the pulp forager wasp. The rules of interaction are described as simple material transfer: if the states of the interaction are matching (one giver and one receiver) then material transfer happens. If no interaction is possible, the agent retains its behavioral state and makes a random landing again on the interaction platform in the next turn. This simplified routine is close to what we can observe in real wasp colonies during a 10 second long time interval: the wasp either makes an interaction with a neighbor and material transfer happens or she moves around on the interaction platform [15]. The small size of the interaction platform and the speed the wasps travel between material exchanges allowed us to simplify the movement patterns to a series of random landings.

The number of delivered pulp load divided by the number of foragers (PF+WF) was used as a measure of efficiency in a given colony. We assumed that colonies which operate with fewer foragers and provide high pulp input are more effective, because it results in a high construction rate with a minimized risk of losing foragers to predation. Each simulation started with empty common stomach wasps and all foragers landed on the interaction platform with full load. To avoid the effect of this colony initiation, the first 100 time steps (about 20 complete foraging cycles) were discarded and only the pulp arrival of the next 100 time steps was measured. The average of twenty parallel runs for each colony combination was calculated and used.

3 Results

First, we compared two very distinct scenarios: colonies which only operate with foragers (CS:WF:PF=0:1:1) versus colonies where half of the colony members are common stomach wasps and half of the colony members are foragers (CS:WF:PF=2:1:1). With increasing colony size, the amount of pulp delivered to the nest is larger [17] because of the increasing number of pulp foragers. Colony efficiency (pulp delivered/number of foragers) is significantly higher (MW U test p<0.05, N=40) in case of a larger colony size in both scenarios (Fig. 3).

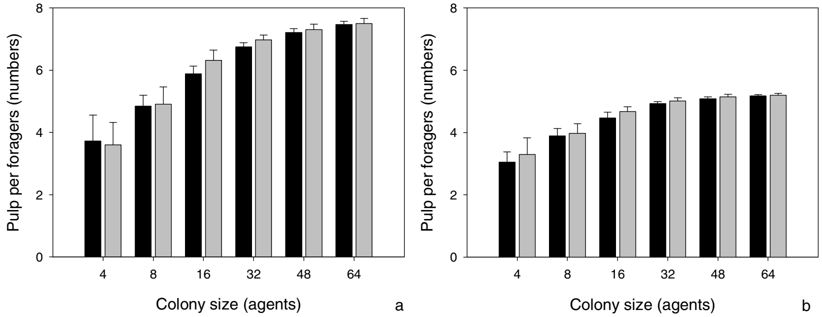


Fig. 3. Construction efficiency (total pulp returned/(WF+PF)) (*average and std.dev*) as a function of colony size. Workforce: 50% of the colony is common stomach wasp and 25- 25% are pulp and water foragers, respectively (*grey*); 50-50% of the colony are pulp and water foragers, respectively (*black*). Panel a: Tw=Tp=4; panel b: Tw=2, Tp=8.

When pulp foraging was 4 times more time consuming than water foraging (Fig 3, a vs.b), the efficiency of the colonies was significantly smaller (MW U test p<0.05, N=40) except in the smallest colony size where common stomach wasps were also present (MW U test p>0.1 (two tailed), N=40). Comparing the effectiveness of the 2 scenarios at each colony size showed significant differences (MW U test p<0.05, N=40) at colony sizes 16, 32 and 48, and the same pattern also emerged when pulp foraging was more time consuming. These indicated that using non-forager common stomach wasps can result in a more efficient solution for middle-sized colonies in these special mixes of workforces (Fig 3).

After studying two specific combinations of workforce along the wide range of population size above, we focused on two colony sizes (8 and 48), but generated all possible workforce combinations. In cases of small colonies when collection of pulp and water took the same time (TP=TW=4) the most effective workforce combination consists of 6 common stomach wasps, and 1 pulp and water forager, respectively.

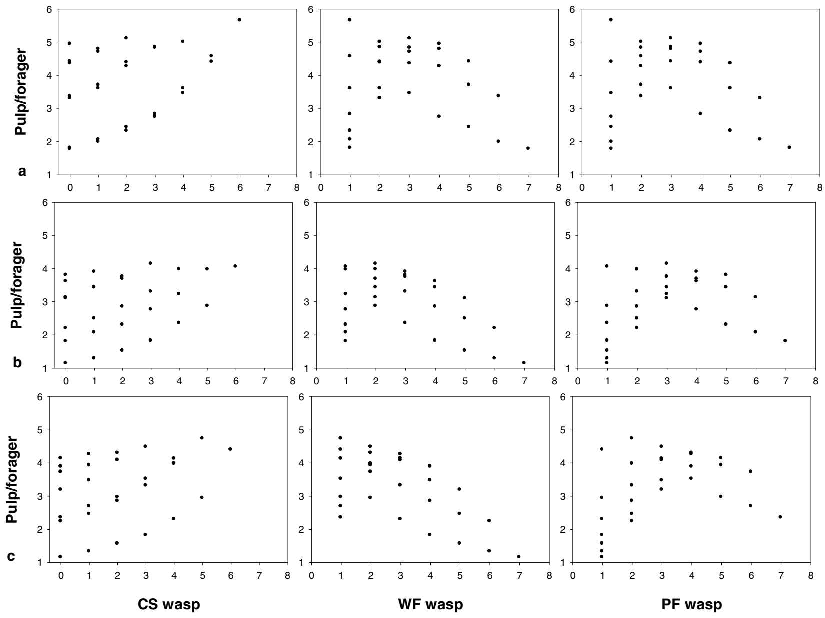


Fig. 4. Average efficiency (*dots*) of every possible workforce combination in small (8 individuals) colonies. Efficiency (pulp delivered/(WF+PF)) are calculated from 20 parallel runs of the same type of colony mix; a: TP=TW=4; b: TW=4, TP=8, c: TW=2, TP=8.

The increase of the number of foragers beyond 2-3 foragers clearly decreased the efficiency. On the other hand, as the number of common stomach wasps increased the efficiency (especially its lower bound) increased. Making pulp foraging more costly than water foraging (Fig. 4. b, c) resulted in a similar picture with a slight preference for more foragers. When pulp foraging was very costly, the most effective mix consisted of 5 common stomach wasps, 2 pulp foragers and 1 water forager (Fig 4 c).

In the larger colonies (colony size 48), the main trends were similar to the observed patterns of small colonies, but due to the larger workforce, there were a larger number of highly effective combinations (Fig. 5). The most effective colonies consist of a few foragers of both types and many common stomach wasps, but there were other combinations with less common stomach wasps and more foragers that provided only slightly less efficiency. When foraging for the two resources required the same time (Tw=Tp=4), then if either forager types dominated in numbers (more than half of the colony belonged to that forager types), the effectiveness of the colony is dropped sharply. As the time cost of pulp foraging increased compared to water foraging, the effective colonies had fewer water foragers, but they operated with high range (5-35) of pulp foragers (Fig 5. c). However, even in this case the most effective colonies are operated by small number of foragers and large number of common stomach wasps.

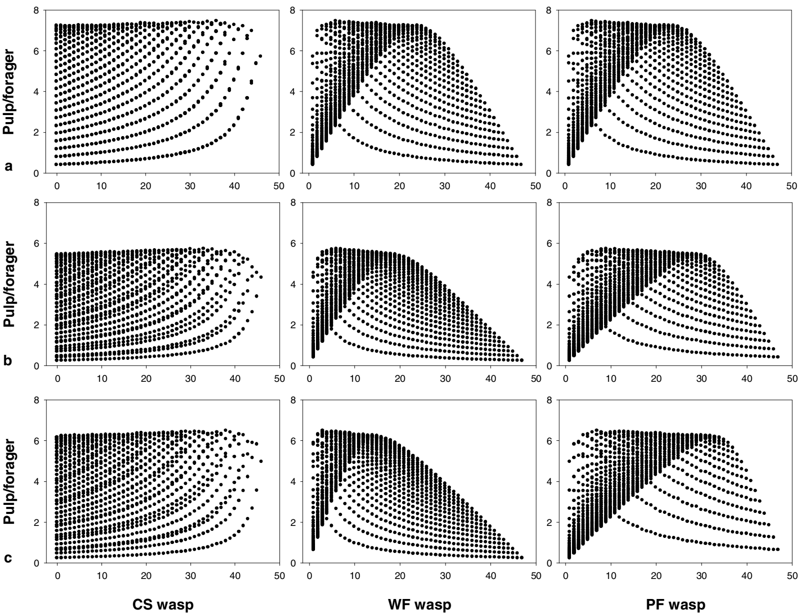


Fig. 5. Average efficiency (*dots*) of every possible workforce combination in large (48 individuals) colonies. Efficiency (pulp delivered/(WF+PF)) are calculated from 20 parallel runs of the same type of colony mix; a: TP=TW=4; b: TW=4, TP=8, c: TW=2, TP=8.

4 Conclusions

Our model predicted that the effective and low risk use of worker force via worker connectivity (common stomach) is affected by both colony size and the time required for retrieving the resources. We showed that the usage of the common stomach was beneficial in most cases, except if the density of the wasps is very low (hard to find partners) or very high (easy to find a partner for direct transmission). The size of the interaction platform could be a consequence of evolutionary pressures that prefer to keep most wasps on the nest. The regulation mechanism we presented is also able to adapt to the changing cost of resource collection via adjusting the number of foragers, but still retaining high number of common stomach wasp to ensure highly effective low-risk foraging. Using the common stomach as a regulator and buffer also provides secondary advantages in the form of additional work that these common stomach wasps can provide while they hold water, such as patrolling, defense and so on. Keeping the number of foragers low is beneficial, because foraging is dangerous [18]. By using the common stomach, only a few foragers need to take up the risky trips to the resources, and these individuals will be highly effective due to experience gained by the frequent trips [19]. Using the common stomach or social crop seems to be an efficient self-organizing mechanism for regulation of work of agents.

Acknowledgments

This work was supported by the Department BISC and the Honors College at ETSU.

References

1. Holldobler, B., Wilson, E.O.: The Superorganism. The Beauty, Elegance and Strangeness of insect Societies. W. W. Norton & Company, New York (2008)
2. Seeley, T.D.: Honeybee ecology. A study of adaptation in social life. Princeton University Press, Princeton (1985)
3. Bartholdi, J.J., Seeley, T.D., Tovey, C.A., Vande Vate, J.H.: The Pattern and Effectiveness of Forager Allocation Among Flower Patches by Honey Bee Colonies. J. Theor. Biol. 160, 23–40 (1993)
4. Calabi, P.: Behavioral flexibility in Hymenoptera: a re-examination of the concept of caste. In: Trager, J.C. (ed.) Advances in Myrmecology, pp. 237–258. Brill Press, Leiden (1988)
5. Robinson, G.E.: Regulation of division of labor in insect societies. Annu. Rev. Entomol. 37, 637–665 (1992)
6. Karsai, I., Wenzel, J.W.: Productivity, individual-level and colony-level flexibility, and organization of work as consequences of colony size. PNAS USA 95, 8665–8669 (1998)
7. O'Donnell, S.: Polybia wasp biting interactions recruit foragers following experimental worker removals. Anim. Behav. 71, 709–715 (2006)
8. O'Donnell, S., Bulova, S.J.: Worker connectivity: a review of the design of worker communication systems and their effects on task performance in insect societies. Ins. Soc. 54, 203–210 (2007)
9. Gordon, D.M., Paul, R.E.H., Thorpe, K.: What is the function of encounter patterns in ant colonies? Anim. Behav. 45, 83–100 (1993)
10. Pratt, S.C.: Quorum sensing by encounter rates in the ant Temnothorax albipennis. Behav. Ecol. 16, 488–496 (2005)
11. Sherman, G., Visscher, P.K.: Honeybee colonies achieve fitness through dancing. Nature 419, 920–922 (2002)
12. Dornhaus, A.: Finding optimal collective strategies using individual-bases simulations: colony organization in social insects. In: Troch, I., Breitenecker, F. (eds.) Proceedings MATHMOD 2009, Vienna, pp. 888–894 (2009)
13. Cassill, D.L., Tschinkel, W.R.: Task selection by workers of the fire ant Solenopsis invicta. Behav. Ecol. Sociobiol. 45, 301–310 (1991)
14. Schmickl, T., Crailsheim, K.: Inner nest homeostasis in a changing environment with special emphasis on honey bee brood nursing and pollen supply. Apidologie 35, 249–263 (2004)
15. Karsai, I., Wenzel, J.W.: Organization and regulation of nest construction behavior in Metapolybia wasps. Journal of Insect Behavior 13, 111–140 (2000)
16. Karsai, I., Balazsi, G.: Organization of work via a natural substance: regulation of nest construction in social wasps. J. Theor. Biol. 218, 549–565 (2002)
17. Karsai, I., Runciman, A.: The effectiveness of the "common stomach" in the regulation of behavior of the swarm. In: Troch, I., Breitenecker, F. (eds.) Proceedings MATHMOD 2009, Vienna, pp. 851–857 (2009)
18. Sakagami, S.F., Fukuda, H.: Life tables for worker honeybees. Res. Popul. Ecol. 10, 127–139 (1968)
19. Jeanne, R.L.: The organization of work in Polybia occidentalis: costs and benefits of specialization in a social wasp. Behav. Ecol. Sociobiol. 191, 333–341 (1986)

Economics of Specialization in Honeybees
A Multi-agent Simulation Study of Honeybees*

Thomas Schmickl and Karl Crailsheim

Artificial Life Laboratory of the Department of Zoology,
Karl-Franzens University Graz, Universitätsplatz 2,
A-8010 Graz, Austria
thomas.schmickl@uni-graz.at

Abstract. Honeybees are able to share the workload dynamically among the colony members. This division of labor is flexible and robust and is not controlled by a central unit. Thus, it is a 'swarm intelligent' feature. Several models of proximate mechanisms have been proposed which aim to explain how single workers decide on which task they work. We elaborated on an existing model of a honeybee colony which predicts the flow of workforce, information, and nutrients. We tested several models of proximate mechanisms and predicted colony-level fitness parameters: brood survival and net nectar gain. We found significant differences in the impact of specific proximate models on ultimate observables which describe colony fitness. Thus, our model could serve as a tool to predict benefits and costs of these mechanisms in honeybees. It contributes to the discussion of their potential evolutionary background.

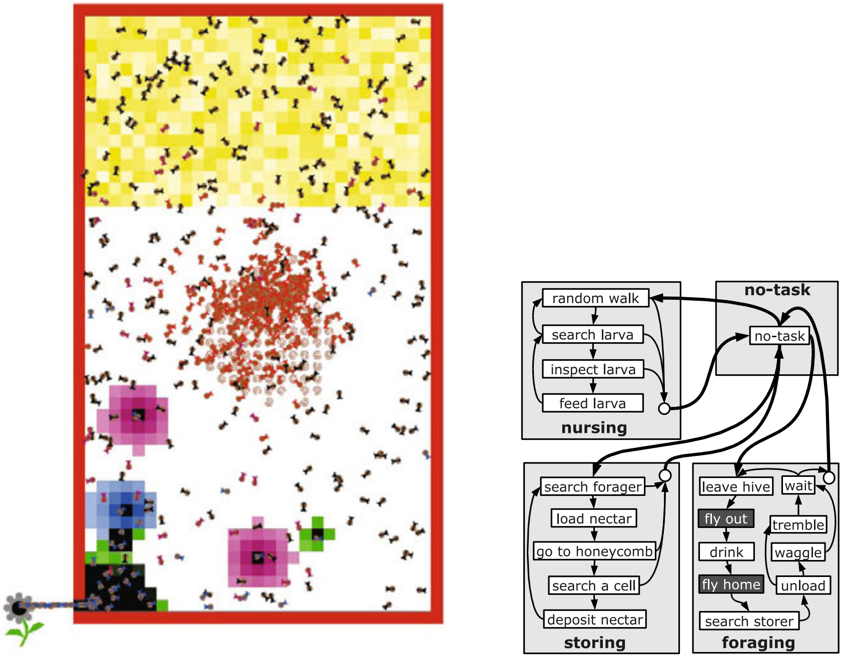
1 Introduction

A honeybee colony consists of several tens of thousand workers which collectively regulate their colony by forming a complex super-organism. A network of feedback loops arises from worker-to-worker interaction, which regulates the homeostasis of this super organism with an impressive degree of precision, robustness and flexibility [1,2]. This control is achieved in a 'swarm intelligent' way [3].

One prominent issue of homeostasis found in many social insects is division of labor (DOL): The colony work is not distributed at random. In contrast, it is observed that workers (workforce) split up into sub-groups by preferentially focusing on specific aspects of work (tasks). At a low degree, this phenomenon is just a slight preference of work groups to engage in specific tasks, which might change over the life time. At the highest degree, it is found as life-long specialization of workers to their tasks. It was observed that specialization is in many cases not rigid. Even in ant species which show high degrees of morphological

* This work was supported by: EU-IST FET project ISwarm, no. 507006; EU-IST-FET project SYMBRION, no. 216342; EU-ICT project REPLICATOR, no. 216240. Austrian Science Fund (FWF) research grants: P15961-B06 and P19478-B16.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 358–366, 2011.



(a) Screenshot of simulation

(b) State automaton

Fig. 1. (a) Screen-shot of a typical situation in our multi-agent simulation 'TaskSelSim': Foragers fly to and from a flower patch and search for storage bees after they have returned to the hive. These storer bees take over the nectar and head to the upper honey comb to store the nectar there. Nurse bees feed nectar to brood in the central part of the hive. Bees are color-coded on screen and nectar cells are brighter the fuller they are. (b) Hierarchical state automaton used in our model: Big boxes indicate tasks an agent can get involved in. Inside these tasks, a set of 'activities' form another automaton which reflects the behavioral program that is performed by agents in this specific task. Bold arrows indicate transitions between tasks, thin arrows indicate transitions between activities. Darker grayed boxes indicate activities that are associated with a higher nectar consumption rate (flying agents).

adaptation to the jobs they are usually specialized in, workers can switch to other jobs in cases of colony disturbances (e.g., artificial removal of a group of specialist workers, [4]).

In honeybees, DOL among workers is not achieved by strong morphological polymorphism, as it is the case in many ant species and termites. Worker bees tend to sequentially specialize on tasks following an age-based scheme, which correlates with physiological changes of the bees, favoring the efficiency of those tasks that are usually performed at the specific age. Younger bees first clean combs, then they nurse the brood. Middle-aged bees receive, handle and store the nectar and build combs. Older bees specialize on foraging for pollen, nectar, propolis and water. The degree of flexibility of workers within this age polyethism regime is still a controversial issue: Older studies emphasis the role of age [5]. In the last decades, worker reversion and flexibility was also demonstrated in honey bees [6]. Studies showed that workers progress more quickly to foraging after brood removal and that foragers revert to nursing after nurse removal. Recent studies suggest, that flexibility is much bigger within jobs that do not need physiological alteration: Developing specific glands or muscles takes time, thus these studies suggest that worker flexibility is over-emphasized in honeybees (e.g., see [7]).

2 Proximate Mechanism Models of DOL in Social Insects

Several models have been proposed for social insects with the aim to investigate proximate mechanisms of self-organized DOL [8]. Social insects are a heterogeneous group, thus it is unlikely that one of these proposed models holds for all species or even for all aspects found in one species. Some models are based on information [9] or on social inhibition [10]. Others rely on exchange of substances [11]. The most frequently discussed models are based on behavioral thresholds that have to be met by local stimuli: To trigger engagement bee i to task j, a stimulus $s_{i,t}$ at time t has to exceed a certain behavioral threshold $\theta_{i,j}$. The probability to engage in this task is frequently modeled by equations similar to

$$p_{i,j,t} = \frac{s_{i,t}^2}{s_{i,t}^2 + \theta_{i,j}^2}. \tag{1}$$

Some of these models work with fixed thresholds that do not change over time [12]. Other models discuss specialization to specific tasks by implementing an additional mechanism [13], which was extended by [14]. As expressed in in Eq. 1, an increase of $\theta_{i,j}$ decreases the probability of agent i to engage in task j. This fact is exploited by 'specialization models' in the following way: Whenever an agent (bee) i engages into task j, the associated threshold $\theta_{i,j}$ is decreased by ξ_j, thus this agent will be more likely to engage in the given task j again. In contrast, the threshold $\theta_{i,m}$ of a tasks m, which is not performed by agent i, will be increased by a constant value ψ_j, what decreases the probability to choose task m later on. In both models, each task is quit with a defined probability per time step l_j.

This article's main aim is to implement some of these discussed mechanisms in our existing multi-agent model of a honeybee colony (TaskSelSim, see Fig. 1) and to investigate how these mechanisms affect predicted DOL and predicted colony-level fitness parameters (net nectar gain, survival of brood). We do not suggest a new threshold reinforcement model. In contrast, this study compares efficiency and robsutness of models already proposed in literature.

3 Specific Honeybee Model

In addition to the mechanisms of task selection mentioned above, our TaskSelSim model implements several significant aspects of honeybee biology in detail. The model is described in [15], here we give those details that are necessary to understand the experiments described in this article:

The model depicts a colony having an entrance/exit in the lower left. The central area contains the brood nest, where immobile larvae wait to be fed by nurse bees. The upper hive area contains the 'honey comb' where nectar is stored by storage bees. All agents (bees) carry a specific energy reserve in form of nectar in their crop (see Fig. 1). Every activity decreases this nectar load at a different extent: immobile larvae consume less nectar than moving adults and flying adults consume more nectar per step than walking adults. If an agent (brood or adult) runs out of nectar, it dies and is removed from the system. Adults refill at the flower patch or at the honey comb, hungry larvae emit a chemical signal, which attracts nurse bees and which are able to switch unemployed bees to the nursing task.

Homecoming filled foraging bees search empty storage bees for transferring their nectar load to them. This is achieved by emitting a short-range 'unloading signal'. Filled storer bees head towards the honey comb, unload the nectar into a cell and head again towards the entrance area to unload the next forager bee. Depending on whether or not the forager searched long for a storer bee, the forager performs either a 'waggle dance' or a 'tremble dance'. Both dances have only a limited range (2-3 bee lengths) and disappear imideatly from the environment after the dancer stopped. The first kind of dance recruits new forager bees, the latter recruits new storer bees. This system of two conflicting dances was found to be a main mechanism to regulate the forager group size according to the current size of the storer group [16]. Nurse bees that run low on nectar reload at the honey comb. Brood emits a chemical hunger stimulus when it runs low on nectar. This stimulus recruits nurse bees, diffuses in space and decays over time. It is additive (brood emits new stimulus as long as it is hungry) and lasts some time in the system after the brood is fed. Our multi-agent simulation has several important characteristics that distinguish it from other models in this field:

1. We carefully implemented the behavioral programs of the three tasks 'foraging', 'storing' and 'nursing' to reflect behaviors of honeybees (Fig. 1b).
2. We implemented several variants of threshold mechanisms in our model.
3. We modeled tactile stimuli (short-range, not additive, not persistent), dance stimuli (mid-range, not additive, not persistent) and chemical stimuli (long range, additive, persistent). See Fig. 1a.
4. We implemented an activity-specific energy consumption and the flow of nectar through the honeybee colony. This allows us to investigate the effects of several models of proximate mechanisms of task selection on the individual bee level and on the colony level (colony's nectar gain/loss).

These characteristics make our model unique and allow us to perform sophisticated analyses of models of proximate mechanism in honeybees. Our model reflects many details of honeybee ethology and ecology, thus predictions of our model should be of high significance for honeybees.

4 Simulation Experiments

To compare literature-suggested threshold models of DOL, we simulated our our multi-agent model with three characteristic distributions of behavioral thresholds ($\theta_{i,j}$), which did *not change over time*, see graphical representations of these parameter-sets '1Caste', '3Castes', and 'random' in Fig. 2. These distributions reflect 3 potential cases of threshold distributions within the worker population: (1) All bees have the same mean probability to engage in any task, thus they are generalists. (2) Bees belong to one of three distinct worker castes, each preferring one task, thus they are specialists. (3) Every bee is different from other bees. In addition, we performed one series of experiments, in which the thresholds were initially distributed randomly (uniformly) between 0 and 1. These thresholds *did change over time* according to the mechanism described in section 2. We set the threshold adaptation parameters to values of $\xi_{nursing} = \xi_{foraging} = \xi_{storing} = 0.1$ and $\phi_{nursing} = \phi_{foraging} = \phi_{storing} = 0.001$, which was found to be an rather insensitive (to colony performance) default setting [15]. The abandonment rate of agents was set to $l_{nursing} = l_{storing} = 0.005$ and $l_{foraging} = 0.001$. We started all simulations with 700 unemployed adults and 100 medium-fed larvae. Each run lasted 8000 time steps and was repeated 16 times.

To test our 3 non-adapting threshold models ('1Caste', '3Castes', 'random') and the self-reinforcement model ('dynamic'), we performed 3 types of simulation experiments: First, we kept the environment unchanged for 8000 time steps. Second, we increased the length of a foragers' flight cycle by 50% at $t = 4000$, what relaxes the workload of storer bees. And third, we removed 50% of all adult bees at $t = 4000$, what doubles suddenly the workload on worker bees.

Statistical analysis was done in R. To indicate differences in colony variables depending on the used threshold model, we performed one-way ANOVA. Between-group medians were compared with paired two-sided Wilcoxon tests ($\alpha = 0.05$) using Bonferroni correction.

5 Results

Fig. 3 shows, that in an undisturbed environment the performance of all three threshold distributions that do not adapt over time lead to a better colony efficiency in terms of net nectar gain and brood survival compared to the threshold reinforcement model (Wilcoxon test, $p < .05, N = 16$ per setting).

For a sudden increase of the flight cycle of foragers (Fig. 4), the threshold-reinforcement model led to a significantly lower nectar gain than the other models (Wilcoxon test, $p < .05, N = 16$ per setting). In terms of brood survival, the 'random' (non-adapting) and the 'dynamic' (self-reinforcement) model slightly

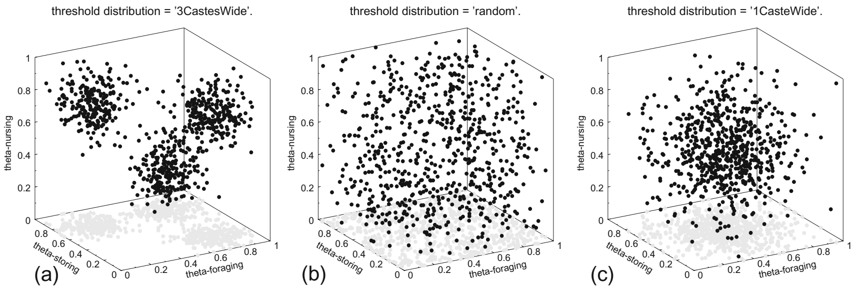


Fig. 2. Distribution of behavioral thresholds in our simulation experiments

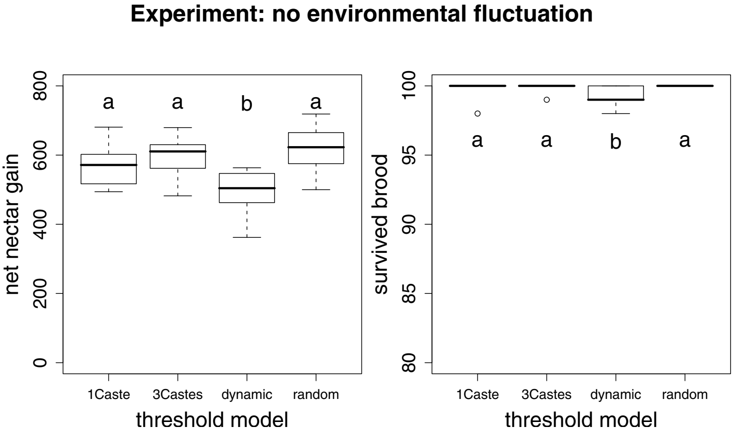


Fig. 3. Colony level fitness parameters as a result of the tested threshold models in an undisturbed environment. Graphs show medians and quartiles of the net nectar gain and the number of survived brood at $t = 8000$.

performed worse than the other two non-adapting models, which both have no extreme thresholds present in the population (Wilcoxon test, $p < .05, N = 16$ per setting).

When we removed 50% of the adult bees (Fig. 5), the largest differences between the used threshold model were observed: The self-reinforcement model ('dynamic') led to the lowest net nectar gain and to the smallest brood survival rate (Wilcoxon test, $p < .05, N = 16$ per setting). In addition, the model '3Castes' was less efficient compared to fixed random thresholds in terms of brood survival (Wilcoxon test, $p < .05, N = 16$ per setting).

6 Discussion and Outlook

The approach to use our existing model of the 'internal collective physiology' of a foraging honeybee colony showed to be useful. It allowed us to successfully

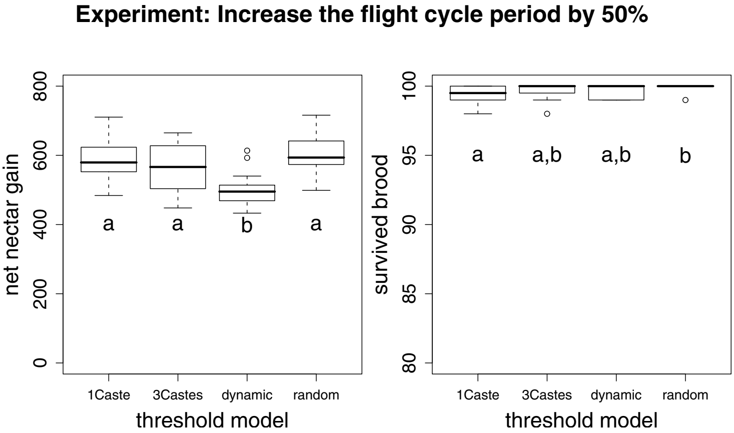


Fig. 4. Colony level fitness parameters as a result of the tested threshold models in a simulation run, in which the distance between flower patch and hive was increased by 50% at time step $t = 4000$. Graphs show medians and quartiles of the net nectar gain and the number of survived brood at $t = 8000$.

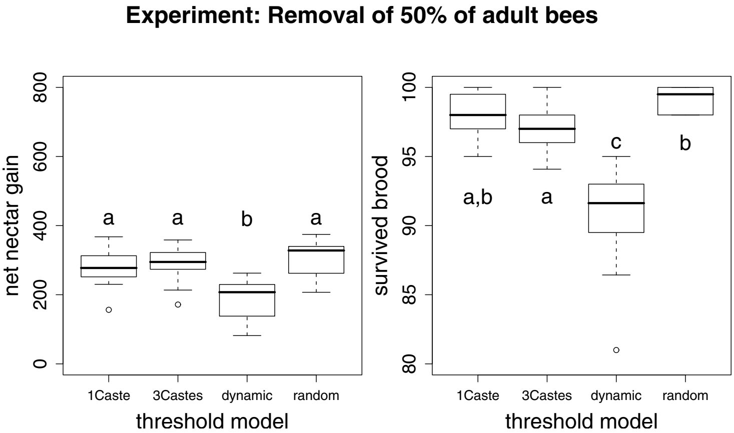


Fig. 5. Colony level fitness parameters as a result of the tested threshold models in a simulation run, in which 50% of the adult population was we moved at time step $t = 4000$. Graphs show medians and quartiles of the net nectar gain and the number of survived brood at $t = 8000$.

investigate the differences between two model paradigms that were suggested in literature to describe the proximate mechanisms governing task selection in social insects. We could show that in both models, the colony was able to deal with environmental fluctuations, but that the threshold-reinforcement mechanism is associated with the costs of a lowered net nectar gain and with lowered brood survival rate. This was shown for the modeled tasks in a honeybee model and must not be generalizable for all social insects. Differences between the two model paradigms were bigger in the simulation experiments with environmental fluctuation. When the colony reaches an equilibrium in worker groups, the reinforcement mechanisms drives θ values of tasks that are not performed by agents towards the maximum value ($\theta_{max} = 1$). This makes it harder for these agents to switch to these tasks after environmental fluctuations, even though such a switching might be favourable from the colony perspective (data not shown). When comparing the different fixed distributions of θ among each other, we found only few significant differences (see Fig. 4b and Fig. 5b).

In future, we plan to investigate more threshold distributions, more disturbances (e.g., adding and removing brood) in our model. We also plan to investigate the effect of the reinforcement parameters ξ and ϕ in more detail, as well as the parameters θ_{max} and l. It might prove to be useful to correlate the θ values with actual working efficiency parameters: For example. we plan to make predisposed foragers flying faster. Overall, the level of detail of our multi-agent simulation of a honeybee colony allows us to perform perform comparative studis of proximate mechanisms in bees. Such findings are a significant input to the discussion of the volutionary background of these mechanisms.

References

1. Seeley, T.D.: The wisdom of the hive. Harvard Univ. Press, Cambridge MS (1995)
2. Moritz, R.F., Southwick, E.E.: Bees As Superorganisms. An Evolutionary Reality. Springer, Berlin (1992)
3. Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Series in Evolutionary Computation. Morgan Kaufmann, San Francisco (2001)
4. Wilson, E.: The relation between caste ratios and division of labour in the ant genus pheidole (hymenoptera: Formicidae). Behavioral Ecology and Sociobiology 16, 89–98 (1984)
5. Lindauer, M.: Ein Beitrag zur Frage der Arbeitsteilung im Bienenstaat. Journal of Comparative Physiology A 34, 299–345 (1952)
6. Robinson, G., Page Jr., R., Strambi, C., Strambi, A.: Colony integration in honey bees: Mechanisms of behavioral reversion. Ethology 90, 336–348 (1992)
7. Johnson, B.: Limited flexibility in the temporal caste system of the honey bee. Behavioral Ecology and Sociobiology 58, 219–226 (2005)
8. Beshers, S., Fewell, J.: Models of division of labor in social insects. Annual Reviews of Entomology 46, 413–440 (2001)
9. Seeley, T.: The information center strategy of honeybees. Fortschritte der Zoologie 31, 75–90 (1985)
10. Naug, D., Gadagkar, R.: Flexible division of labor mediated by social interactions in an insect colony a simulation model. Journal of Theoretical Biology 197, 123–133 (1999)
11. Karsai, I., Balazsi, G.: Organization of work via a natural substance: regulation of nest construction in social wasps. Journal of Theoretical Biology 218, 549–565 (2002)
12. Bonabeau, E., Theraulaz, G., Deneubourg, J.L.: Fixed response thresholds and the regulation of division of labor in insect societies. Bulletin of Mathematical Biology 60, 753–807 (1998)

13. Gautrais, J., Theraulaz, G., Deneubourg, J.L., Anderson, C.: Emergent polyethism as a consequence of increased colony size in insect societies. Journal of Theoretical Biology 215, 363–373 (2001)
14. Merkle, D., Middendorf, M.: Dynamic polyethism and competition for tasks in threshold reinforcement models of social insects. Adaptive Behavior 12, 251–262 (2004)
15. Schmickl, T., Crailsheim, K.: Taskselsim: a model of the self-organization of the division of labour in honeybees. Mathematical and Computer Modelling of Dynamical Systems 14, 101–125 (2008)
16. Seeley, T.D.: Social foraging in honey bees: How nectar foragers assess their colony's nutritional status. Behavioral Ecology and Sociobiology 24, 181–199 (1989)

How Two Cooperating Robot Swarms Are Affected by Two Conflictive Aggregation Spots

Michael Bodi, Ronald Thenius, Thomas Schmickl, and Karl Crailsheim

Artificial Life Laboratory of the Department of Zoology,
Karl-Franzens University Graz, Universitätsplatz 2,
A-8010 Graz, Austria
{michael.bodi,ronald.thenius,thomas.schmickl,karl.crailsheim}@uni-graz.at

Abstract. Previous studies showed that two swarms of autonomous robots pursuing two conflicting goals can cooperate efficiently, especially at small swarm sizes. In this study we investigate how the spatial separation of the two conflictive aggregation spots affect the cooperation behaviour. The swarms are controlled by the BEECLUST algorithm, which is a robot control algorithm inspired by honeybee behaviour. We found that the spatial separation of the optima does not affect the aggregation efficiency of swarm sizes of 9 individuals or more. In contrast smaller cooperating swarms take advantage in their aggregation efficiency. Heterogeneous swarms are a big challenge in swarm robotics. When several tasks have to be achieved in parallel, swarms have to split up in task-related sub-swarms. Then efficiency enhancement by cooperation and the exploitation of side effects are a successful recipe for developing swarm intelligent algorithms.

1 Introduction

In social insects (e.g. honeybees) a huge number of individuals form a superorganism which shows self-organisation and swarm intelligent behaviour [1]. Even if all individuals exhibit "simple" behaviour, the swarm as a whole is able to solve complex challenges. Honeybees for example exploit rich foods sources more massively than poorer ones [2]. In the field of swarm robotics it is very important to keep individuals as simple as possible because resources are limited (e.g. memory or energy). For this reason social insects are a perfect source of inspiration for the field of swarm robotics [3] [4]. In swarm robotics the aggregation of agents is a very common goal but the approaches are very diverse. Dorigo et al. used an evolving neural network which consisted of 12 neurons for robot aggregation [5]. Other aggregation experiments were made with cockroach-like robots in simulation experiments as well as in real world, whereas an unique ID was required and communicated between the robots [6].

In this work we made experiments with robots controlled by the BEECLUST algorithm which is inspired by honeybee behaviour [7]. This algorithm consists of four simple rules (see Fig. 1):

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 367–374, 2011.

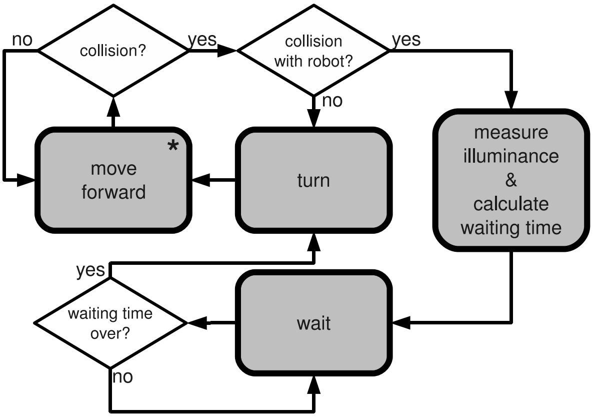


Fig. 1. Finite state machine of the BEECLUST algorithm. Boxes represent the different states of the robots. Diamonds represent if-else decisions. The asterisk (*) indicates the starting state of the controller.

1. The robots move straight forward through the arena. Whenever a robot detects an obstacle it checks whether the obstacle is a wall or another robot.
2. If this obstacle is a wall, the robot turns and continues with step 1.
3. If the obstacle is another robot, the robot measures the local illuminance and calculates a waiting time, depending on the illuminance.
4. When the waiting time is over, the robot turns and continues with step 1.

In [7] it has been shown that a swarm of Jasmine III robots, controlled by this algorithm, is able to find a spot of highest illuminance in an arena. A swarm of robots controlled by this algorithm responds dynamically on spontaneous environmental changes and satisfies all needs for being classified as swarm intelligent [1] [7]. The reasons for this intelligent behaviour are the feedback loops which emerge from within the swarm [8]. In [9] we showed that swarms of robots controlled by this algorithm act robust against disturbances induced by other swarms. Small swarms can even take advantage in their ability to aggregate if another swarm is present, even if the other swarm is performing a different task. Elaborating on this work, we wanted to find out if the spatial separation of two conflicting target sites for both swarms has an influence on the aggregation efficiency.

2 Methods

We performed our experiments in SMARS which is a simulation environment for experiments with Jasmine III robots, written in NetLogo [10]. We implemented two different swarms which act in parallel within the same environment: One swarm waits longer at places of high illuminance and is further called “light finders”. The other swarm waits longer at places of low illuminance and is called “shadow finders”. So the only difference between the two swarms is that a

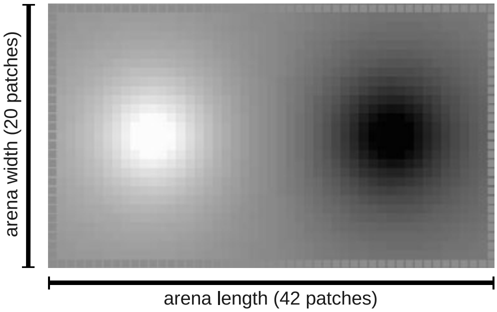


Fig. 2. Screenshot of an empty arena. The white area indicates the light spot, the black area indicates the shadow spot. The arena has the size of 42x20 patches, whereby a patch is a square with the side length of 3 cm.

different waiting time function was implemented (see Fig. 4) whereas the robots do not discriminate between the two swarms. The tested population sizes were 2, 3, 6, 9, 11, 18 and 24 robots per swarm. Each experiment was repeated six times. We implemented the following light distribution: The arena shows an ambient illuminance of 500 lux, one light spot of approx. 1000 lux and one shadow spot of approx. 0 lux. The arena has a size of 46 patches in length and 20 patches in width (see Fig. 2), whereby a patch is a square with a side length of 3 cm. In a first experiment we changed the distance between the light spot and the shadow spot in each run. These spots had a distance of 7, 9, 11 and 13 patches to the arena centre. All tested light distributions are shown in Fig. 3. In a second experiment we tested the response of the robot swarm on spontaneous changes in the environment. For this reason we started the experiments with the same setup as already mentioned above. After four minutes we swapped the positions of the light and the shadow spot. After four more minutes we swapped them again and monitored the following reaction of the robot swarm for four more minutes. In this experiment we compared runs with a distance of 7 patches between the optima and the arena centre to runs with a distance of 13 patches between the optima and the arena centre.

In our analysis we defined a target zone for the "light finders" which includes all patches, on which an illuminance between 600 and 1000 lux was present. This area covers 40% of the maximum light value in the arena. To analyse the aggregation quality we monitored the percentage of "light finders" within the state "wait" in the target zone during the last minute of every repetition. To analyse the aggregation speed of the swarm we monitored the point of time, in which 50% of the "light finders" were aggregated in the target zone (TA_{50}). Each run took 4 minutes. To analyse the swarm's respond on changes in environment, we monitored the "light finders" in the target zone during the whole experiments. For quantifying possible enhancement of aggregation (% of the total swarm) we defined the index ΔAL as

$$\Delta AL = AL_7 - AL_{13}. \qquad (1)$$

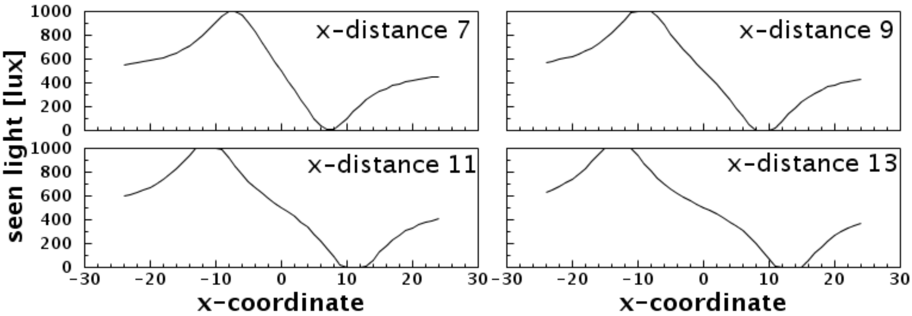


Fig. 3. Tested light distributions. The centres of the extreme spots were located in a distance of 7 (A), 9 (B), 11 (C) and 13 (D) patches from the arena centre.

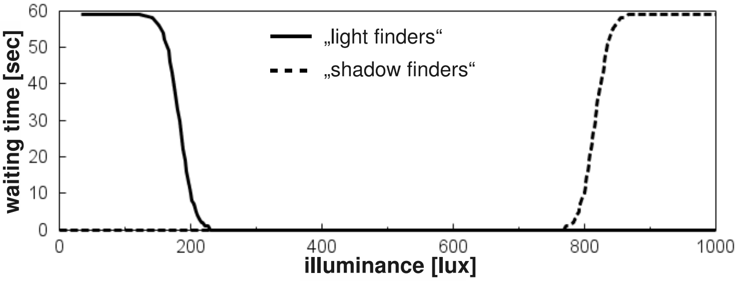


Fig. 4. Dependence of the waiting time on the local illuminance. The solid line shows the function which was implemented in the "light finders", the dashed line represents the same for the "shadow finders".

ΔAL represents the aggregation enhancement of the "light finders". AL_7 is the percentage of aggregated "light finders" when the optimum has a x-distance of 7 patches. AL_{13} is the percentage of aggregated "light finders" when the optimum has a x-distance of 13 patches from the arena centre.

3 Results and Discussion

In Fig. 5 we show that small swarms (2 and 3 individuals) show a lower aggregation quality than larger swarms. The distance between the two optima has an effect on small swarms by decreasing the fraction of aggregated robots. Larger swarms (9 individuals or more) are not influenced by the distance between the optima. In each case 70% to 80% of the swarm is aggregated under the light spot. Concerning the aggregation speed we found that larger swarms (9 individuals or more) are not affected by the distance between the optima (see Fig. 5). In each case it takes about 30 to 40 seconds to place 50% of the swarm under the light source. But contrary to aggregation quality, small swarms aggregate faster when the optima are close to each other.

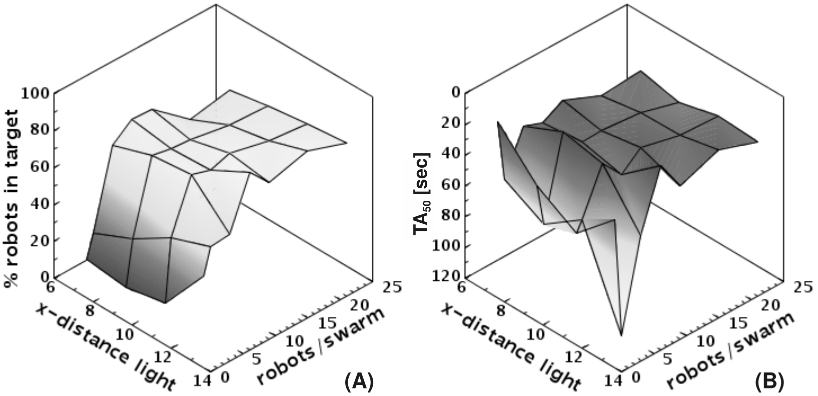


Fig. 5. (A) Percentage of "light finders" aggregated in the target zone within the last minute of the observation. This represents the aggregation quality. (B) Point of time when 50% "light finders" are aggregated in the target zone. This is an indicator for aggregation speed (n = 6 repetitions per experiment).

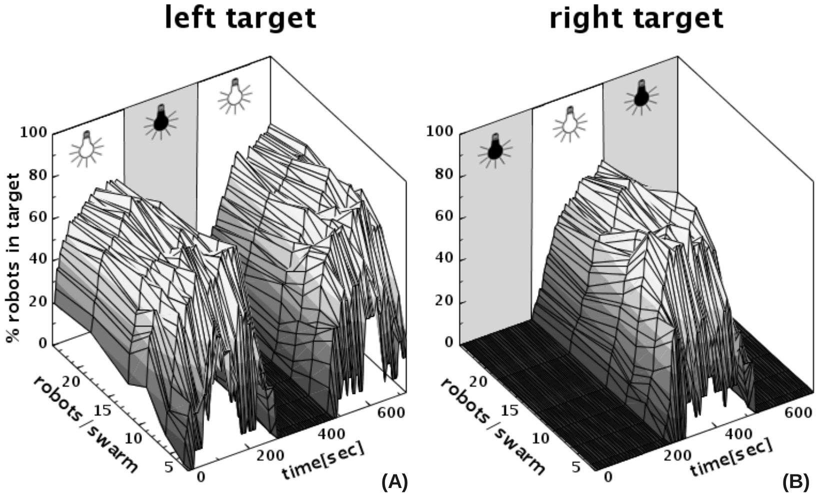


Fig. 6. Percentage of aggregated "light finders" in the target zone. The distance from an optimum to the arena centre (x-distance) is 13 patches. (A) Left target zone. (B) Right target zone.

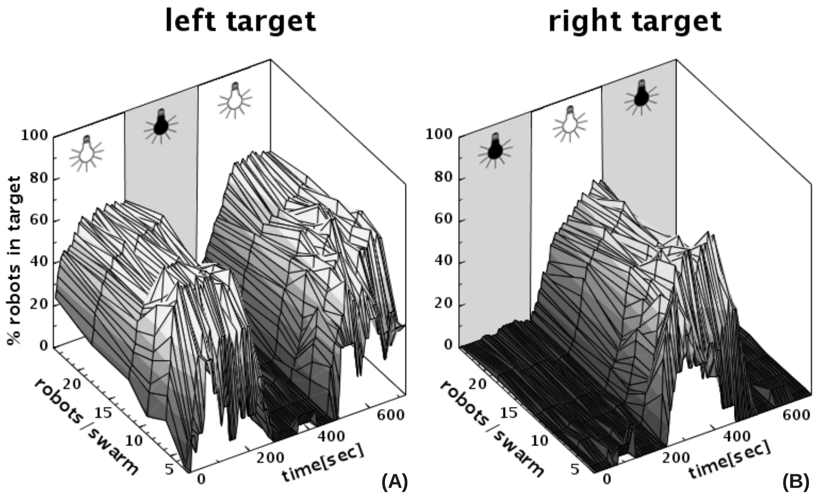


Fig. 7. Percentage of aggregated "light finders" in the target zone. The distance from an optimum to the arena centre (x-distance) is 7 patches. (A) Left target zone. (B) Right target zone.

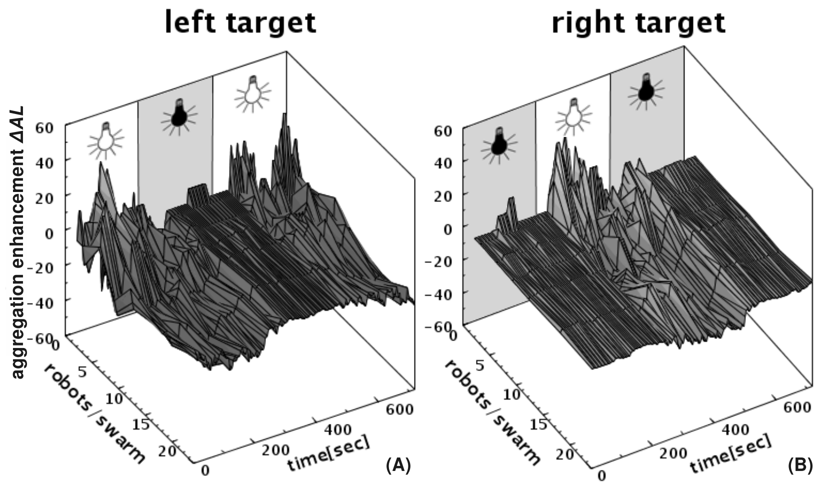


Fig. 8. Shown is the aggregation enhancement ΔAL in percent. For the formulation of ΔAL see equation 1. (A) Left target zone. (B) Right target zone. Please notice, that the y-axis (robots/swarm) is flipped for a more clearly representation in this the figure.

We show that the swarm reacts fast and reliable to spontaneous changes in the environment and follows the light spot in the arena when the two optima are far (x-distance = 13 patches) from each other (see Fig. 6) as well as when the two optima are close (x-distance = 7 patches) to each other (see Fig. 7). In both cases we found that swarms of 9 individuals or more show a robust aggregation efficiency: approx. 60% of the swarm aggregates when the optima are close to each other and 70% of the swarm aggregates when the optima are far from each other. But as it can bee seen in Fig. 7 small swarms (2 and 3 individuals) show a higher aggregation quality than larger swarms when the optima are close but the aggregation quality decreases when the two optima are far from each other (see Fig. 6).

In Fig. 8 we show that small swarms can achieve an aggregation enhancement of 30% after 100 seconds when the two targets are close to each other. After that the enhancement decreases again. This means that small swarm aggregate faster when the two optima are close to each other. The same effect can be seen after the swap of the two optima in second 240 in the right target and second 480 in the left target. This shows that small swarms can take advantage of close optima and are able to react more dynamically on spontaneous environmental changes. Larger swarms on the other hand are not significantly affected by the distance between the two targets.

Our results corroborate our presumption of the BEECLUST algorithm being a robust control algorithm for robot swarms. In Fig. 6 and Fig. 7 we show that the swarm is able to react to spontaneous changes in the environment. This corresponds well with reports in [7]. As it is shown in Fig. 5, the aggregation quality increases with the swarm size until an optimal robot density (9 individuals) is reached. This fits well to the results shown in [9]. Such a "critical minimum swarm density" is characteristic for swarm algorithms. The distance between the two optima affects small swarms (6 individuals and below) significantly in aggregation quality and speed. In contrast large swarms are affected only slightly. Small swarms do not only aggregate faster when the two optima are close to each other (see Fig. 5), they also react faster to environmental changes than larger swarms (see Fig. 8). The reason for better aggregation performance with close targets is that both swarms of aggregated robots build a kind of one big cluster which is shared by both swarms when the two optima are close to each other. This leads to a high number of robot-to-robot encounters near their optima. In large swarms, jamming effects induced by high robot density cancel this benefit out. Nevertheless large swarms show robust aggregation which is not affected significantly by the spatial separation of the optima.

4 Summary and Outlook

In summary, we say that the BEECLUST algorithm works very robust even with two cooperating swarms without discriminating the swarm affiliation. Larger swarms are not affected by the spatial separation of the optima, whereas smaller swarms gain benefit from optima which are close to each other. In future we will investigate whether or not the light gradient steepness in the arena (flat vs. steep vs. discrete steps) has an influence on the aggregation efficiency. Furthermore we will investigate whether or not the BEECLUST algorithm acts dynamically enough to follow moving light spots.

Acknowledgements

This work was supported by: EU-IST FET project I-Swarm, no. 507006; EU-IST-FET project SYMBRION, no. 216342; EU-ICT project REPLICATOR, no. 216240. Austrian Science Fund (FWF) research grants: P19478-B16.

References

1. Millonas, M.M.: Swarms, phase transitions, and collective intelligence. In: Langton, C.G. (ed.) Artificial Life III. Addison-Wesley, Reading (1994)
2. Seeley, T.D., Camazine, S., Sneyd, J.: Collective decision-making in honey bees: how colonies choose among nectar sources. Behavioral Ecology and Sociobiology 28(4), 277–290 (1991)
3. SYMBRION: Project website (2008), http://www.symbrion.eu/
4. REPLICATOR: Project website (2008), http://www.replicators.eu/
5. Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T.H., Baldassarre, G., Nolfi, S., Deneubourg, J.L., Mondada, F., Floreano, D., Gambardella, L.M.: Evolving self-organizing behaviors for a swarm-bot. Autonomous Robots 17(2-3), 223–245 (2004)
6. Garnier, S., Jost, C., Jeanson, R., Gautrais, J., Asadpour, M., Caprari, G., Theraulaz, G.: Collective decision-making by a group of cockroach-like robots. In: Proc. of Swarm Intelligence Symposium, SIS 2005, pp. 233–240. IEEE, Los Alamitos (2005)
7. Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., Crailsheim, K.: Get in touch: Cooperative decision making based on robot-to-robot collisions. Autonomous Agents and Multi-Agent Systems 18(1), 133–155 (2008)
8. Hamann, H., Wörn, H., Crailsheim, K., Schmickl, T.: Spatial macroscopic models of a bio-inspired robotic swarm algorithm. In: IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS 2008), pp. 1415–1420. IEEE Press, Los Alamitos (2008)
9. Bodi, M., Thenius, R., Schmickl, T., Crailsheim, K.: Robustness of two interacting robot swarms using the BEECLUST algorithm. In: MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling (2009)
10. Wilensky, U.: Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999)

A Minimalist Flocking Algorithm for Swarm Robots

Christoph Moeslinger, Thomas Schmickl, and Karl Crailsheim

Karl-Franzens University Graz, Artificial Life Lab of the Department of Zoology,
Universitätsplatz 2, 8010 Graz, Austria
{christoph.moeslinger,thomas.schmickl}@uni-graz.at

Abstract. In this paper we describe a low-end and easy to implement flocking algorithm which was developed for very simple swarm robots and which works without communication, memory or global information. By adapting traditional flocking algorithms and eliminating the need for communication, we created an algorithm with emergent flocking properties. We analyse its potential of aggregating an initially scattered robot swarm, which is not a trivial task for robots that only have local information.

1 Introduction

One of the most amazing developments in biological evolution is the domain of social insects. These animals, although being very small, achieve impressive feats. Bees, ants and termites live in elaborately constructed nests which are, in comparison to the insect, gigantic [1]. The colony super-organism can be characterized as being *swarm-intelligent* [2] because its abilities, for example to optimally allocate foragers to food sources, are a result of the interactions within the swarm and cannot be achieved by the single individual. This decentralized and distributed way of achieving a goal is an interesting and useful field of study which has inspired the fields of swarm-intelligence [3, 4] and swarm robotics [5]. In the last decade, a lot of control strategies and algorithms for robotic swarms have been presented, both in simulated and real robot swarms.

Innovation and industrial progress make it possible to manufacture such swarm robots in smaller and smaller sizes [6]. For such small robots, reaching a common goal is not an easy task. This is due to the constraints that come with decreasing size, like small sensor ranges, very limited computational power, little memory and imprecise locomotion. Thus, developing control algorithms for small scale swarm robots raises interesting questions, like "how can a robot, which only knows about the very small part of the environment around itself, achieve a common goal with the rest of the swarm?" and "given all constraints, what is the most efficient way to reach a common goal?".

Reaching such a common goal often implies that the swarm has to be aggregated for cooperative work, transport [7, 8] or assembly [9, 10]. This aggregation task seems to be relatively trivial, but the constraints of a swarm of small robots

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 375–382, 2011.

create some conceptual problems. Small robots usually do not have the capability of long range communication and do not possess global information like position and heading. When only local information is available, the coordination of numerous autonomous agents requires a different approach. Examples of such an approach already exist in nature: the phenomenon of flocks, herds and schools.

Flocks can be aggregations of up to several thousand individuals which move together with astounding elegance and flexibility. Craig Reynolds was amongst the first to abstract this behaviour to steer a swarm of simulated birds which he called *boids* [11]. To do this, he implemented three behavioural rules in his autonomous boids: *collision avoidance* to evade obstacles and flock mates which are too close, *flock centering* to stay close to flock mates and *velocity / heading matching* to move in the same direction as nearby flock mates. The resulting simulated flocks appear very similar to real flocks. As a result, several approaches to adapt this behaviour to a robotic swarm have been made. Although these pursues were successful in creating flocks, all of them incorporated something that is usually not found in real bird flocks or fish schools: communication and knowledge of the own heading [12–15] or predefined leaders [16]. Other interesting approaches used light beacons (and the resulting shadows) in an arena to generate a flocking swarm, either using evolved neural networks [17] or adapted aggregation algorithms in combination with *situated communication* [18], where the only significance is the presence or absence of an "empty" message. Usually, the matching of velocity and heading was only possible when the robots were able to communicate with their flock mates through a stable communication channel. This means that each robot needed to know its own heading (through a digital compass) and the headings and speeds of all its (near) flock mates.

This involves extensive communication, which can be far too complex for small and numerous swarm robots, and, more importantly, that is also not in accordance to how a real flock achieves alignment. In real swarms, the animals identify the heading of each other because they derive it visually from their flock mates' body form, although most fish species also have a specialized *lateral line organ* which is additionally used for alignment [19]. Such methods could be adapted for swarm robots by using on-board cameras and image recognition, but that would probably be way too complex for very small robots.

We found out that there is also a much easier way to create flocking behaviour. By discretizing the robots' sensor fields into sectors and using different distance thresholds for attraction and repulsion in these sectors, robot swarms can achieve *emergent* alignment.

2 Material and Methods

2.1 Algorithm Requirements

The main aim of our flocking algorithm is simplicity. This means that we developed our algorithm in consideration of very limited computational power and only minimalist swarm robot equipment. Such a basic equipment is a set of distance sensors, which are usually used for collision avoidance. We wanted to utilize just these sensors to generate complex swarm behaviours.

Swarm robots usually have IR-sensors in all directions which enable the detection of reflecting surfaces like walls, obstacles and other robots. A major disadvantage of these IR-sensors is, that they have very limited range and cannot discriminate robots from obstacles, except when using a combination of active and passive sensing. Here, active sensing means that the robot activates its IR-light at the position of the sensor and checks for reflections, whereas passive sensing means that the robot only checks for IR-light from other robots without emitting IR-light itself. Usually, the brighter the sensed IR-light, the higher is the value that the sensor returns. These IR-sensor values are all that is needed for the effectivity of our algorithm. Of course other distance measuring sensors, like ultrasonic sensors, can be used for our algorithm as well.

The minimal requirements of the algorithm are 4 circumferential distance sensors with limited range and 3 discrete reactions in movement: *move straight*, *turn left* or *turn right*. The algorithm does not require any global information about positions or headings, precise sensor information, memory, elaborate robot-to-robot recognition or communication.

2.2 Flocking Algorithm

Each robot in the swarm periodically emits IR-pulses. The robots then react (*move straight*, *turn left* or *turn right*) depending on information from their active and passive IR-sensors. These sensors are polled periodically and the returned values are then checked against predefined thresholds (Fig. 1B) in a simple subsumption architecture (Fig. 1A).

First, the active IR-value for the front sensor is polled to find out whether there is an obstacle in front. If the value for the reflected IR-light is above a certain threshold, the robot turns away in a random direction. This is the basic *collision avoidance* of our robots.

If there are no objects in its way, the robot checks the passive IR-values of all sensors. If the front, left or right sensor is above a certain threshold, the robot turns away from what is presumably another robot which is too close. This rule is usually referred to as the *separation rule* in flocking algorithms.

If there is no other robot too close, the robot checks the passive IR-values of its left, right and rear sensors. For every sensor that returns a value that is above the environmental IR-light threshold but below the threshold which defines the maximally desired distance to another robot in that sector, the robot performs a basic *vector addidtion* and adds up all turns. It then decides to turn in a direction depending on whether there were more left or more right turns. Robots in the rear zone trigger a random turn reaction. This rule is usually referred to as the *cohesion rule* in flocking algorithms.

The third rule in flocking algorithms is usually the *alignment rule* which generates the common direction of movement in a flock. Since we wanted our algorithm to be as simple as possible we wanted to exclude complex communication

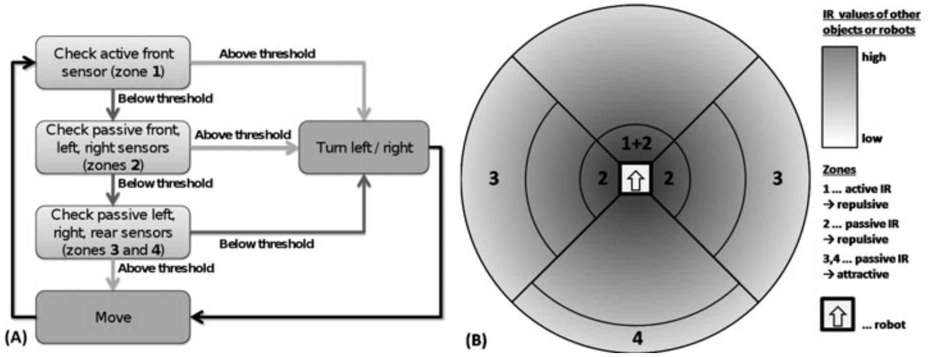


Fig. 1. A: Simple subsumption architecture depicting the flocking algorithm. The first decision results in *collision avoidance*, the second decision results in *robot separation* and the third decision results in *flock cohesion* and *emergent alignment*. B: Simplified depiction of the perceived IR-values of other objects (reflected active IR) or flock mates (passive IR) dependent on their distance to the robot. Thresholds and the resulting zones for a robot with 4 IR-sensors.

or image recognition procedures and implemented a method which generates *emergent* alignment. To achieve this we adjusted the thresholds for the *cohesion rule* so that robots tend to follow other robots. This is done by simply shifting the threshold for the rear sensor more outwards in comparison to the thresholds for the left and right sensors (zones 3 and 4 in Fig. 1B). Depending on the position and heading of two approaching robots, one robot will be behind the other robot. When both robots move, the robot behind will turn towards the robot in front *before* the robot in front reacts and turns around. This creates a *leader robot* and a *follower robot*, purely by chance. These two robots will then move around in the arena without separating. If the path of these two robots is blocked by an obstacle or another robot joins the flock, the arrangement can change instantly. If two robots approach frontally, they will avoid each other, only to turn back to each other shortly after, which can create a *deadlock* situation. To prevent such situations, we implemented a random-turn reaction which means that robots will randomly turn either left or right when avoiding other robots in front (zone 1 in Fig. 1B).

2.3 Simulator

We conducted simulations firstly as a *proof-of-concept* and secondly to evaluate the aggregation and flocking capability of a robot swarm which uses our algorithm. These simulations use very abstracted and ideal two-dimensional models of autonomous mobile agents with circumferential sensors. For our simulations we used a custom-built simulator (Fig. 2A) based on the multi-agent programming language NetLogo [20]. A short video can be seen at [21].

The tests were conducted with different numbers of robots in differently sized arenas. For generality, we assumed 1 robot-diameter as the unit of measurement. A simulated robot moved with 3 units per second and polled its distance-sensors, which had a maxmium range of 5 units, 60 times per second. The thresholds were set to 1 unit for zone 1, 1 unit for zones 2, 2 units for zones 3 and 3.5 units for zone 4. These values were derived from tests with real swarm robots [22]. To emulate real-world conditions and to avoid certain deadlocks (e.g., two robots circling each other) we introduced inaccuracies. In the simulations, the sensor values were subject to a 5% random-normal error in measurement and the speed of the individual robots was subject to a 5% random-normal variance. We simulated swarms of 5, 10, 15, 20 and 25 robots in bordered square arenas with the sizes of 3600, 7200, 10800, 14400 and 18000 robot-diameter^2. The sizes of the arenas were chosen to create density-neutral setups so that we could measure the impact of swarm size on the efficiency of the algorithm.

3 Results

Our simulations contribute to two distinct analyses. On the one hand, we wanted to investigate the flocking capabilities of our swarms and, on the other hand, we wanted to determine the algorithm's efficiency to let randomly scattered robots form an aggregation in an arena. Here, *aggregated* or *flock* means that we counted each robot which was inside the IR-sensor field of another robot and counted the total number of robots within that *connected* swarm.

3.1 Flocking Analysis

Our first simulations were set up to find out the mobility of an aggregated robot swarm. For this we simulated a huge 300x300 robot-diameter^2 arena so that the robots were not confined in their movement by the borders. An already aggregated swarm of 1 to 25 robots with randomized headings was placed into the middle of this arena. We then added up the movements of all robots for 60 seconds and calculated the average distance covered by a robot in the swarm. Additionally, we measured the movement of the coherent flock by calculating its center of mass and adding up its path.

An ideally mobile flock would cover the same distance as the average distance covered by a robot in the flock. Since our algorithm does merely impede but not prevent a flock from splitting up, we only counted simulation runs where the whole swarm stayed together the whole time of the simulation, which was the case, on average, in 46% of all runs. Each experiment was repeated until there were 10 successful runs. A comparison of the distance measurements for the robots and the center of mass of the flock can be seen in Fig. 2B. The average movement of a robot in a flock differs slightly due to the implemented speed error.

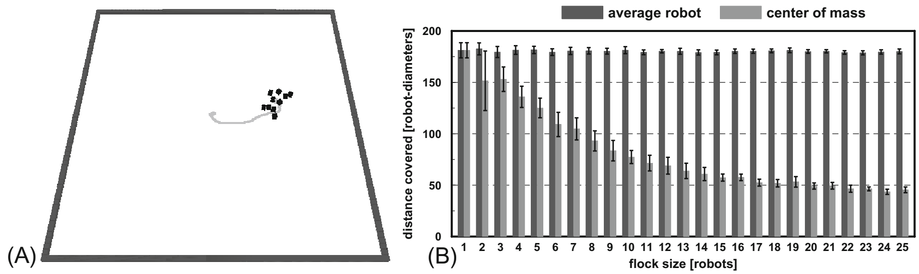


Fig. 2. A: Screenshot of our simulator with a swarm of 10 robots (black cubes) in a small arena with borders. The path of the center of mass of the flock is depicted by a grey line. B: Average movement of a robot in a flock (dark grey) in comparison to the average movement of the center of mass of the flock (light grey) in 60 seconds. Means and standard deviations of 10 repetitions.

3.2 Aggregation Analyses

General Efficiency: Next, we focussed on finding out the aggregation efficiency of a robot swarm which uses our algorithm. In these simulations the robots had been randomly scattered in an arena and had to aggregate. There was no global information about where to aggregate and the robots reacted purely on short-range IR-detection. A swarm was considered as being aggregated when at least 60% of its members were in the same flock. We measured the time it took a scattered swarm to aggregate under different combinations of swarm size and arena size. 100 repetitions with randomized starting positions and headings were evaluated, the result can be seen in Fig. 3A.

Density-neutral Efficiency: Further simulations were conducted only under density-neutral conditions, for example 5 robots in a 3600 robot-diameter^2 arena or 10 robots in a 7200 robot-diameter^2 arena. This was done to measure the influence of swarm size on the efficiency of the algorithm. A supplementary measurement was made by increasing the desired aggregation size from 60% to 80% (Fig. 3B). Each simulation was evaluated by the median, first and third quartiles of 100 repetitions.

4 Discussion

The flocking analysis showed that the mobility of a flock of swarm robots which use our algorithm was dependent on the flock's size. A small flock of 5 robots still moved 69% of the distance a single robot could have moved, whereas increasing the flock size to up to 25 robots decreased the mobility down to 25%. This decrease is of course a major drawback in comparison to traditional flocking algorithm implementations, where large flocks can still be quite mobile.

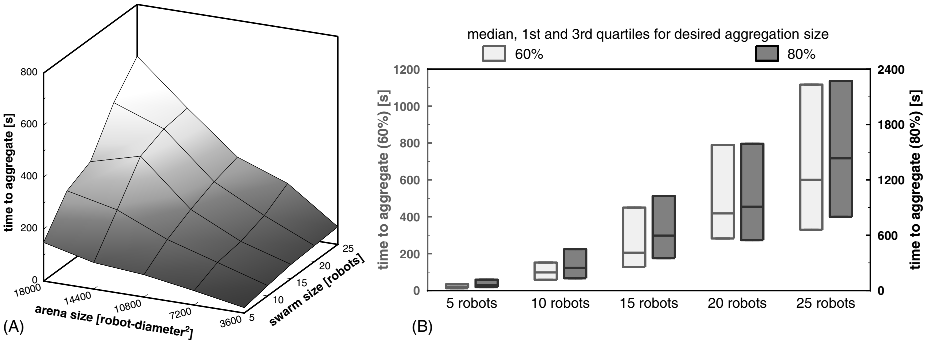


Fig. 3. A: Aggregation speed analysis for all swarm sizes and all arena sizes. We measured the median time it took for an initially scattered robot swarm to form an aggregation of at least 60% of the whole swarm. B: Comparison of aggregation time for an initially scattered robot swarm in density neutral setups for 60% (light grey) and 80% (dark grey) desired aggregation size. Median, first and third quartiles of 100 repetitions.

The aggregation analyses suggested that our flocking algorithm was quite efficient concerning aggregating scattered robots. It should be noted that, contrary to our expectations, there was an increase in aggregation time when increasing the number of robots in the same arena. In retrospective this can be explained by the flocking analysis, which showed that with increasing numbers of flock mates, the movement of the flock was more and more limited. Thus it became harder for two larger flocks to merge in order to attain the desired aggregation size.

Our density-neutral experiments showed that the aggregation time was increased when increasing the number of robots and the size of the arena. This is due to the larger distances that flocks have to cover in order to join other flocks and the aforementioned decrease of flock movement in bigger flocks. When the desired aggregation size was changed from 60% of the whole swarm to 80% of the whole swarm, there was a large increase in aggregation time. The reason why we did not test the goal of achieving 100% aggregation was that, when working with numerous swarm robots, one has to expect a small fraction of malfunctioning or stuck robots, so that *perfect* achievements are rather improbable.

Summing up, our flocking algorithm enables swarm robots to form a coherent and reactive flock which moves around in the arena randomly. Our results suggest that it works well with small swarms and is especially suited for robots with minimal equipment. It could, for instance, be used for the aggregation part of a more comprehensive scenario. Moreover, the algorithm can also be used for a heterogeneous robotic swarm when the robot types use the same distance-sensing method. Our next step will be to port the algorithm to a real swarm of up to 30 heterogeneous swarm robots [22, 23].

[1] Supported by: EU-IST-FET project ‘SYMBRION’, no. 216342; EU-ICT project ‘REPLICATOR’, no. 216240; EU-IST FET project ‘I- Swarm’, no. 507006.

References

1. Wilson, E.O.: The Insect Societies (Harvard Paperbacks). Belknap Press (November 1974)
2. Beni, G., Wang, J.: Swarm intelligence. In: Proc. of the Seventh Annual Meeting of the Robotics Society of Japan, pp. 425–428 (1989)
3. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
4. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford Univ. Press, Oxford (1999)
5. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: A taxonomy for swarm robots. In: Intelligent Robots and Systems 1993, vol. 1, pp. 315–325 (1993)
6. Seyfried, J., et al.: The I-SWARM project: Intelligent small world autonomous robots for micro-manipulation. In: Şahin, E., Spears, W.M. (eds.) SAB 2004. LNCS, vol. 3342, pp. 70–83. Springer, Heidelberg (2005)
7. Trianni, V., Groß, R., Labella, T., Şahin, E., Dorigo, M.: Evolving aggregation behaviors in a swarm of robots. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) ECAL 2003. LNCS (LNAI), vol. 2801, pp. 865–874. Springer, Heidelberg (2003)
8. Martinoli, A., Easton, K., Agassounon, W.: Modeling swarm robotic systems: A case study in collaborative distributed manipulation. Int. Journal of Robotics Research 23(4), 415–436 (2004)
9. SYMBRION: Project website (2009), http://www.symbrion.eu/
10. REPLICATOR: Project website (2009), http://www.replicators.eu/
11. Reynolds, C.W.: Flocks, herds, and schools. Computer Graphics 21(4), 25–34 (1987)
12. Mataric, M.J.: Designing emergent behaviors: from local interactions to collective intelligence. In: Proc. of the Second Int. Conf. on From Animals to Animats 2: Simulation of Adaptive Behavior, pp. 432–441 (1993)
13. Turgut, A., Çelikkanat, H., Gökçe, F., Şahin, E.: Self-organized flocking in mobile robot swarms. Swarm Intelligence 2(2), 97–120 (2008)
14. Hayes, A.T., Dormiani-Tabatabaei, P.: Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots. In: Int. Conf. on Robotics and Automation, pp. 3900–3905 (2002)
15. Balch, T., Hybinette, M.: Social potentials for scalable multi-robot formations, vol. 1, pp. 73–80 (2000)
16. Kelly, I.D., Keating, D.A.: Flocking by the fusion of sonar and active infrared sensors on physical autonomous mobile robots. In: Proc. of the The Third Int. Conf. on Mechatronics and Machine Vision in Practice, vol. 1, pp. 1–4 (1996)
17. Baldassarre, G., Nolfi, S., Parisi, D.: Evolving mobile robots able to display collective behaviors. Artificial Life 9(3), 255–267 (2003)
18. Bjerknes, J.D., Winfield, A., Melhuish, C.: An analysis of emergent taxis in a wireless connected swarm of mobile robots. In: IEEE Swarm Intelligence Symposium, pp. 45–52. IEEE Press, Los Alamitos (2007)
19. Partridge, B.L., Pitcher, T.J.: The sensory basis of fish schools: relative roles of lateral line and vision. Journal of Comparative Physiology 135(4), 315–325 (1980)
20. Wilensky, U.: Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999)
21. Moeslinger, C.: Video link (2009), http://zool33.uni-graz.at/artlife/flocking
22. Swarmrobot: Project website (2009), http://www.swarmrobot.org/tiki-index.php
23. ePuck: e-puck desktop mobile robot - website (2009), http://www.e-puck.org/

Networks of Artificial Social Interactions

Peter Andras

School of Computing Science, Newcastle University, Claremont Tower,
Claremont Road, Newcastle upon Tyne,
NE1 7RU, United Kingdom
peter.andras@ncl.ac.uk

Abstract. Evolution of cooperation is a fundamental question of socio-biology. Intrinsic factors like kinship play an important role in cooperation among selfish individuals. External factors like uncertainty and the structure of the social interaction network also contribute significantly to the evolution of cooperation. Here I use agent-based simulations to generate artificial social networks. I show that some of these networks have similar scale-free structure as real social networks. The analysis shows that having agents with memory and with the ability to share their memory through gossiping does not have a significant effect on the scale-free nature of simulated social networks. However the presence of high uncertainty in the cooperation games played by the agents is required for the generation of scale-free social interaction networks.

Keywords: agent-based modeling, cooperation, evolution, network analysis, simulation, uncertainty.

1 Introduction

The evolution of cooperation among selfish individuals is fundamental question of socio-biology [1-4]. Natural examples show that such cooperation may emerge in various species, including plants [5], fish [6], and various mammals [7,8]. The most influential theories argue that kin-selection or similarity-selection [1,9] and direct and indirect reciprocity [2,10] play a key role in the emergence and evolution of cooperation among selfish individuals. Other theories emphasize the role of segregation of cooperators and defectors [11], or effects of some kind of group selection [4].

In addition to intrinsic factors (e.g. genetic relatedness, similarity) external factors also play an important role in facilitating the evolution of cooperation. For example, uncertainty plays a major role in facilitating the emergence of cooperative arrangements in human economic activity (e.g. insurance, common goods) [12]. Another important factor is the nature of interaction networks between individuals. It has been shown that in case of humans such interaction networks have small-world and scale-free features [13,14]. Such networks may play an important role in providing the confirmed trust base that is critical for the maintenance of human cooperation [15].

In agent-based simulation studies of cooperation the usual assumption is that the interaction network is a fully connected graph [3]. More recent simulation studies

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 383–390, 2011.

considered the replacement of the fully connected graph by less connected interaction graphs, e.g. scale-free and small-world graphs [16] – note that in all these cases the interaction graph is given by design of the simulation. Another approach is to let the interaction graph be driven by spatial neighborhood arrangements, i.e. the interaction partners should be spatially sufficiently close to each other and as they may move they may change their neighborhood [17].

Here I study emergent interaction networks between agents in the context of agent-based simulations of evolution of cooperation using uncertain cooperation games [17]. The interaction networks are determined by neighborhood relations and also possibly by memories of interactions and gossip about interactions between other agents. The aim of the analysis is to investigate to what extent the emerging interaction networks are similar to scale-free networks that are a characteristic of real world social interaction networks. I show that the presence of memories and gossip features does not influence significantly the structure of simulated social networks, and that the presence of high uncertainty represented in the cooperation games played by the agents is required to generate scale-free social interaction networks.

The rest of the paper is organized as follows. First I introduce simulation studies of the evolution of cooperation. Next I introduce social interaction networks, including real and simulated networks. This is followed by the presentation of the agent-based simulation used here and the presentation of the results about emergent artificial social interaction networks. The paper is closed by the conclusion section.

2 Simulation of Evolution of Cooperation

In agent-based simulations of evolution of cooperation [3,9,11,17] simple agents play cooperation games with other agents and accumulate resources. The agents follow some simple game playing strategy. When the agents reach the end of their simulated life they generate offspring, which inherit the game playing strategy of the parent. The number of offspring increases with the resource wealth of the agent. These studies aim to study the extent to which cooperation oriented strategies become dominant in simulated agent populations.

The cooperation games that simulated agents play are defined by a pay-off matrix, which determines how much the agents get out of the game playing depending on their choice of cooperation / defection decision. A commonly used example is the Prisoner's Dilemma game, for which the game matrix is presented in Table 1 and the rules that apply to the pay-off values are given by the inequalities $t>r>p>s$ and $2r>t+s$.

A way to represent uncertainty in cooperation games is to use pay-off distributions instead of fixed values of pay-offs [17]. The external uncertainty in such case is represented by the variance of the pay-off distributions. When the agents play the game they determine their actual pay-off by sampling the corresponding pay-off distribution. In order to keep the game's characteristic inequalities satisfied it is possible to consider distributions for a smaller set of parameters to determine actual pay-offs of the game matrix. For example, in the case of the Prisoner's Dilemma, the pay-offs can be determined as $r=b+d/2$, $t=b+d$, $s=0$, and $p=b$, where b and d are determined by sampling two normal distributions. If the variance of these distributions increases the samples may deviate more from the mean, implying increased variation of possible pay-offs.

Table 1. The pay-off matrix of a Prisoner's Dilemma game

	Player 2		
Player 1		Cooperate	Defect
	Cooperate	*r,r*	*s,t*
	Defect	*t,s*	*p,p*

Agents usually play a fixed number of repeated games with the same playing partner [3,9]. It is also possible that they play a variable number of games or even just one game in each round of game playing [17]. The agents that an agent plays with are determined by a play graph, which may be fully connected, sparse, or possibly dynamically determined for example in function of neighborhood relations between agents [3,16,17]. Over many rounds and many generations of agents some game playing strategies may become dominant; these are the evolutionarily stable strategies (ESS). The question is whether cooperation is supported by ESS and in what conditions does this happen.

Analytical and simulation studies show that inherent mechanisms, like kin selection and reciprocity can lead to the dominance of cooperation oriented strategies [1,2,9,10]. However, external conditions may facilitate this or may prevent it. For example, increased externally induced uncertainty increases the level of cooperation in simulated agent populations [17].

3 Social Interaction Networks

Social networks have been analyzed for several decades. The classical result of six degrees of separation established that human social networks have the small-world feature [13]. More recent analysis of social networks of co-acting of actors [13], of sexual partner relationships [14], and of mobile phone calls [18] shows that these human interaction networks have the scale-free feature. An important implication of the scale-free feature is the resilience of the functionality of the network to random changes. While the small-world and the scale-free features are not equivalent they are related, and they characterize jointly most human social networks.

The key factor that leads to generation of scale-free social networks is the mechanism of preferential attachment, i.e. that people tend to be introduced or to get in contact with other people who already know many other people. This means that the probability of establishing a new connection between two nodes is proportional with the numbers of existing connections of the two nodes. Similarly, the probability of breaking of a connection is inversely proportional with the number of connections of the two nodes.

The structure of the interaction network is very important from the perspective of evolution of cooperation [13-16]. This determines the spread of information in the social system, the likely interaction partners of individuals, and constraints on similarity selection and reciprocity. The social interaction network structure may constrain or may facilitate the evolution of cooperation in the community of individuals.

The usual interaction network in simulation studies of cooperation is a pre-set fully connected network [3,9]. This means that all agents play with all other agents in each round of the game. This setting may fit the case of very small social groups, but it is not a valid model in case of larger social groups. More recently researchers used also particular pre-set scale free networks [16]. The latter setting approximates better real world social situations, but still its disadvantage is the static nature of the pre-set interaction network.

A more dynamic alternative is to use at start a pre-set scale-free (or other random) network and rewire the network according to its design principle after every round of play between agents [19]. Another approach is to consider the agents in some spatial setting (e.g. on a rectangular grid) and to use spatial neighborhood relations between agents to determine the network of interactions [17]. Again this may be done by considering fully connected neighborhood networks or random networks within neighborhoods. Allowing the number of game plays to change randomly; or in the extreme allowing only a single game play between interacting agents may add further realism to the interaction networks used in simulation studies [17].

4 Emergent Artificial Social Networks – Are These Scale-Free Networks ?

The results and analysis reported here aim to check whether scale-free interactions networks can be generated in agent-based simulations without explicitly coding such interaction networks into the simulation. While preferential attachment is a simple and intuitive mechanism in social context, a question is whether other less explicit mechanisms may lead to the same or similar results in terms of the arrangement of social interaction networks. Here I analyze the role of uncertainty, memory and gossip through an agent-based simulation setting.

The simulated agent world that I use here has been described in detail in earlier papers [17,20]. The agents play an uncertain Prisoner's Dilemma game and gain resources by playing the game. They spend some of these resources on living costs in each turn. The playing strategy of agents is fixed, and it is given as the probability of choosing the cooperation decision (*p*). The agents move randomly in a rectangular world with warped edges. In each round, each agent tries to pick an interaction partner from its neighborhood. The agents play a single game with their interaction partner in each round. The agents live in average for 100 rounds. At the end of their life they may produce offspring asexually. The number of offspring depends on the amount of resources accumulated by the parent agent. Higher amount of resources means more offspring according to a saturating sigmoid function. The offspring inherit the playing strategy of their parent with random modifications. The offspring start their life from the location of their parent in diffuse from there through their random movements.

The agents are equipped with memory and can remember the outcome of their interactions with the last ten other agents. The memories fade with time. Depending on their memory agents may increase or decrease the probability of cooperation with another agent (i.e. the actual probability of cooperation becomes $p'=a.p$, where $a>1$ in case of positive memories and $0<a<1$ in case of negative memories). The agents are also able to gossip. This means that they can share memories about interactions with other agents and combine such information received from other agents with their own memory information. In this way they can adapt their cooperation behavior in the case of meeting of agents that they have not encountered previously. The memory and gossip abilities were switched on in some experiments, while there were not used in others in order to study the effect of these features on the formation of artificial social networks.

To measure social interaction networks the simulations were run 20 times at two levels of uncertainty (low and high) with all memory/gossip settings (no memory/no gossip, memory/no gossip, memory and gossip). Each time the simulation was run for 1000 rounds (~10 generations of agents). All interactions were recorded and the generated interaction graphs were considered for measurement (an interaction graph was calculated for every 100 rounds). The distributions of the connectedness of nodes of these graphs were calculated. I measured whether these distributions were power law distributions or not.

To test the power low nature of these distributions it is assumed that the power law has the probability density function

$$p(x)=\frac{\gamma-1}{x_{\min}}\cdot\left(\frac{x}{x_{\min}}\right)^{-\gamma} \tag{1}$$

where x_{min} is the cut-off value above which the distribution follows the power law, and γ is the exponent of the power law. I used the estimators proposed in [21] to estimate the parameters and the fit of the assumed power law considering the actual connectivity distributions of the generated networks. In particular the exponent γ is estimated as

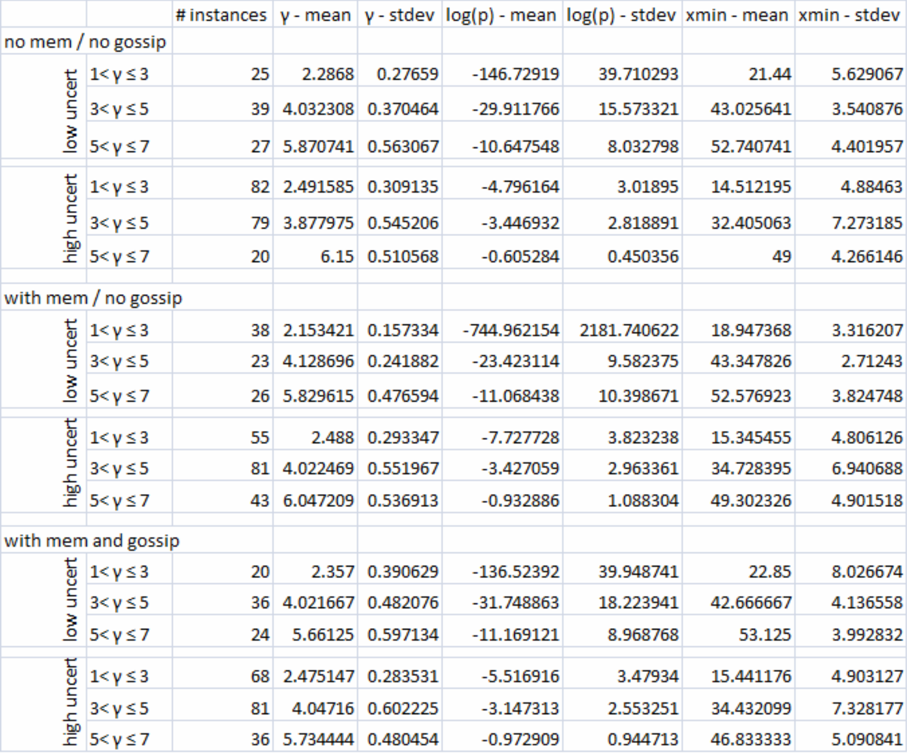
$$\bar{\gamma}=1+\frac{n}{\sum_{i=1}^{n}\ln\left(\frac{x_i}{x_{\min}}\right)} \tag{2}$$

where $x_i>x_{min}$ are the observed values of connectedness, n is the number of graph nodes for which the connectedness was calculated, and the estimate of the value of x_{min} is calculated as the value for which the Kolmogorov-Smirnov fitness measure between the actual distribution and the expected power law distribution is the best. The fitness measure is calculated as

$$D=\max_{x\geq x_{\min}}|S(x)-P(x)| \tag{3}$$

where $S(x)$ is the cumulative probability distribution of the data and $P(x)$ is the cumulative probability distribution of the estimated power law distribution; the lower D value is the better. The Matlab algorithms used for the distribution fitness calculations [21] are stable for exponents $1.5<\gamma<7.0$. The corresponding significance levels of the match with the best fitting power law were calculated using the Kolmogorov-Smirnov test of the Matlab.

Table 2. Analysis of experimental connectedness distributions of simulated social networks. The log(p) is the logarithm of the calculated significance level – the network is significantly different from a scale-free network if log(p) < –2.



		# instances	γ - mean	γ - stdev	log(p) - mean	log(p) - stdev	xmin - mean	xmin - stdev
no mem / no gossip								
low uncert	1< γ ≤ 3	25	2.2868	0.27659	-146.72919	39.710293	21.44	5.629067
	3< γ ≤ 5	39	4.032308	0.370464	-29.911766	15.573321	43.025641	3.540876
	5< γ ≤ 7	27	5.870741	0.563067	-10.647548	8.032798	52.740741	4.401957
high uncert	1< γ ≤ 3	82	2.491585	0.309135	-4.796164	3.01895	14.512195	4.88463
	3< γ ≤ 5	79	3.877975	0.545206	-3.446932	2.818891	32.405063	7.273185
	5< γ ≤ 7	20	6.15	0.510568	-0.605284	0.450356	49	4.266146
with mem / no gossip								
low uncert	1< γ ≤ 3	38	2.153421	0.157334	-744.962154	2181.740622	18.947368	3.316207
	3< γ ≤ 5	23	4.128696	0.241882	-23.423114	9.582375	43.347826	2.71243
	5< γ ≤ 7	26	5.829615	0.476594	-11.068438	10.398671	52.576923	3.824748
high uncert	1< γ ≤ 3	55	2.488	0.293347	-7.727728	3.823238	15.345455	4.806126
	3< γ ≤ 5	81	4.022469	0.551967	-3.427059	2.963361	34.728395	6.940688
	5< γ ≤ 7	43	6.047209	0.536913	-0.932886	1.088304	49.302326	4.901518
with mem and gossip								
low uncert	1< γ ≤ 3	20	2.357	0.390629	-136.52392	39.948741	22.85	8.026674
	3< γ ≤ 5	36	4.021667	0.482076	-31.748863	18.223941	42.666667	4.136558
	5< γ ≤ 7	24	5.66125	0.597134	-11.169121	8.968768	53.125	3.992832
high uncert	1< γ ≤ 3	68	2.475147	0.283531	-5.516916	3.47934	15.441176	4.903127
	3< γ ≤ 5	81	4.04716	0.602225	-3.147313	2.553251	34.432099	7.328177
	5< γ ≤ 7	36	5.734444	0.480454	-0.972909	0.944713	46.833333	5.090841

The summary of the results is presented in Table 2. The analysis shows that many of the generated simulated social networks can be considered as scale free networks according to the above tests. The calculated power value (γ) varies in a wide range (2 to 6.5 – we ignored all cases when the calculated γ value reached above 7, since at this level the used algorithms become unstable); however the goodness of fit between the experimental connectedness distribution and the corresponding power law distribution is much better for γ values above 5 than for values below 3. The calculated x_{min} values also vary accordingly from 12 to 60; higher x_{min} values being associated with higher γ values.

The comparison of results between the three memory and gossip settings shows that there is no significant effect of these features on the power law nature of the connectedness distributions of the corresponding simulated social networks. In all three settings we find similar power (γ) and cut-off value (x_{min}) ranges and similar levels of goodness of fit with corresponding theoretical power laws. This means that in the context of the presented agent-based simulations having memory and being able to share memories through gossip do not influence the generic nature of the structure of the resulting simulated social networks.

However the results show that the level of uncertainty incorporated into the pay-off matrices of the games played by the agents has a much more important effect on the network structure of the social interaction networks. In case of high uncertainty in all memory/gossip settings the connectedness distributions of the social networks match much better the power law distribution than in the case of low uncertainty. This is true at all levels of power (γ) and cut-off (x_{min}) values.

In summary, the results show that it is possible to generate simulated social networks that have a scale-free networks structure, using agent-based simulations of cooperative behavior. The scale-free nature of the resulting simulated social networks does not depend on the presence of memory and gossip features, but depends critically on the presence of high uncertainty in the cooperation games played by the agents. It is important to note that in these simulations the principle of preferential attachment was not explicitly encoded, and that social networks which are in agreement with this growth principle resulted simply by playing high uncertainty cooperation games by simulated agents.

5 Conclusions

The structure of social interaction networks is likely to contribute to the evolution of cooperation in socio-biological systems supported by these networks. Here I show that it is possible generate realistic scale-free simulated social interaction networks through agent-based simulations.

The results show that the scale-free nature of the simulated social networks is primarily influenced by the uncertainty present in the cooperation games played by the simulated agents. The results also show that the network structure does not depend significantly on the presence of memory and gossip features in the simulations.

Being able to generate realistic simulated social networks without the explicit encoding of mechanisms that generate scale-free networks is also an important step towards the improved analysis of social networks. Having such simulated social networks allows specifying the functionality of the simulated social system (e.g. evolution of cooperative behavior) and then the generation of simulated social networks that match their real counterparts in terms of structural features. In this way the role of the network structure in the functionality of the supported social system can be analyzed more independently since the network structure is not explicitly specified in the simulation.

References

1. Hamilton, W.D.: The genetical evolution of social behaviour I and II. J. Theor. Biol. 7, 1–52 (1964)
2. Trivers, R.L.: The evolution of reciprocal altruism. Q. Rev. Biol. 46, 35–57 (1971)
3. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity by image scoring. Nature 393, 573–577 (1998)

4. Scheuring, I.: Evolution of generous cooperative norms by cultural group selection. J. Theor. Biol. 257, 397–407 (2009)
5. Callaway, R.M., et al.: Positive interactions among alpine plants increase with stress. Nature 417, 844–847 (2002)
6. Seghers, B.H.: Schooling behaviour in the guppy (Poecilia reticulata): an evolutionary response to predation. Evolution 28, 486–489 (1974)
7. Spinks, A.C., Jarvis, J.U.M., Bennett, N.C.: Comparative patterns of philopatry and dispersal in two common mole-rat populations: implications for the evolution of mole-rat sociality. J. Anim. Ecol. 69, 224–234 (2000)
8. Fehr, E., Fischbacher, U.: The nature of human altruism. Nature 425, 785–791 (2003)
9. Riolo, R.L., Axelrod, R., Cohen, M.D.: Evolution of cooperation without reciprocity. Nature 414, 441–443 (2001)
10. Brandt, H., Sigmund, K.: Indirect reciprocity, image scoring, and moral hazard. PNAS 102, 2666–2670 (2005)
11. Pepper, J.W., Smuts, B.B.: A mechanism for the evolution of altruism among nonkin: Positive assortment through environmental feedback. American Naturalist 160, 205–213 (2002)
12. Smith, B.D., Stutzer, M.A.: A theory of mutual formation and moral hazard with evidence from the history of insurance industry. Rev. Fin. Stud. 8, 545–577 (1995)
13. Collins, J.J., Chow, C.C.: It's a small world. Nature 393, 409–410 (1998)
14. Liljeros, F., Edling, C.R., Amaral, L.A.N., Stanley, H.E., Aberg, Y.: The web of human sexual contacts. Nature 411, 907–908 (2001)
15. Jones, G.R., George, J.M.: The experience and evolution of trust: implications for cooperation and teamwork. Acad. Manag. Rev. 23, 531–546 (1998)
16. Ohtsuki, H., Nowak, M.A.: Evolutionary stability on graphs. J. Theor. Biol. 251, 698–707 (2008)
17. Andras, P., Lazarus, J., Roberts, G.: Environmental adversity and uncertainty favour cooperation. BMC Evol. Biol. 7(240) (2007)
18. Palla, G., Barabasi, A.L., Vicsek, T.: Quantifying social group evolution. Nature 446, 664–667 (2007)
19. Traulsen, A., Nowak, M.A., Pacheco, J.M.: Stochastic payoff evaluation increases the temperature of selection. J. Theor. Biol. 244, 349–356 (2007)
20. Andras, P., Lazarus, J., Roberts, G., Lynden, S.J.: Uncertainty and cooperation: Analytical results and a simulated agent society. J. Artif. Soc. & Soc. Simul. 9(1/7) (2006)
21. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. arXiv: 0706.1062 (2007)

An Ant-Based Rule for UMDA's Update Strategy

C.M. Fernandes[1, 2], C.F. Lima[3], J.L.J. Laredo[1], A.C. Rosa[2], and J.J. Merelo[1]

[1] Department of Computer's Architecture and Technology, University of Granada, Spain
{juanlu,jmerelo}@geneura.ugr.es
[2] LaSEEB-ISR-IST, Technical Univ. of Lisbon (IST), Portugal
{cfernandes,acrosa}@laseeb.org
[3] Informatics Lab., University of Algarve, Gambelas Campus, 8000-117, Faro, Portugal
clima.research@gmail.com

Abstract. This paper investigates an update strategy for the Univariate Marginal Distribution Algorithm (UMDA) probabilistic model inspired by the equations of the Ant Colony Optimization (ACO) computational paradigm. By adapting ACO's transition probability equations to the univariate probabilistic model, it is possible to control the balance between exploration and exploitation by tuning a single parameter. It is expected that a proper balance can improve the scalability of the algorithm on hard problems with bounded difficulties and experiments conducted on such problems with increasing difficulty and size confirmed these assumptions. These are important results because the performance is improved without increasing the complexity of the model, which is known to have a considerable computational effort.

1 Introduction

Estimation of Distribution Algorithms (EDAs) [8] constitute a class of Evolutionary Algorithms (EAs) [7] in which the standard crossover and mutation operators are replaced by: 1) an estimation of the joint probability of promising solutions, and; 2) the generation of new solutions by sampling from the corresponding estimated distribution. During the optimization process, an EDA makes use of the probabilistic models to build possible solutions to the problem (*sampling*). The probability model is then updated in a way that reflects the quality of those solutions (*selection*).

Unlike traditional EAs, this class of algorithms does not use genetic operators, and relies exclusively on the probability model, not only to converge to a proper solution but also to capture the underlying problem decomposition during runtime. This particular behaviour overcomes one of the disadvantages of traditional EAs: fixed and problem-independent operators often disrupt and do not mix properly the raw building-blocks provided by the initial population. Some EDAs, being able to capture the problem structure, scale much better than traditional EAs, namely on problems with strong dependencies between variables.

Different strategies may be used at the sampling and selection steps, meaning that the diversity of the model (and resulting algorithm's convergence) is strongly dependent on the chosen schemes. EDAs are usually classified according to the complexity of the probabilistic model they rely on, i.e., the level of interactions they can capture between the variables of the problem. On univariate models, for instance,

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 391–398, 2011.

the variables are assumed to be independent. One of the algorithms based on such models is the the *Univariate Marginal Distribution Algorithm* (UMDA) [10], which is the one addressed in this paper. However, there are EDAs based on other types of models, like the bivariate, which represent pairwise interactions between variables, usually via chain or tree structures, and the multivariate [9], which relies on much more complex probabilistic models.

Univariate models' learning complexity is very low and they perform well on linear problems, while failing when the interactions between the variables are strong. Multivariate EDAs rely on more complex model learning, but they are able to detect interactions between variables, thus being very efficient on some hard problems that are intractable by other EAs. However, multivariate EDAs require an extra computational effort in order to estimate the distribution and learn the variables' interactions. It is thus necessary to find the best compromise between the simplicity of univariate models and the learning abilities of multivariate ones when facing a new problem. Therefore, any effort that results in improvements on simpler models may be widening the range of application of such algorithms. This paper addresses that theme and studies the performance of an update strategy for UMDA's probabilistic model inspired by the Ant Colony Optimization (ACO) transition probabilities [5]. These equations have one parameter that may be tuned in order to control the algorithm's balance between exploration and exploitation and it is expected that a proper tuning of the parameter enhances UMDA's behaviour on hard problems, namely its scalability on m-k trap functions. The results confirmed this hypothesis. Before proceeding to the description of the algorithm and results, the following section addresses EDAs in the model-based search context and relates them to ACO.

2 Model-Based Search

EDAs are classified by Zlochin et al. [11] within a larger framework that includes also stochastic gradient ascent, cross-entropy method and ACO – see [11] for a description of these methods. The authors identified the similarities between these algorithms, with the aim of unifying them in a single framework. All these heuristics rely on a probabilistic model that is updated according to the results of the search process. It is thus plausible that some features that have been proven to be effective in one model-based search heuristic can be used in another algorithm of the same kind with success. We are particularly interested in merging EDAs and ACO.

ACO is a computational paradigm inspired by the foraging behaviour of real ants, i.e., their ability to find the shortest paths between the nest and food sources. ACO belongs to a wider class of algorithms called ant algorithms, which rely on a principle called stigmergy, or *indirect communication via the environment* [6]. Ant algorithms may be used not only on combinatorial optimization but also on clustering and classification problems, image processing, robotics and other real-world problems [1].

Before being tackled by an ACO algorithm, a problem must be reduced to finding the shortest paths in graphs. Then, artificial ants will travel possible trails, and deposit, on the edges, an amount of pheromone that is proportional to the quality of the path. In addition, the ants go from node to node according to a probability that depends on the pheromone. Ant System [3] – the first ACO algorithm –, for instance, may be described by equation 1, which defines the probability with which at t-th iteration an ant k located in node i chooses an adjacent node $j \in \mathcal{N}_i^k$ to move to:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} \quad \forall_{j \in \mathcal{N}_i^k} \tag{1}$$

where $\mathcal{N}_i^k$ is the feasible neighbourhood of node i if for ant k, τ_{ij} is the amount of pheromone in edge ij, η_{ij} is a component that incorporates a priori knowledge, α is a parameter that controls the influence of τ_{ij} and β controls the influence of η_{ij}. The pheromone, in every edge ij, is updated in every iteration:

$$\tau_{i,j}(t) = \Delta\tau_{i,j} + \tau_{i,j}\,(t-1) \times (1-\rho) \tag{2}$$

where ρ is the pheromone's evaporation rate and $\Delta\tau_{i,j}$ is the amount of pheromone deposited, which must depend on cost of the ant's tour (if the edge is part of an ant's tour; otherwise, $\Delta\tau_{i,j} = 0$). This is very similar to EDAs' process for updating the probabilistic model according to the previously found solutions. In fact, Fernandes *et al.* based their Binary Ant Algorithm [4] on ACO, but a closer look soon revealed that the resulting algorithm mimics the behaviour of a bivariate EDA. More recently, and within this line of work, Fernandes *et al.* [5] proposed an ACO-like update strategy for UMDA with the objective of preserving diversity and tackle time-varying fitness functions. The results presented in [5] show that the method is able to outperform several other previously proposed strategies that reduce UMDA's diversity loss. This paper broadens that study and investigates the performance and scalability of that ACO-like model on stationary problems with different degrees of difficulty.

3 UMDA and Diversity Loss

UMDA is a discrete EDA with independent variables. It starts by initializing the probability model, assigning 0.5 to each parameter γ_i of the model, meaning that the first population is randomly generated. Parameters γ_i are defined as the probability that each variable takes the value 1:

$$\gamma_i \equiv P(x_i = 1) \tag{3}$$

where i = 1,…l, and l is the string length. After initializing the model, UMDA generates n strings x^{μ} according to the equation below:

$$P(x_i^{\mu}) = \prod_{i=1}^{l}[(\gamma_i x_i^{\mu} + (1-\gamma_i)(1-x_i^{\mu})] \tag{4}$$

where x_i^{μ} is the ith component of string x^{μ}. The $f \times n$ fittest strings are then selected by truncation or any other method ($f \in]0{,}1]$ is a parameter that defines UMDA's selective pressure). The chosen solutions update the model in the following manner:

$$\gamma_i \leftarrow \frac{1}{(f \times n)} \sum_{\mu \in D_S} x_i^{\mu} \tag{5}$$

where D_s is the selected population. The algorithm repeats until a stop criterion is met.

Please note that once a parameter of the model loses diversity ($\gamma_i = 0$ or $\gamma_i = 1$) UMDA has no means to regain it. (This effect is similar to genetic drift in traditional EAs.) For that reason, it is of extreme importance to avoid diversity (or variance) loss in order to escape premature convergence of the algorithm to local optima. Since the loss occurs at two steps of the algorithm – sampling and selection – those are the components that must be addressed. This work addresses selection.

Variance loss due to selection can be reduced by changing the way the probabilistic model is updated. In [2], the authors present four methods to correct parameter γ_i. In a previous study [5], our proposal – the Reinforcement-Evaporation (RE) correction – was included in a test set together with these techniques and proved to be more effective on dynamic problems.

RE strategy corrects probability distribution in order to reduce diversity loss by replacing equation 5 by ACO-like equations. Consider two vectors τ_i^0 and τ_i^1 that are updated in each time step as defined by equations 6 and 7:

$$\tau_i^1 \text{ (t)} \leftarrow \tau_i^1 \text{ (t)}+ \frac{1}{(fN)} \sum_{\mu \in D_S} x_i^\mu \tag{6}$$

$$\tau_i^0 \text{ (t)} \leftarrow \tau_i^0 \text{ (t)}+(1 - \frac{1}{(fN)} \sum_{\mu \in D_S} x_i^\mu) \tag{7}$$

where i = 1,...*l*, and $\tau_i^0(0) = 0$ and $\tau_i^1(0) = 0$. These vectors emulate ACO's pheromone maps and act as kind of memory, allowing UMDA to incorporate information from prior distributions into the current parameters. The parameters γ_i are then updated in the following manner:

$$\gamma_i \leftarrow \left(\frac{\tau_i^1}{\tau_i^0 + \tau_i^1} \right)^\alpha \tag{8}$$

where $\alpha \in]0,1]$ is a parameter that controls the relative weights of τ_i^0 and τ_i^1. Before the update stage, vectors $\tau_i^{0,1}$ are "evaporated" according to equation 9:

$$\tau_i^{0,1}(t) \leftarrow \tau_i^{0,1}(t) \times (1 - \rho) \tag{9}$$

where $\rho \in]0,1]$ is the evaporation rate. Please note that setting $\rho = 1$ implies that the vectors $\tau_i^{0,1}$ are set to 0 at the beginning of each time step, thus meaning that the previous equations are reduced to equation 10.

```
Set $\gamma_i$ ← 1/2 for all i = 1 . . .l; Set $\tau_i^{0,1}$← 0 for all i = 1 . . .l; Set $\alpha$ and $\rho$
repeat
    sample n strings according to Eq. 2 to make a population D.
    generate new population Ds by selecting the  f×n fittest strings.
    reinforce (equations 6 and 7) and evaporate pheromone (equation 9)
    for i = 1 to L do update model (equation 8)
  until stop criterion met
```

Fig. 1. Reinforcement-Evaporation (RE) UMDA

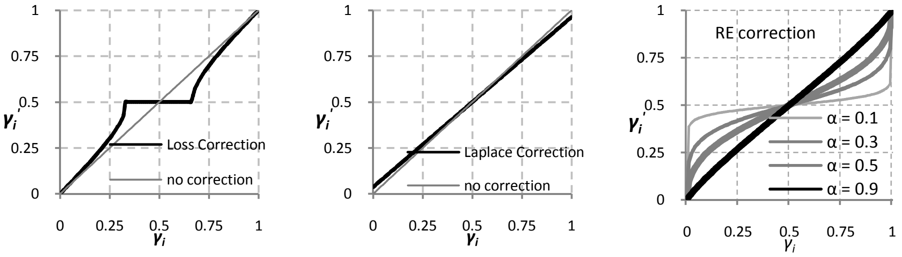


Fig. 2. Loss [2], Laplace [2] and RE correction effect on γ_i

$$\gamma_i \leftarrow \left(\frac{\sum_{\mu\in D_S} x_i^{\mu}}{f \times N}\right)^{\alpha} \tag{10}$$

Fig. 1 shows the pseudo-code of an UMDA with the proposed update strategy. Fig. 2 shows RE correction with different α values (ρ is set to 1) and compares it with loss correction and Laplace correction [2]. With low α values, the method approaches random search, since γ_i is kept close to 0.5 (with α = 0, UMDA does random search: γ_i is always 0.5). Increasing α cause the model to relax its exploration efforts and the diversity is reduced. When $\alpha \approx 1$ the model becomes very close to the standard update strategy (equation 3). Evaporation rate also controls diversity: when $\rho < 1$, UMDA incorporates prior distributions in the probabilities. Setting α = 1 and ρ = 1 results in the standard UMDA.

RE sigmoid shape indicates that the search process is kept (if α is properly tuned) in a more exploratory stage in the beginning, before engaging in a strong local search effort when the variables are converging. In that sense, RE is more similar to loss correction, but the α parameter allows the user to tune the balance between exploration and exploitation (please note that in the beginning of the search, loss correction does not bias the search towards good solutions). The following section investigates the effects of varying α on the performance of UMDA. Evaporation rate ρ is set to 1 (standard UMDA) and it is not addressed in this study.

4 Experimental Setup and Results

In order to investigate how the UMDA with the RE strategy scales when the size (or difficulty) of a problem increases, an experimental setup was designed that comprises the onemax problem and two m-k trap functions. Onemax is a simple linear problem that consists in maximising the number of ones in a binary string. A trap function is a piecewise-linear function defined on *unitation* (the number of ones in a binary string) that has two distinct regions in the search space, one leading to a global optimum and the other leading to the local optimum. Depending on its parameters, trap functions may be deceptive or not. The trap functions in these experiments are defined by:

$$F(u(\vec{x})) = \begin{cases} k, & if\ u(\vec{x}) = k \\ k - 1 - u(\vec{x}), & otherwise \end{cases} \tag{11}$$

where u($\vec{x}$) is the unitation function and k is the problem size (and also the fitness of the global optimum). With these definitions, order-3 traps are in the region between deceptive and non-deceptive, while order-2 are non-deceptive. Under these settings, it is possible to investigate not only how UMDA scales on order-k trap functions but also to observe how that performance varies when moving from non-deceptive to nearly-deceptive search spaces. For that purpose, l-bit decomposable functions are constructed by juxtaposing m trap functions and summing the fitness of each sub-function to obtain the total fitness. This way, we obtain the so-called $m-k$ trap problems, and by increasing m it is possible to investigate how an algorithm scales.

Scalability tests are important to assess how the algorithm behaves when problem size increases, and to what extent large-scale instances of the problem are still tractable. The following test intends to check the effects of RE strategy on UMDA's scalability by testing it on onemax, order-2 and order-3 traps with different sizes. For that purpose, the standard UMDA the baseline for comparison. In both approaches, the bisection method [11] is used to determine the optimal population size so that the given algorithm is able to find the optimum in 49 out of 50 runs. After determining the optimal population size, the configuration with that size is run for 100 times and the number of evaluations necessary to reach the optimum is averaged over the successful runs (that is, the runs in which the algorithm converged to the global optimum). The UMDAs are tested with parameter f set to $f = 0.1$.

The tests with the onemax problem had the expected outcome. As seen in fig. 3 – which represents the scalability of the algorithms in terms of evaluations to reach the optimum and optimal population size – RE does not improve standard UMDA's scalability and for $\alpha < 0.96$ the performance is even degraded. More exploration for onemax problem does not help because this is a linear problem that is easily solved with any pure exploitation method such as a hullclimbing local searcher.

As for the $m-k$ traps, RE strategy proved to be able to improve standard UMDA's performance. Fig. 4 shows that, on order-2, UMDA's scalability is improved by setting α to values around 0.96. Lower values did not enhance the performance. However, the tests with order-3 traps show that significant improvements in the scalability are only attained with $\alpha < 0.90$. The graphics in the bottom row of fig. 4 compare UMDA with $\alpha = 1.0$ (standard UMDA) and $\alpha = 0.80$. In this case, the population size scalability is particularly interesting, since the optimal size seems to remain stable when $l > 50$. A good scalability in terms of evaluations is very important because it saves computational effort and broadens the spectrum of problems that are tractable, but the scalability of the population is also crucial because of memory issues, which can feedback to performance increasing the running time.

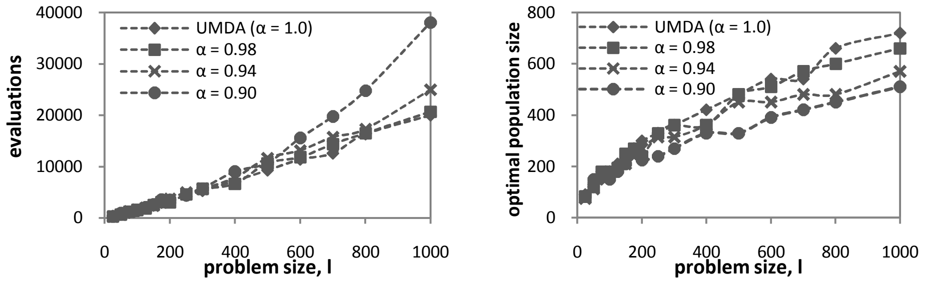


Fig. 3. Scalability on onemax problems

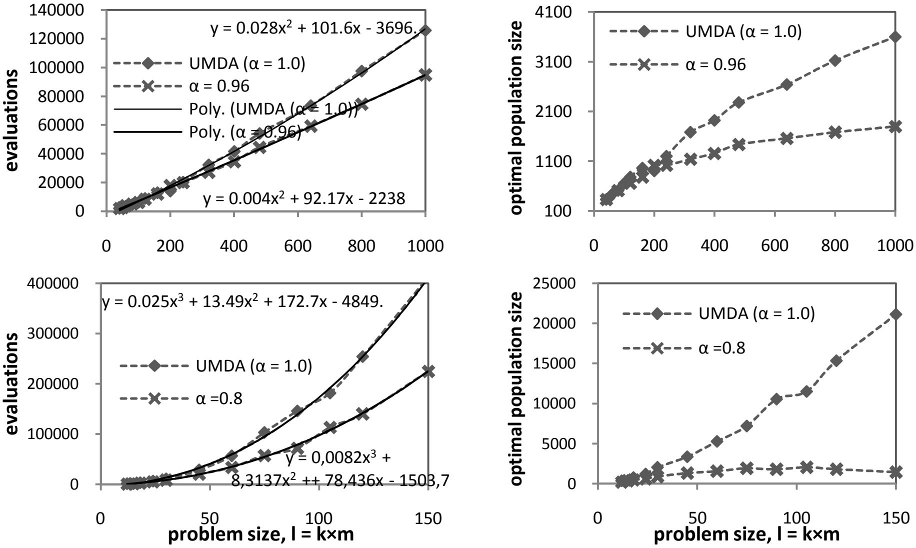


Fig. 4. Scalability on order-2 (top row) and order-3 (bottom row) m-k functions. Evaluations curves are fit by a polynomial function.

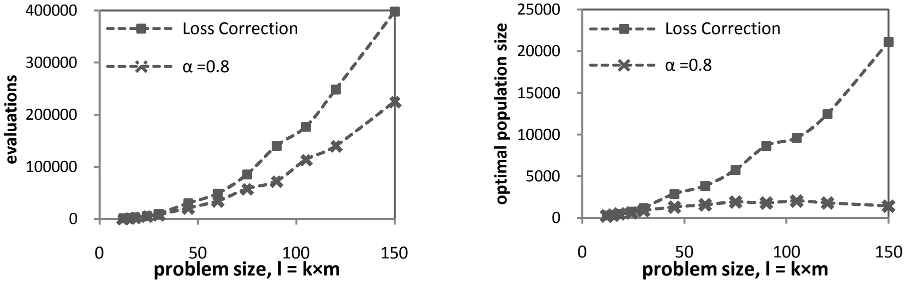


Fig. 5. Loss correction [2] and RE scalability on order-3 m-k functions

The significance of the results in fig. 4 were assessed via Kolmogorov-Smirnov tests and RE strategy was found to be statistically better (faster) than standard UMDA when $l \geq 320$ (order-2) and $l \geq 45$ (order-3). A final test with order-3 traps was made in order to compare RE with the loss correction method [2]. The results in fig. 5 show that, unlike RE, loss correction does not improve standard UMDA's performance on order-3 traps.

5 Conclusions and Future Work

This paper presents a study on the scalability of the Univariate Marginal Distribution Algorithms (UMDAs) with a non-traditional update strategy inspired by the Ant Colony Optimization (ACO) paradigm that aims at correcting the probabilities in order to avoid diversity loss: the Reinforcement-Evaporation (RE) correction. The proposed method, which mimics the pheromone deposition and evaporation found in ant colonies and modeled by ACO, incorporates a parameter that controls the balance between exploration and exploitation. The algorithms are tested on a linear problem (onemax), a non-deceptive trap (order-2) function and quasi-deceptive (order-3) trap function. Results show that increasing the exploration effort does not improve the performance on onemax, as expected, since the problem is easily solved by pure exploitation methods. However, tuning the parameter in order to increase exploration leads to better results than standard UMDA on trap problems. RE also outperformed a previously proposed correction method on order-3 traps.

In the future, the efficiency of this method on more complex EDAs will be investigated and the effects of evaporation rate on stationary functions will be studied. Finally, RE will be tested on dynamic trap functions.

Acknowledgements. This paper has been funded in part by the Spanish MICINN project NoHNES (Ministerio de Educación y Ciencia - TIN2007-68083-C02-01), the Junta de Andalucía P06-TIC-02025 and P07-TIC-03044, and the Portuguese Research Fellowship SFRH/BD/18868/2004 and Project FCT PTDC-EIA-67776-2006.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: from natural to artificial systems. Oxford University Press, Oxford (1999)
2. Branke, J., Lode, C., Shapiro, J.: Adressing sampling errors and diversity loss in UMDA. In: 2007 Genetic and Evolutionary Computation Conference, pp. 508–515. ACM, New York (2007)
3. Dorigo, M.: Optimization, learning and natural algorithms. Doctoral thesis, Politecnico de Milano, Italy (1992)
4. Fernandes, C., Rosa, A., Ramos, V.: Binary Ant Algorithm. In: 2007 Genetic and Evolutionary Computation Conference, pp. 41–48. ACM, New York (2007)
5. Fernandes, C.M., Lima, C., Rosa, A.C.: UMDAs for dynamic optimization problems. In: 2008 Genetic and Evolutionary Computation Conference, pp. 399–406. ACM, New York (2008)
6. Grassé, P.-P.: La reconstrucion du nid et les coordinations interindividuelles chez bellicositermes et cubitermes sp. La théorie de la stigmergie: Essai d'interpretation du comportement des termites constructeurs. Insectes Sociétés (6), 41–80 (1959)
7. Holland, J.H.: Adaptation in natural and artificial systems. The University of Michigan Press, MIT Press, Ann Arbor (1975)
8. Larrañaga, P., Lozano, J.A.: Estimation of distribution algorithms: A new tool for evolutionary computation. Kluwer Academic Publishers, Boston (2002)
9. Pelikan, M., Goldberg, D.E., Cantu-Paz, E.: BOA: The Bayesian Optimization Algorithm. In: 1999 Genetic and Evolutionary Computation Conference, pp. 525–532. Morgan Kaufmann, San Francisco (1999)
10. Mühlenbein, H., Paass, G.: From recombination of genes to the estimation of distribution I, binary parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN IV 1996. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
11. Sastry, K.: Evaluation-relaxation schemes for Genetic and Evolutionary Algorithms. Msc Thesis, University of Illinois, Urbana, IL, USA (2001)
12. Zlochin, M., Biratari, M., Meulequ, N., Dorigo, M.: Modelbased search for combinatorial optimization: A critical survey. Annals of Operations Research 131(1-4), 373–395 (2004)

Niche Particle Swarm Optimization for Neural Network Ensembles

Camiel Castillo*, Geoff Nitschke, and Andries Engelbrecht

Department of Computer Science, University of Pretoria, South Africa
castillo.camiel@kpmg.nl, {gnitschke,engel}@cs.up.co.za

Abstract. This research investigates a swarm intelligence based multi-objective optimization algorithm for optimizing the behavior of a group of *Artificial Neural Networks* (ANNs), where each ANN specializes to solving a specific part of a task, such that the group as a whole achieves an effective solution. *Niche Particle Swarm Optimization* (NichePSO) is a speciation technique that has proven effective at locating multiple solutions in complex multivariate tasks. This research evaluates the efficacy of the NichePSO method for training a group of ANNs that form a neural network ensemble (NNE) for the purpose of solving a set of multivariate tasks. NichePSO is compared with a gradient descent method for training a set of individual ANNs to solve different parts of a multivariate task, and then combining the outputs of each ANN into a single solution. To date, there has been little research that has compared the effectiveness of applying NichePSO versus more traditional supervised learning methods for the training of neural network ensembles.

1 Introduction

In nature, biological systems such as ant and termite colonies optimize solutions to their tasks via having a set of simple individuals specialize to solving different (and complementary) parts of the problem [2]. A goal of artificial life is to replicate the mechanisms that allow groups of behaviorally simple individuals to cooperate in order to optimize solutions to complex tasks [5]. Particle Swarm Optimization (PSO) has close ties to artificial life models such as that of Reynolds [13] and Heppner [9], which indicated that emergent group dynamics such as bird flocking behavior are based on local interactions. These studies were the foundation for the development of PSO with applications that include industrial process control [11] and multi-objective function optimization [1].

Most evolutionary and swarm intelligence techniques are designed to converge on a single solution in a search space, where the quality of the solution depends on a task dependent fitness function. These techniques implicitly assume that only a single solution exists in the search space, and therefore that the search space is univariate. When presented with a multivariate task domain, typical

[*] Current work address: KPMG N.V. Burgemeester Rijnderslaan 10-20. 1185 MC Amstelveen. The Netherlands.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 399–407, 2011.

univariate techniques will either favor a single solution, or fail to converge due to the confusion introduced by multiple possible solutions [3]. Niching techniques attempt to overcome the deficiencies of univariate optimization techniques by explicitly assuming that multiple solutions exist in a search space.

This paper evaluates the efficacy of a PSO based niching method [3] compared with an established gradient descent method [14] for training Neural Network Ensembles (NNEs) [8] to solve a set of multivariate classification and regression tasks. To date, there has been little research that compares the effectiveness of using more traditional supervised learning techniques such as back propagation to train NNEs with more recent niche based (multi-population) swarm intelligence techniques such as that of Zhang *et al.* [15] and Brits *et al.* [3] to train NNEs. Results elucidate that NichePSO outperforms back propagation as a NNE training method for a majority of the multivariate classification and regression tasks. Traditionally, back propagation has been successfully applied as a supervised learning approach to train NNEs for solving such tasks [10]. Given this, the following research goal, hypotheses, and performance measure were formulated.

- **Research Goal:** To elucidate that the NichePSO algorithm [3] is able to outperform a back propagation algorithm [14] for training NNEs applied to solve a given set of multivariate classification and regression tasks.
- **Hypothesis 1:** For the given task set, back propagation is able to train a NNE such that the NNE outperforms each of its constituent ANNs.
- **Hypothesis 2:** For the given task set, NichePSO is able to train a NNE such that the NNE outperforms each of its constituent ANNs.
- **Hypothesis 3:** For the given task set, NichePSO is able to train a NNE such that it outperforms a back propagation trained NNE.
- **Performance Measure:** The portion of misclassified cases and the mean squared error, for classification and regression tasks, respectively.

2 Methods for Training Neural Network Ensembles

This section describes comparative methods evaluated for solving a given set of classification and regression tasks. These methods are: *Gradient Descent trained Ensembles* (GDE) and *Niche Particle Swarm Optimized Ensembles* (NPSOE). GDE uses back propagation to train a NNE, and NPSOE uses the NichePSO algorithm to train a NNE in order to solve a given task set. Previous research has indicated that when a single network is not capable of correctly representing a given data set, the fusion of a set of networks, each of which is specialized to a part of the data set, can significantly improve performance [8]. The key idea behind the performance increase yielded by NNEs, is that each network in the ensemble specializes to solving a complementary part of the task. Collectively, these specializations result in a task performance that is superior to that of a single ANN applied to solve the same task. From a behavioral perspective, an input layer is processed by all constituent ANNs of a NNE, and a *fusing* scheme is then applied in order to combine the outputs of each ANN into one NNE output layer [8]. This research uses a uniformly weighted output scheme for the regression tasks, and a majority voting scheme for classification tasks [10].

When training a NNE using GDE or NPSOE (for either a classification or regression task), the input layer consisted of attributes from training data. This input layer was split over the constituent ANNs of the NNE. For measuring the performance of a trained NNE, the validation data was passed as the input layer to the NNE, and NNE performance compared to NNE performance using training data. An average task performance was calculated over multiple runs.

Table 1. GDE and NPSOE method parameter settings

GDE and NPSOE Method Parameter Settings	
Number of hidden nodes (NPSOE / GDE)	8
Input / hidden node transfer function (NPSOE / GDE)	linear
Output node transfer function (NPSOE / GDE)	sigmoid
Learning rate (GDE)	0.001
Momentum (GDE)	0.01
Iterations (NPSOE / GDE)	50000
Number of particles (NPSOE)	30
Weight range (NPSOE / GDE)	[-1.0, +1.0]
Number of networks (NPSOE / GDE)	7
Number sub-swarms (NPSOE)	7
Initial ρ (NPSOE)	0.1
ρ increment value (NPSOE)	15
ρ decrement value (NPSOE)	5

Both the GDE and NPSOE methods used a homogeneous ensemble, meaning each of the constituent ANNs was the same. The number of input neurons used by each ANN equalled the number of attributes that were being passed as the input layer for a given classification or regression task. The number of outputs always equalled one, which was the prediction or classification value. Hence, the value type of the input and output neurons depended on the value of the attributes being used by a given classification or regression task. For both methods, prior to training, the weights of each ANN were randomly initialized within the range [-1.0, 1.0]. Also, for both methods, the fusion of each of the outputs of each ANN was done according to a majority voting [8] or weighted average [12] scheme for classification and regression tasks respectively. Table 1 presents the parameter values used by the GDE and NPSOE methods. These values were derived in a set of exploratory experiments, and minor changes to these parameter values produced similar results for both GDE and NPSOE applied to all tasks.

2.1 GDE Method: Back Propagation Trained Ensemble

Figure 1 illustrates the architecture of the method for training a NNE with back propagation. First, the training data is given to the input layer of the NNE. Each ANN is then trained by a back propagation algorithm [14]. The validation data is then passed to the NNE input layer, the output is compared with that produced by the training data, and the weights of the NNE are adapted accordingly.

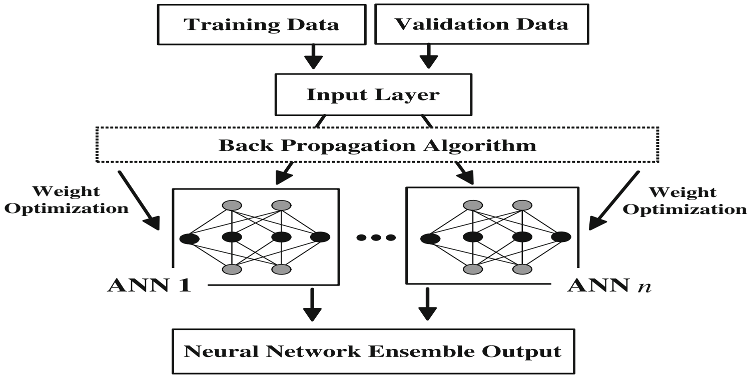


Fig. 1. Architecture of back propagation trained neural network ensemble

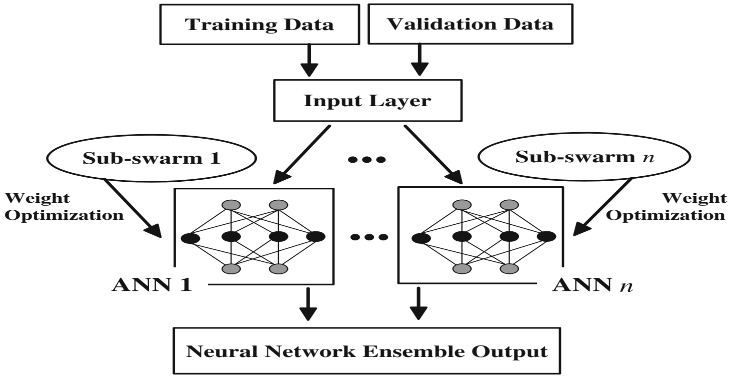


Fig. 2. Architecture of the NichePSO trained neural network ensemble

2.2 NPSOE Method: NichePSO Trained Ensemble

NichePSO [4] is a niche-based PSO method that dynamically creates sub-swarms that converge upon multiple newly discovered optima in the search space. The function of the initial main swarm is thus to continually explore the search space. Figure 2 illustrates the architecture used for training a NNE with NichePSO (NPSOE). For the experiments presented in section 3, NPSOE derived one sub-swarm in order to optimize the weights of each constituent ANN. Each sub-swarm particle represents the weight vector of a given ANN. Each sub-swarm uses the GCPSO algorithm [6] as the particle velocity update strategy. GCPSO was selected since it has been demonstrated to work well with low particle numbers, and is more appropriate for exploitation than exploration. In order to train each ANN the training data is passed to each ANN as the input layer, and the output is compared to that produced when the validation data is passed. The error is used as a fitness value by the NichePSO algorithm. The weights of each

ANN were set according to each sub-swarm's best particle. For a complete description of the NichePSO algorithm refer to Brits *et al.* [4].

3 Experiments

The performance of the GDE and NPSOE methods were both evaluated on five regression, and five classification tasks. The performance of each method was measured as the number of misclassified cases for classification tasks, and mean squared error for regression tasks. For the performance evaluation of each method, the validation set, and not the training set, was used. For each method applied to each task, 20 simulation runs were executed, and results presented are averages of these 20 runs. Method parameters for all classification and regression tasks are as presented in table 1. With the exception of the *random pattern classification* task, and the *Friedman#1* synthetic data set used for the first regression task, the data sets used for all classification and regression tasks were taken from the *UCI Machine Learning Repository*[1].

3.1 Classification Tasks

- **Classification Task 1: Random pattern classification:** The classification of random patterns task investigated by Hansen and Salamon [8] is used. There were 1000 training and 400 validation patterns. Each pattern was a real valued input vector with 20 attributes. Given this training set, the task was to correctly classify each validation pattern to one of the five classes.
- **Classification Task 2: Ozone Level Detection:** This data set uses 2536 instances. Each instance contains 73 attributes. The task is to correctly classify an ozone reading given a set of environment related attributes.
- **Classification Task 3: Abalone:** This data set consists of 4177 data patterns, each with eight real valued attributes. The task is to correctly classify abalones as belonging to a particular age range, given a set of attributes.
- **Classification Task 4: Wine:** The wine data set contains 178 types of wine, each with 13 real valued attributes. The attributes represent physical characteristics of the wines. All wines belong to one of three classes. The task is to correctly classify each wine to the correct class.
- **Classification task 5: Glass:** The glass database consists of 214 patterns each representing a piece of glass. Each pattern contains ten real valued attributes. The task is to classify each piece of glass as crime scene processed, or not, for a given set of attributes.

3.2 Regression Tasks

- **Regression Task 1: Friedman #1:** The Friedman#1 synthetic data set [7] corresponds to training vectors with five input and one output variable. The data set was created by randomly generating real valued input vectors

[1] http://archive.ics.uci.edu/ml/datasets.html

with attributes in the range [0.0, 1.0], and computing a corresponding output [7]. A set of 1200 patterns was split into a training set baring 1000 instances, and the remaining 200 patterns were assigned to the validation set.
- **Regression Task 2: MPG Auto:** This data set consists of 398 instances, having eight real valued attributes. The task is to determine the fuel consumption of cars with given attributes.
- **Regression Task 3: Computer Hardware:** This data set contains 209 instances, where each instance has nine integer attributes. The task is to predict the relationship between hardware and performance given a set of computer hardware attributes.
- **Regression Task 4: Servo:** The servo data set consists of 167 instances, where each instance has two continuous and two discrete attributes. The task is to predict a a servo-mechanism rise time (the time required for the mechanism to respond to a change in position) given a set of attributes.
- **Regression Task 5: Wisconsin breast cancer:** The Wisconsin breast cancer data set comprises 198 instances each with 34 real valued attributes. The task is to predict cancer (benign or malignant) given a set of attributes.

4 Results and Discussion

Table 2 presents the results of an independent t-test (0.95 confidence value) applied in order to test for a statistically significant difference between the average task performance results of GDE and NPOSOE applied to each of the classification and regression tasks. That is, the NPSOE method out-performs GDE on three out of the five classification tasks and four out of the five regression tasks.

Hypothesis 1: T-tests applied to misclassification and mean square error results of the NNE and each of the constituent ANNs trained by back propagation indicates the following significant differences. For the classification tasks, it is only for the *random pattern* classification task, that back propagation trained NNEs are able to out-perform each of their constituent ANNs (table 2). For the regression tasks, the *Friedman #1*, *Wisconsin Breast Cancer*, and *MPG Auto* trained NNEs out-perform each of their constituent ANNs (table 2).

Hypothesis 2: T-tests applied to misclassification and mean square error results of the NNE and constituent ANNs trained by NichePSO indicates the following significant differences. For the classification tasks, NichePSO trained NNEs out-perform their constituent ANNs for the *random pattern*, *abalone*, *wine*, and *Ozone Level Detection* tasks (table 2). NichePSO trained NNEs out-perform each of their constituent ANNs for all regression tasks (table 2).

Hypothesis 3: T-tests applied to misclassification and mean square error results of the back propagation (GDE method) and the NichePSO (NPSOE method) trained NNEs indicates the following significant differences. For the classification tasks, NPSOE out-performs GDE for the *random pattern*, *glass* and *Ozone Level Detection* tasks (table 2). However, for the *abalone*, and *wine* tasks, both methods

Table 2. Overview of acceptance or rejection of hypotheses for the results of GDE and NPSOE when applied to each of the classification and regression tasks

Classification Tasks	**Hypothesis 1**	**Hypothesis 2**	**Hypothesis 3**
	Accept/Reject	**Accept/Reject**	**Accept/Reject**
Random Pattern Classification	Accepted	Accepted	Accepted
Glass	Rejected	Rejected	Accepted
Abalone	Rejected	Accepted	Rejected
Wine	Rejected	Accepted	Rejected
Ozone Level Detection	Rejected	Accepted	Accepted
Regression Tasks	**Hypothesis 1**	**Hypothesis 2**	**Hypothesis 3**
	Accept/Reject	**Accept/Reject**	**Accept/Reject**
Friedman #1	Accepted	Accepted	Accepted
Wisconsin Breast Cancer	Accepted	Accepted	Rejected
MPG Auto	Accepted	Accepted	Accepted
Computer Hardware	Rejected	Accepted	Accepted
Servo	Rejected	Accepted	Accepted

Table 3. Average portion of misclassified cases (classification) and mean squared error (regression), and standard deviation in parentheses for GDE and NPSOE

Classification Tasks	**GDE**	**NPSOE**
Random Pattern Classification	0.33 (0.022)	0.291 (0.02)
Glass	0.378 (0.157)	0.291 (0.074)
Abalone	0.374 (0.01)	0.432 (0.016)
Wine	0.646 (0.068)	0.652 (0.047)
Ozone Level Detection	0.648 (0.015)	0.567 (0.02)
Regression Tasks	**GDE**	**NPSOE**
Friedman #1	2.169 (0.055)	0.249 (0.128)
Wisconsin Breast Cancer	29.217 (5.644)	51.413 (16.256)
MPG Auto	63.059 (0.018)	41.745 (16.938)
Computer Hardware	60.578 (6.299)	54.625 (6.853)
Servo	1.211 (0.228)	1.019 (0.006)

yield comparable results. NPSOE out-performs GDE for all of the regression tasks, except the *Wisconsin Breast Cancer* task. A statistical comparison of mean squared indicated comparable results for this task (table 2).

These results indicate that NichePSO (hypothesis 2) comparative to back propagation (hypothesis 1) is an appropriate method for training NNEs. That is, NichePSO trained NNEs outperform each of the constituent ANNs trained by NichePSO for 90% of the tasks. Where as, back propagation trained NNEs outperform each of the constituent ANNs trained by back propagation for only 40% of the tasks. Regarding hypothesis 3, results indicate that the NichePSO trained NNE (NPSOE), comparative to a back propagation trained NNE (GDE) is appropriate for solving the given set of multivariate classification and regression tasks. That is, NPSOE outperforms GDE with a statistically significant difference for 70% of the tasks. Table 3 presents the task performance results for GDE and NPSOE applied to the classification and regression tasks.

5 Conclusions

This research was an initial step for establishing NichePSO as being an appropriate algorithm for training neural network ensembles to solve complex multivariate tasks that require different networks to specialize to solve complementary parts of the task. This paper presented a set of multivariate classification and regression tasks. Such tasks have typically been solved via applying gradient descent algorithms to train neural networks or neural network ensembles. Results indicated that a neural network ensemble trained with NichePSO was able to exploit the multivariate nature of these tasks, which in turn lead to a significantly lower classification and prediction error rate compared to the back propagation trained ensemble. Given that NichePSO trained neural network ensembles have been successful at solving more the classification and regression tasks presented in this paper, the approach has potential applications to complex artificial life oriented tasks. For example, automating the design of a group of agent neural controllers such that the controllers develop specialized and complementary behaviors and a collective group behavior is produced that solves a given task.

References

1. Blum, C., Li, X.: Swarm intelligence in optimization. In: Swarm Intelligence - Introduction and Applications, pp. 43–85. Springer, Heidelberg (2008)
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford (1998)
3. Brits, R., Engelbrecht, A., van den Bergh, F.: A niching particle swarm optimizer. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002 (SEAL 2002), pp. 692–696. MIT Press, Singapore (2002)
4. Brits, R., Engelbrecht, A., van den Bergh, F.: Locating multiple optima using particle swarm optimization. Applied Mathematics and Computation 189(1), 1859–1883 (2007)
5. Darwen, P., Yao, X.: Every niching method has its niche: fitness sharing and implicit sharing compared. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN IV 1996. LNCS, vol. 1141, pp. 398–407. Springer, Heidelberg (1996)
6. Engelbrecht, A., van den Bergh, F.: A new locally convergent particle swarm optimizer. In: Proceedings of the IEEE Conference on Systems, Man and Cybernetics, pp. 96–101. IEEE Press, Los Alamitos (2002)
7. Friedman, J., Grosse, E., Stuetzle, W.: Multidimensional additive spline approximation. Journal of Scientific and Statistical Computing 4(1), 291–301 (1983)
8. Hansen, L., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(10), 993–1001
9. Heppner, F., Grenander, U.: A stochastic nonlinear model for coordinated bird flocks. In: Krasner, S. (ed.) The Ubiquity of Chaos, pp. 1–10. AAAS Publications, Washington (1990)
10. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: Advances in Neural Information Processing Systems, pp. 231–238. MIT Press, Cambridge (1995)

11. Li, L., Zhou, J., Yu, X., Li, X.: Constrained power plants unit loading optimization using particle swarm optimization algorithm. WSEAS Transactions on Information Science and Applications 4(2), 296–302
12. Munro, P., Parmanto, B.: Competition among networks improves committee performance. In: Mozer, M., Petsche, J. (eds.) Advances in Neural Information Processing Systems, pp. 592–598. MIT Press, Cambridge (1997)
13. Reynolds, C.: Flocks, herds and schools: A distributed behavioral model. Computer Graphics 21(4), 25–36 (1987)
14. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by error propagation: Parallel Distributed Processing. Foundations, vol. 1. MIT Press, Cambridge (1986)
15. Zhang, J., Zhang, J.-R., Li, K.: A sequential niching technique for particle swarm optimization. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 390–399. Springer, Heidelberg (2005)

Chaotic Patterns in Crowd Simulation

Blanca Cases, Francisco Javier Olasagasti, Abdelmalik Moujahid,
Alicia D'Anjou, and Manuel Graña

Computational Intelligence Group of the UPV/EHU
Facultad de Informática,
P. Manuel de Lardizabal 1, 20016 San Sebastián, Spain
{blancarosa.cases,abdelmalik.moujahid,alicia.danjou}@ehu.es,
javierol@teleline.es

Abstract. We have developed a model of crowd based on social agents and swarm intelligence. The model takes the form of a randomly generated directed graph where nodes represent individuals locked in a room where a fire occurs. Each individual follows connected individuals, called the references. There are two special individuals, the firefighters, situated at the two exits from the room: exit T or true and exit F or false. The agents are directed at targets T and F in a computer animation implemented in Netlogo. They come into conflict of over-information when they receive contradictory information from their references.

We studied experimentally the influence of the following mechanisms of conflict resolution: follow the mode or the anti-mode, random resolution and the effect of excluding the own opinion. We found that these mechanisms lead people following the mode to unanimously choose one of the options, flocking towards the selected goal. In the case of anti-mode, the population oscillates between the two options. The number of references is critical to this behavior and following one or two references with exclusion of the own state leads the system to chaotic patterns of convergence.

Keywords: crowd simulation, swarm intelligence, complex systems, computational intelligence, chaos theory, collective decision-making.

1 Introduction

A conduct that necessarily leads to the formation of social groups without any internal representation of the world is tracking mimetic behavior [1,2,3,4,5]. We can easily imagine this through examples: in a situation of economic crisis, two media A and B respectively advise to draw or not the money from banks, or promote the vote for a candidate in an election. In the field of crowd psychology, we can imagine a situation like a fire, where firefighters are situated at the emergency exits A or B to evacuate the room. Let us identify option A to boolean value T (true) and B to F (false). T and F agents are sources that remain in a fixed position and do not receive any input. We describe the mechanism of a stampede: In the face of danger, the agents create references, i.e. they choose

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 408–415, 2011.

a number of agents to follow. Assume for simplicity that all of them select the same number of references. Those agents close to the sources T and F, will move towards these areas, but those that are a little farther will set up camp in the direction of their neighbors following the stream to zone T or to F. The conflict arises when the agents at the border of the two basins are required to move in opposite directions, or when they go in one direction and see other agents running in the opposite direction. These agents enter in conflict and hence stop.

The way to get out of conflict is to create new references to informed choices that will lead to the goal T or to F. In this work we assume randomly created references. Figure 1 graphically expresses this self-organizational logic in which the system moves from conflict to the information values T or F.

We might ask what kind of collective choice made the agents: from observing the environment, they develop a decision over a process that lasts until the changes of opinion of the environment no longer occur (convergence). The final counting of agents in each state, gives us a collective choice. Note that to solve the problem of evacuating the room, the best solution is to split the crowd in two equilibrated groups.

Our simulation shows that this solution is anecdotal if conflict crowds following the Mode (the most common direction in the neighborhood) or the Anti-Mode (the less common). Judging from the average results would appear that the group is divided into two homogeneous sub-groups, but on the contrary, we obtain dynamics that lead to a unanimous choice as well as dynamics that show periodic or chaotic patterns of convergence.

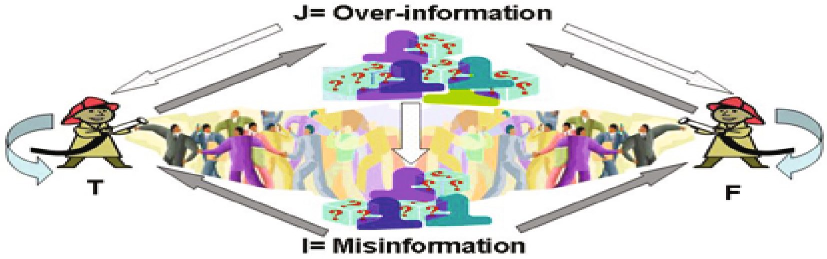


Fig. 1. Interpretation of Diamond logic. Resolution of conflict J ensures the convergence to informed values T or F.

2 Outline of Model Formalization

Each agent, who is called R_i, can identify a maximum number k of agents that we call their references, $R_{i1} \dots R_{ij} \quad 1 \leq j \leq k$. This reference network is a directed graph created randomly. The agents are initially in the state denoted by the symbolic value of misinformation $I =$ *"Does not know."*. The boolean values T="true" and the F= "false" represent information. The conflict due to over-information is expressed by the value J = "Does not answer", meaning that an agent is receiving two contradictory pieces of information.

The transmission of messages and therefore the movement of the group was formalized through the Diamond Logic [7,8], a tetra-valued extension of the propositional logic that allows for the formulation of assertions in the form of self-referential equations. The set of values is the logic Diamond D = (T, F, I, J), where T and F are the Boolean values True (true) and false (false), while I and J are meta-logical values denoting paradox, both solutions of the auto-denied equation $X = not(X)$.

The transmission is modeled by a self-referential equation associated with each agent:

$$R_i = state - operator(R_i, R_{i1}, ..., R_{ik})$$

which is interpreted as follows: The R_i, R_{ij}, are variables that identify the agents taking their values in the diamond D. The value of R_i at time $t+1$ depends on the action itself and what they hear from their references at the moment t just before.

We study experimentally the following dynamics: they are all identical in the absence of conflict, the values T, F have priority over misinformation I. When conflict by over-information J arises, the differences appear.

- BASIC: The concurrence of contradictory information, T and F, represents an excess of information expressed as a conflict J and that's going to override all other values.
- MODE: In the event that the basic conflict J occurs, the agent will choose the more frequent state T or F. In case of a tie, remain in conflict J.
- ANTI-MODE: Like operator mode but selecting the less frequent state T or F.

These dynamics are analyzed in two ways:

- Include own opinion: $R_i = state - operator(R_i, R_{i1}, ..., R_{ik})$.
- Exclude the opinion: $R_i = state - operator(R_{i1}, ..., R_{ik})$.

Regarding conflicts J there are two options:

- Not random resolution: Doesn't reset the individual in conflict, maintaining the value J. Random.
- Random resolution: Reset the agent with new references and initialize to the state of misinformation I.

Basic dynamic of transmission corresponds to the rule "the more the information, the more the priority of transmission". Note that the value J represents an excess of information being transmitted with the maximum priority. Given a list of logical values $m = v_1, v_2, \ldots, v_n$ with $n \geq 2$, let $n_I(m)$, $n_T(m)$, $n_F(m)$, $n_J(m)$ be the number of occurrences of the sub-indexed value in the list.

Basic	T	F	I	J
T	T	J	T	J
F	J	F	F	J
I	T	F	I	J
J	J	J	J	J

$$Basic(m) = \begin{cases} J \ (n_T(m) > 0 \ \wedge \ n_F(m) > 0) \ \vee n_J(m) > 0 \\ T \ n_T(m) > 0 \ \wedge \ n_F(m) = 0 \ \wedge \ n_J(m) = 0 \\ F \ n_T(m) = 0 \ \wedge \ n_F(m) > 0 \ \wedge \ n_J(m) = 0 \\ I \ n_T(m) = 0 \ \wedge \ n_F(m) = 0 \ \wedge \ n_J(m) = 0 \end{cases}$$

J is transmitted with the highest priority and means something like a warning to others: be careful, I'm following one who is still missing. For example, given the following system of self-referential equations and iterating from the initial condition I assigned all the variables, we have the following stages until the system reaches a fixed point:

R1= T
R2= F
R3= Basic(R3 , R1 , R4)
R4= Basic(R4 , R3 , R5)
R5= Basic(R5 , R4 , R6)
R6= Basic(R6 , R2 , R5)

	R1	R2	R3	R4	R5	R6
t=0	I	I	I	I	I	I
t=1	T	F	I	I	I	I
t=2	T	F	T	I	I	F
t=3	T	F	T	T	F	F
t=4	T	F	T	J	J	F
t=5	T	F	J	J	J	J
t=6	T	F	J	J	J	J

3 A Netlogo Implementation of the Model: Comparing the Dynamics of the Mode and the Anti-mode

In order to understand the above concepts, we developed a simulation on Netlogo[1]. The button setup creates a random graph with as many nodes as number-of-agents and as many sources of each class, T or F, as specified in the respective inputs sources-T or sources-F. The dynamics can be situated geographically or not: Neighborhood input variable specifies the size of the environment around an agent. The world positions vary between coordinates (-10, -10) and (10, 10) leading to 21 x 21 patch positions in a square.

The kind of dynamics produced by Basic state operator in non situated experiments (the neighborhood of an agent is the whole world) is typically: **(a** a final configuration where all the agents stop in conflictive state J, **(b** Activating mode as a method of resolution of conflicts the typical final state is such that the crowd unanimously select goals T or F respectively. This is a surprising result since one can expect heterogeneous groups of agents in final states T, F and J. In the case of anti-mode method of conflict resolution, the mean results are similar to the ones obtained for the mode, but the crowd normally enters in a periodic point and individuals oscillate between the two goals.

In Figure 2 we compare the extent of information for operators Basic, Mode and Anti-mode in a population of 100 agents by varying the number of references parameter from 1 to 10 and conducting 100 runs for each experiment. The criterion for stopping the system is either reaching a fixed point or completing 100 steps of iteration. One can observe that while conflict is dominant in the basic dynamics, information becomes dominant in the Mode and Anti-Mode models reaching T value about the 50%.

[1] Available at http://www.ehu.es/ccwintco/index.php/Sociedades_Artificiales_--_Artificial_Societies

Table 1. The first part of the table shows the average results over 100 agents and 100 runs including own opinion. The second part presents the results of the same models excluding own opinion.

Basic	Steps	I	T	F	J	Steps	I	T	F	J
Mean	4.84	8.70	2.86	2.32	86.12	4.97	7.94	3.87	3.98	84.22
Standard deviation	2.72	26.79	14.96	13.58	34.55	2.84	25.70	17.97	18.53	36.44
Mode	Steps	I	T	F	J	Steps	I	T	F	J
Mean	6.95	7.94	47.09	44.97	0.00	11.66	8.32	45.88	45.80	0.00
Standard deviation	3.18	25.44	47.89	47.76	0.00	20.82	26.22	48.53	48.47	0.00
Anti-Mode	Steps	I	T	F	J	Steps	I	T	F	J
Mean	87.59	7.91	46.74	45.35	0.00	85.12	8.51	45.80	45.70	0.00
Standard deviation	31.77	25.36	28.13	27.74	0.00	33.98	26.65	28.18	28.29	0.00

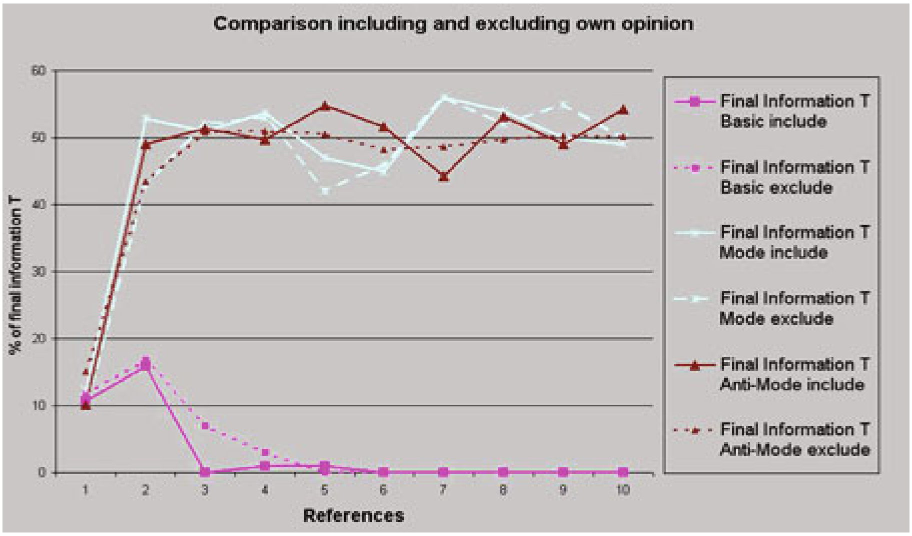


Fig. 2. Comparison of the transmission of information according to the model. The independent variable is the number of references. The dependent variable counts the number of agents in state T at the end of the experiment.

Since T and F values are symmetric, misinformation and over-information are reduced to nearly 0% from 2 references avoiding conflict. With respect to the time of convergence, Mode and Anti-Mode dynamics reflect the same curve than Basic, although the maximum shifts to three references, as shown in Figure 3.

Concerning to the average values over all experiments, the following table describes the mean and deviation obtained on each variable. Note that the dynamics Mode and Anti-Mode completely eliminate the conflict and obtain unanimous dynamics navigating to T and to F, as we can infer from the values of standard deviation in table 1 of 1.

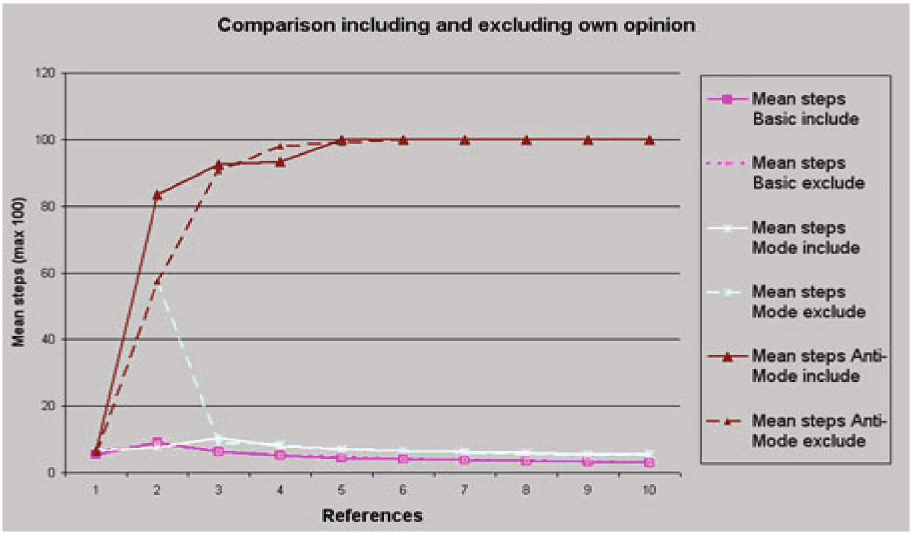


Fig. 3. Number of iterations until convergence or termination of the experiment (100 steps) comparing the three models

4 Discussion: On the Role of the Own Opinion

We have studied experimentally the mode and anti-mode mechanisms of conflict resolution. Note that mode amplifies the information value most frequent in the references of the agent, and in this way we can expect an unanimous selection of the group. Anti-mode makes just the opposite, selecting the less frequent information, but the emergent behavior is similar in mean to the case of mode operator.

For more than 3 references, the basic model ends in configurations of total conflict J, while following 1 or 2 references implies less transmission of information values over misinformation I. This suggests that the percolation index of the graph [9] has to do with this behavior of the system.

Interpreting our model as a generalization of cellular automata [5], the kind of computations obtained in the experiments described before correspond to the classes I (short transient lengths and fixed points), II (greater transient lengths and periods) and III (long transient lengths and short periods) of the Wolfram-Langton hierarchy.

We wonder if the exclusion or inclusion of the own opinion in the model has importance to obtain behaviors at the edge of chaos. The number of references is critical to this behavior and following one or two references with exclusion of the own opinion leads the system to chaotic patterns of convergence. Results of experimentation excluding own opinion (with identical parameters to experimentation presented before) are given in table 2 of [1].

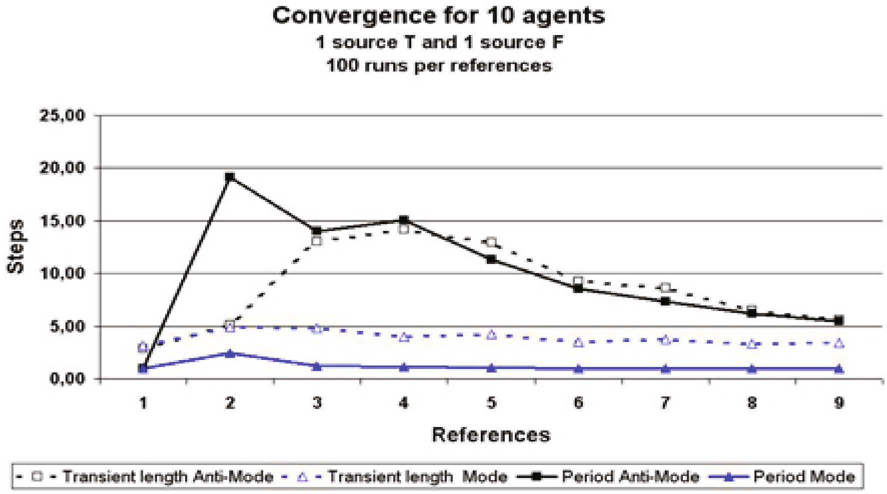


Fig. 4. Comparing the period and the transient-lengths following the mode and de the anti-mode for 10 agents

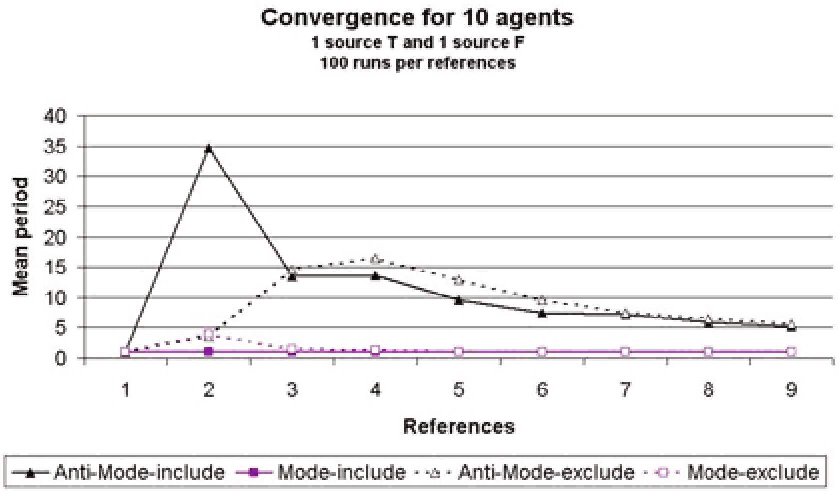


Fig. 5. Comparing the period and the transient-lengths following the mode and de the anti-mode for 10 agents

To explore completely the periodic patterns of convergence, we have repeated the experiments for 10 agents with one source T and one F and 100 runs per experiment. States operators Mode and Anti-Mode were compared, excluding and including own opinion and exploring the whole range from 1 to 9 references. Experimentation has shown that following the anti-mode produces more variability than following the mode as show figures 4 and 5. The values of periods and transient lengths are significantly greater for the anti-mode state operator. Following the Mode with 2 references per agent and excluding the own opinion makes arise computations at the edge of chaos (type III), while the Anti-Mode produces the same behavior either excluding or including the own opinion. Summarizing, conflict crowds, but paradoxically, solving the problem of splitting the crowd in two equilibrated groups that navigate to the exits occurs in a tiny amount of cases.

References

1. Chen, Z.: Computational Intelligence For Decision Support. The Crc Press LLC, U.S.A (2000)
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural To Artificial Systems. Oxford University Press, Oxford (1999)
3. Camazine, S., Deneubourg, J.L., Franks, N., Sneyd, J., Bonabeau, E., Theraulaz, G.: Self-Organization In Biological Systems. Princeton University Press, Princeton (2001)
4. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Academic Press, U.S.A (2001)
5. Schut, M.C.: Scientific Handbook for Simulation of Collective Intelligence (2000), Edicin electrnica en http://sci.collectivae.net/
6. Langton, C.G.: Life At The Edge Of Chaos. In: Langton, C.G. (ed.) Artificial Life II, pp. 41–90. Addison Wesley, Reading (1990)
7. Hellerstein, N.S.: Diamond, A Logic Of Paradox. In: CYBERNETICS, vol. 1(1). American Society For Cybernetics, USA (Summer-Fall 1985)
8. Hellerstein, N.S.: Diamond, A Paradox Logic. World Scientific Publishing Company, Singapore (1997)
9. Grimmett, G.: Percolation, 2nd edn. Springer, Heidelberg (1999)
10. Cases, B., Hernandez, C., Graa, M., D'Anjou, A.: On The Ability Of Swarms To Compute The 3-Coloring Of Graphs. In: En Bullock, S., Noble, J., Watson, R., Bedau, M.A. (eds.) Artificial Life XI: Proceedings Of The Eleventh International Conference On The Simulation And Synthesis Of Living Systems, MIT Press, Cambridge (2008)

Simulating Swarm Robots for Collision Avoidance Problem Based on a Dynamic Bayesian Network

Hiroshi Hirai, Shigeru Takano, and Einoshin Suzuki

Department of Informatics, ISEE, Kyushu University, Fukuoka 819-0395, Japan
2IE09034N@s.kyushu-u.ac.jp, {takano,suzuki}@inf.kyushu-u.ac.jp
http://www.i.kyushu-u.ac.jp/~suzuki/slabhomee.html

Abstract. This paper presents a simulator for the behaviors of swarm robots based on a Dynamic Bayesian Network (DBN). Our task is to design each robot's controller which enables the robot to patrol as many regions as possible without collisions. As the first step, we use two swarm robots, each of which has two motors each of which is connected to a wheel and three distance-measurement sensors. To design the controllers of these robots, we must determine several parameters such as the motor speed and thresholds of the three sensors. The simulator is used to reduce the number of real experiments in deciding values of such parameters. We fist performed measurement experiments for our real robots in order to get probabilistic data of the DBN. The simulator based on the DBN revealed appropriate values of a threshold parameter and interesting phase transitions of their behaviors in terms of the values.

Keywords: Swarm Robots, Dynamic Bayesian Network, Simulation.

1 Introduction

Recently, swarm robots have been attracting a lot of attention in the research community since they possess advantages such as scalability, flexibility, robustness, and cost-performance [4]. It is a difficult problem to design a controller of swarm robots which accomplish given tasks as a system due to uncertainties of robots, the complex interaction among robots, and their limited capabilities.

We aim at developing each robot's controller which enables the robot to patrol as many regions as possible without collisions. As a simple case, we assume two robots and one parameter that is a threshold of the robot's sensor. Finding a good value for the parameter necessitates us a large number of experiments and thus we use a simulator.

M. Toussaint [5] recently proposed to use a Dynamic Bayesian Network (DBN) for planning problems of a manipulator robot. The DBN is a probabilistic model, which consists of a set of states and probabilistic information between the states. In his work, DBN returns the joint probability distribution of the states of the robot. As the final state is given in the planning problem unlike in our collision

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 416–423, 2011.

avoidance problem, we can not directly apply his method to our problem. We therefore propose another type of the DBN to model the behaviors of the swarm robots for our task.

2 Collision Avoidance Problem

2.1 Definition

Our problem is to design controllers for swarm robots so that they patrol as many regions as possible without collisions with other robots or wall. We put $\boldsymbol{D}$ as a $a \times a$ closed domain in the 2 dimensional space in which the swarm robots can move, where a represents the size of one edge of $\boldsymbol{D}$. At time $t \in [1, T]$, a variable $\boldsymbol{x}_t^i$ is the state of the i-th robot and its action is represented by a_t^i. Similarly, a variable s_t^i is a value of distance-measurement sensors of the i-th robot at time t. The action of each of the swarm robots is determined by a controller δ_i as follows:

$$\delta^i \left(\boldsymbol{s}_t^i\right) = a_t^i.$$

In order to evaluate the performance of the program δ_i, we consider to divide the domain $\boldsymbol{D}$ into $b \times b$ square sub-domains, where b represents the size of one edge of the sub-domain, and define a probability of visiting each sub-domain as $\boldsymbol{P}_{\mathrm{S}earch}$. and a probability of collisions with another robot in each sub-domain as $\boldsymbol{P}_{\mathrm{C}ollision}$. Note that these probabilities are defined in terms of the two robots from $t = 1$ to the time of a collision. In this paper, each of the swarm robots determines its action based on the values of the distance-measurement sensors and the values depend on their states. Our task is to find an appropriate value for the threshold so as to increase $\boldsymbol{P}_{\mathrm{S}earch}$ and decrease $\boldsymbol{P}_{\mathrm{C}ollision}$. To determine the value, we use a simulator to imitate the real behaviors of the two robots. We store the observed behaviors as the log records of their positions and their angles in time sequence. From the log data, we calculate $\boldsymbol{P}_{\mathrm{S}earch}$ and $\boldsymbol{P}_{\mathrm{C}ollision}$.

2.2 Controller

In this paper, we put a set of the states of the robot at time t as $\boldsymbol{X}_t$, so $\boldsymbol{X}_t = \{s_t^1, s_t^2\}$. Concretely, $s_t^i = (\boldsymbol{p}_t^i, d_t^i)$, where $\boldsymbol{p}_t^i$ and $d_t^i \in [0, 360)$ represent the position and the angle of the i-th robot at time t, respectively. In the rest of the paper, we may omit i and t if they are clear from the context.

Swarm robots are autonomous agents each of whom interacts with their environment locally based on a relatively simple program but as a system they exhibit complex collective behaviors. As the first step, we tackle a simple problem to study the feasibility of our approach by understanding its essence without being overwhelmed by a complex problem. We use the same controller for the two robots, i.e., each controller δ^i for determining the action of the i-th robot is given below.

```
Controller of each robot
while(1)
{
  s = GetSensorValue();
  if (s > threshold)
    turn right;
  else
    move forward;
}
```

In the controller, GetSensorValue() returns the average value of the distance-measurement sensors. Each of the robots can only perform two kinds of actions: move forward and turn right to avoid the collision with another robot or a wall.

3 Simulator Based on the Dynamic Bayesian Network

A simulator may enable a quick and easy design of the controller of the swarm robots. To create the swarm robots' simulator, we must know how each robot moves for each action. Unfortunately, our robots can not necessarily move forward or turn right as ordered in the robot's program. Moreover, the values of our sensors include noise, i.e., a sensor may overlook or mis-detect an object. Therefore, we model the uncertain behaviors of such robots as a DBN, which has been successfully used in [5] as a model of a manipulator robot in planning problems.

3.1 Dynamic Bayesian Network

A DBN [3] consists of a pair of Bayesian Networks $(B_1, B_\rightarrow)$, where B_1 defines the prior probability distribution $\boldsymbol{P}(Z_1)$ over the initial state at time $t = 1$, and $B_\rightarrow$ defines the the conditional probability distribution $\boldsymbol{P}(Z_t|Z_{t-1})$ over the states at previous-time $t-1$ to current-time t. Here Z_t represents all probability variables at time t. In this paper, we assume that the $\boldsymbol{P}(Z_t|Z_{t-1})$ is realized in time $t > 1$.

3.2 Structure of Our DBN

We propose the following DBN as a model of n swarm robots. The initial Bayesian Network B_1 has a set of nodes which represent probabilistic state variables $A_1^1, \cdots, A_1^n$, $\boldsymbol{X}_1^1, \cdots, \boldsymbol{X}_1^n$, and a set of arrows which represents these state transitions. Here A_t^i represents the set of actions of the i-th robot at time t. The state transitions are given by $\boldsymbol{P}(A_1^i|\boldsymbol{X}_1^1, \cdots, \boldsymbol{X}_1^n)$ in each of $i \in [1, n]$. Similarity, Bayesian Network $B_\rightarrow$ has a set of probabilistic state variables $A_{t-1}^1, \cdots, A_{t-1}^n$, $\boldsymbol{X}_{t-1}^1, \cdots, \boldsymbol{X}_{t-1}^n$, $A_t^1, \cdots, A_t^n, \boldsymbol{X}_t^1, \cdots, \boldsymbol{X}_t^n$, and a set of arrows which represent the state transitions given by $\boldsymbol{P}(A_{t-1}^i|\boldsymbol{X}_{t-1}^1, \cdots, \boldsymbol{X}_{t-1}^n)$, $\boldsymbol{P}(A_t^i|\boldsymbol{X}_t^1, \cdots, \boldsymbol{X}_t^n)$ and $\boldsymbol{P}(\boldsymbol{X}_t^i|A_{t-1}^i, \boldsymbol{X}_{t-1}^i)$ in each of $i \in [1, n]$. An example of the structure of the proposed DBN model in case of $n = 2$ is as shown in Fig. 1.

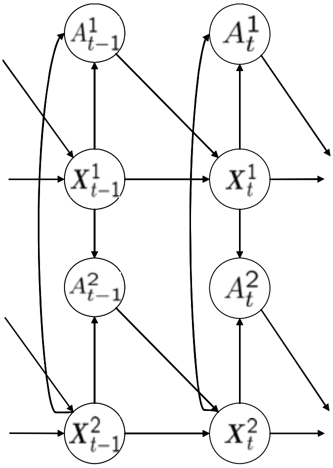


Fig. 1. DBN model in case of 2 robots

Here we introduce the following two assumptions. One is that the states of each robot follow the same conditional probability distribution, that is, the following equation is realized:

$$\boldsymbol{P}(A^i_t|\boldsymbol{X}^1_t, \cdots, \boldsymbol{X}^n_t) = \boldsymbol{P}(A^j_t|\boldsymbol{X}^1_t, \cdots, \boldsymbol{X}^n_t),$$

where $i, j \in [1, \cdots, n], i \neq j$. Similarity, the other assumption is that the following equation is realized:

$$\boldsymbol{P}(\boldsymbol{X}^i_t|A^i_{t-1}, \boldsymbol{X}^i_{t-1}) = \boldsymbol{P}(\boldsymbol{X}^j_t|A^j_{t-1}, \boldsymbol{X}^j_{t-1}).$$

4 Experimental Results

In this section, we show the results of two types of experiments. In the first experiment, we measure the performance of the swarm robot to determine the probability distributions used in our DBN model. The second experiment is a simulation of the behavior of the swarm robots based on the DBN.

4.1 Measurement Experiment

This section describes the environment of the first experiment. As a swarm robot, we use a robot kit called Robo Designer, which is commercially available from Japan Robotech ltd [7]. This robot consists of two motors each of which is connected to a wheel, a battery, a controller board, and three distance-measurement sensors as shown in Fig. 2. The controller board orders to move forward or to turn right, and can read the value of the sensors.

In order to observe the behavior of the swarm robots, we use a USB camera (Logicool Qcam Orbit) of 300,000 pixels and we set it on a position of 160 cm

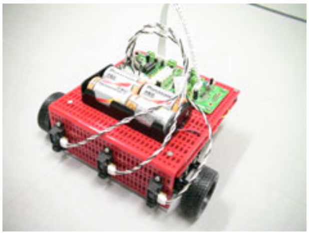


Fig. 2. ROBO DESIGNER, our swarm robot

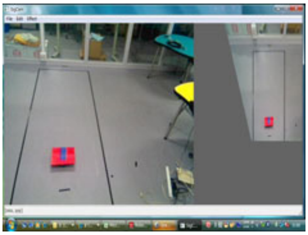


Fig. 3. Detection of the position and the direction of the robot

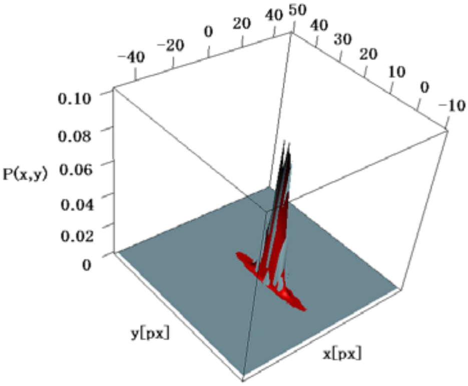


Fig. 4. Regression analysis of the position (in case of a move forward)

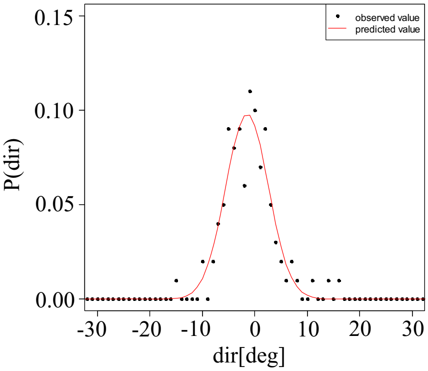


Fig. 5. Regression analysis of the direction (in case of a move forward)

height with a 45-degree angle to the ground [6]. We cover the robot with a sheet of colored paper, and detect the position and direction of the robot by using the color information. Figure 3 illustrates the overview of the detection of the position and the direction of the robot.

We observe the actions to move forward and turn right, and then calculate the change of positions and directions of the robots observed by USB camera. Here we assume that the velocity of the robot is constant.

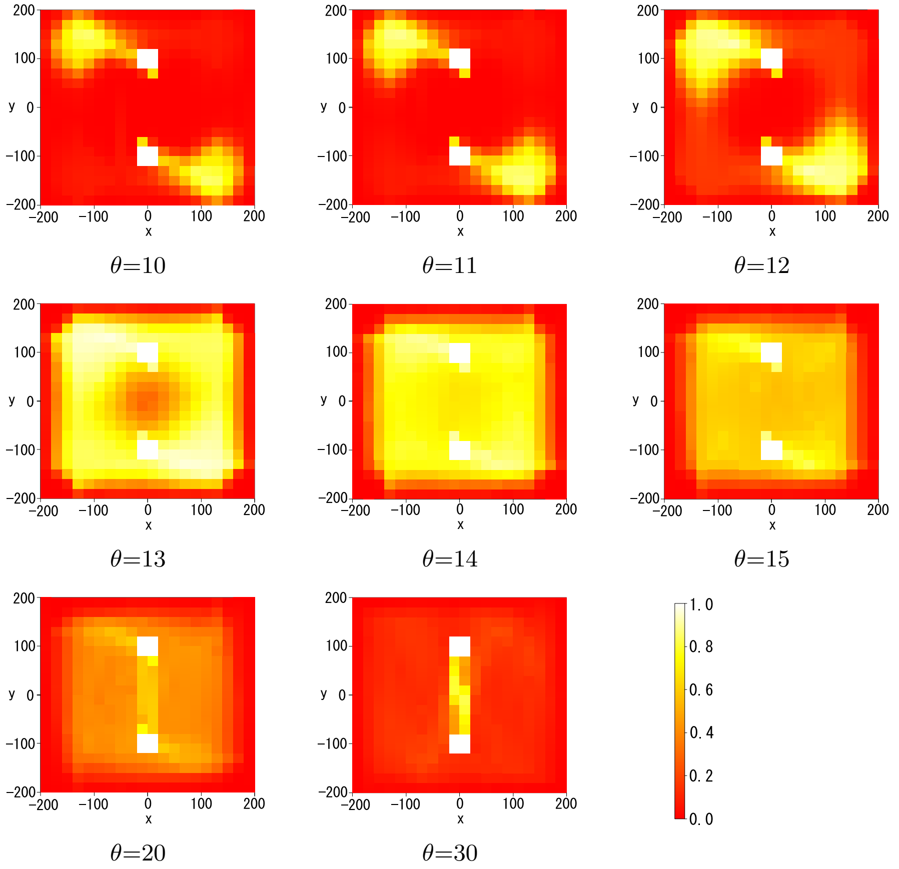
The probability distributions of the positions and directions is computed by applying a regression analysis to the observed log data. In the regression analysis, we assume that $\boldsymbol{p}_t^i$ and d_t^i follow two or one dimensional normal distributions as stated in equations (1) and (2)), respectively. The parameters $\boldsymbol{\mu_p}$, $\Sigma_{\boldsymbol{p}}$, μ_d, σ_d are estimated from the observed data.

$$f(\boldsymbol{p}) = \frac{1}{2\pi\sqrt{|\Sigma_{\boldsymbol{p}}|}} \exp\left(-\frac{1}{2}(\boldsymbol{p}-\boldsymbol{\mu_p})^T \Sigma_{\boldsymbol{p}}^{-1}(\boldsymbol{p}-\boldsymbol{\mu_p})\right) \tag{1}$$

$$f(d) = \frac{1}{\sqrt{2\pi}\sigma_d} \exp\left(-\frac{(d-\mu_d)^2}{2\sigma_d^2}\right) \tag{2}$$

Table 1. Initial parameters for our simulations

Domain D	$[-200, 200] \times [-200, 200]$
Time limit T	1000 sec.
Number of robots n	2
Initial states (x_0^1, x_0^2)	$((0, -100, 90), (0, 100, -90))$
Threshold θ	$0, 10, \ldots, 100$



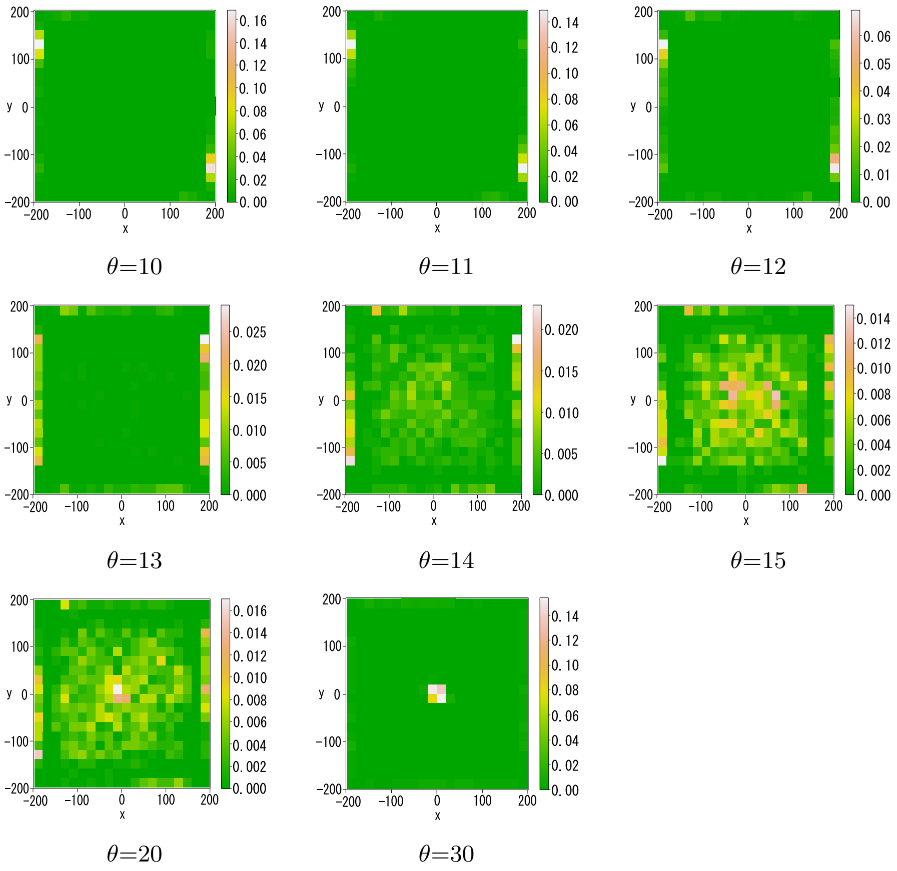
θ=10 θ=11 θ=12

θ=13 θ=14 θ=15

θ=20 θ=30

Fig. 6. Visiting probabilities of sub-domains

Results of the regression analysis using the log data of the positions and directions when the robot moves forward are $\boldsymbol{\mu_p} = (0.069, 17.24)$, $\Sigma_{\boldsymbol{p}} = \begin{pmatrix} 0.52 & 0 \\ 0 & 6.76 \end{pmatrix}$, $\mu_d = -1.47$, $\sigma_d = 3.57$ (Fig. 4 and Fig. 5). Similarly, results of regression analysis when the robot turn right are $\boldsymbol{\mu_p} = (-14.35, 21.09)$, $\Sigma_{\boldsymbol{p}} = \begin{pmatrix} 19.98 & 0 \\ 0 & 14.82 \end{pmatrix}$,



θ=10 θ=11 θ=12

θ=13 θ=14 θ=15

θ=20 θ=30

Fig. 7. Collision probabilities of sub-domains

$\mu_d = -118.49$, $\sigma_d = 11.78$. In this experiment, the interval of time sequence is 1 sec. and the initial position and direction of the robots are $(0, 0)$ and 0 degree, respectively.

4.2 Simulation

Using our DBN model, we perform simulations of the behavior of the swarm robots. For each of domain D, time limit T, number of robots n, initial states (x_0^1, x_0^2), and threshold θ, we define their values as shown in Table 1. We have obtained $\boldsymbol{P}_{\mathrm{S}earch}$ with the simulation and show it for each sub-domain with varying values for θ in Figure 6. Detailed experiments have been performed for $\theta = 10, 11, \ldots, 15$. As the result, is is clear that either 13, 14, or 15 is appropriate as the value for θ, where the robot is neither too sensitive nor too insensitive. Interestingly, there is a phase transition between $\theta = 12$ and 13, which tells us the importance of the simulation in investigating the behavior of a system composed of swarm robots.

Similarly, we have obtained $\boldsymbol{P}_{Collision}$ for each sub-domain and show them in Figure 7. The results show two phase transitions and we can conclude that $\theta = 13$ is the best value.

5 Conclusion

This paper has proposed the DBN model to simulate the behavior of the swarm robots, and calculated the probability distribution based on these model using the observed real behavior of the swarm robots. Especially, in case of the 2 robots and some thresholds for the sensor, we obtained the probability of visits and collisions, and analysed the results of our simulations. We believe that our simulation helps us to design the controller of the robot by decreasing the number of real experiments. In a future work, we will introduce a new method to discover appropriate values for the parameters.

References

1. Russel, S., Norvig, P.: Artificial Intelligence, a Modern Approach, 2nd edn. Prentice Hall, Upper Saddle River (2003)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. Doctor of Philosophy in Computer Science in the Graduate Division of the University of California, Berkeley (2002)
4. Nouyan, S., Campo, A., Dorigo, M.: Path Formation in a Robot Swarm Self-organized Strategies to Find Your Way Home. Swarm Intelligence 2(1), 1–23 (2008)
5. Toussaint, M., Goerick, C.: Probabilistic inference for structured planning in robotics. In: Proc. Int'l Conf. on Intelligent Robots and Systems (IROS), pp. 3068–3073 (2007)
6. Suzuki, E., Hirai, H., Takano, S.: Toward a Novel Design of Swarm Robots Based on the Dynamic Bayesian Network. In: Ras, Z.W., Dardzinska, A. (eds.) Advances in Data Management. Springer Studies in Computational Intelligence, Springer, Heidelberg (accepted for publication)
7. Japan Robotech, http://www.japan-robotech.com/

Hybridizing River Formation Dynamics and Ant Colony Optimization*

Pablo Rabanal and Ismael Rodríguez

Dept. Sistemas Informáticos y Computación
Facultad de Informática
Universidad Complutense de Madrid, 28040 Madrid, Spain
prabanal@fdi.ucm.es, isrodrig@sip.ucm.es

Abstract. River Formation Dynamics (RFD) is an evolutionary computation method based on copying how drops form rivers by eroding the ground and depositing sediments. In a rough sense, this method can be seen as a gradient-oriented version of Ant Colony Optimization (ACO). Several experiments have shown that the gradient orientation of RFD makes this method solve problems in a different way as ACO. In particular, RFD typically performs *deeper* searches, which in turn makes it find worse solutions than ACO in the first execution steps in general, though RFD solutions surpass ACO solutions after some more time passes. In this paper we try to get the best features of both worlds by hybridizing RFD and ACO, in particular by using a kind of *ant-drop* hybrid and considering both pheromone trails and altitudes in the environment.

Keywords: River Formation Dynamics, Ant Colony Optimization Algorithms, Heuristic Algorithms, NP-hard problems.

1 Introduction

River Formation Dynamics (RFD) [8–10, 7] is an *Evolutionary Computation* method [3, 4] related to *Ant Colony Optimization* (ACO) [2, 1]. Roughly speaking, RFD can be seen as a gradient-oriented version of ACO where, in particular, a different nature-inspired metaphor is considered. RFD is based on copying how the water forms rivers in nature. The water transforms the environment by eroding the ground when it falls through a high decreasing slope, and it deposits carried sediments when a flatter ground is reached. In this way, altitudes of places are decreased/increased, and paths of decreasing gradient are dynamically constructed. These slopes are followed by subsequent drops to create new gradients, reinforcing the best ones. Eventually, paths consisting in consecutively taking the highest decreasing gradients constitute good paths from *raining places* to the *sea*.

In RFD, drops tend to fall through high decreasing gradients with higher probability. Thus, the probabilistic decision of where a drop will move next depends

* Research partially supported by projects TIN2006-15578-C02-01, CCG08-UCM/TIC-4124, and UCM-BSCH GR58/08 - group number 910606.

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 424–431, 2011.

on the *difference* of some values (the difference of altitudes between the origin and the destination) rather than on the values themselves (in the case of ACO, the pheromone trail of the edge leading to the destination). From the beginning of the execution of RFD, any path from the origin point to the target point has a gradient that, considering this path as a whole (i.e. from the origin to the target), must be decreasing. Hence, all complete paths from the origin to the target not include climbing steps have *some* incentive for drops.[1] As a result, from the beginning of the execution the number of potential paths providing a non-negligible incentive for drops to be taken, and thus simultaneously competing for the drops preferences, is huge. This improves the deepness of the analysis but delays the formation of champion paths. In general, solutions constructed by RFD are better, though ACO obtains acceptable solutions faster.

In this paper we develop an hybrid ACO-RFD method aiming at getting the best features of both methods. In order to enable both methods, each graph node will be decorated with an altitude value, whereas each edge will be endowed with a pheromone trail value. Hybrid drop-ant entities are released in this environment, and both pheromone trails and altitudes will have some weight in deciding where these entities will move next. After moving, both the pheromone trail and the altitude of the place will be modified according to each method. The resulting hybrid method will be influenced by some values taken *as they are* (pheromone trails) and also by some *derivative* values (differences of altitudes).

Though the tendency of RFD to provide better solutions in longer times has been observed in NP-hard problems of different nature, such as the *Traveling Salesman Problem* [8] and the *Minimum Load Sequence* [7], it has been observed that RFD performs particularly well in problems consisting in creating a kind of *covering tree* over a given graph (see [10]). These are problems where the goal is constructing a tree covering some graph nodes, in such a way that a given property is met or a given value is minimized/maximized. In order to adapt RFD to these problems, we assign the lowest altitude to one of the nodes to be covered (it becomes the *sea*) and we make drops *rain* at the rest of nodes to be covered. Eventually, the gravity makes drops form a *tree* of joining tributaries from departure points to the sea, which plays the role of tree root. In this paper we will consider one of such NP-hard problems as a benchmark problem to test our hybrid method, in particular `MDV` (*Minimum Distances tree in a Variable-cost graph*, see [9, 10]; this problem will be introduced in Section 3). Let us note that, in this particular problem, RFD does not only provide better solutions than ACO, but it also takes comparable or even lower times to get these solutions. Thus, in this paper the goal of hybridizing RFD and ACO will be improving the results of RFD for a problem where RFD fits particularly well, and ACO will play the role of *supporting method* in this particular hybridization. Interestingly, the introduction of the ACO part in the RFD scheme does not reduce the times required to get reasonable solutions, as one would expect, but it improves the

[1] One of the improvements applied to the basic RFD scheme consists in allowing drops to climb *increasing* gradients with some low probability (see [8]). In this case, even paths with climbing steps would provide some incentive for drops.

quality of final solutions indeed. The reasons for this behavior will be discussed in next sections. Studying the hybridization in problems where none of both methods is dominant is left as future work. Since the hybridization improves the results in a stress benchmark case where one method dominates the other, we think that the application of the hybrid method to more balanced problems (where the hybridization should fit more naturally) is very promising.

The rest of the paper is organized as follows. In the next section we briefly present the RFD method. Then, in Section 3 we describe the MDV problem, while in Section 4 we introduce the hybrid method and we compare the results obtained by each method (RFD, ACO, and RFD-ACO hybrid) when applied to the MDV benchmark. Finally, in Section 5 we present our conclusions.

2 Brief Introduction to River Formation Dynamics

In this section we briefly introduce the basic structure of River Formation Dynamics (RFD) (for further details, see [8, 10]). Given a working graph, we associate *altitude* values to nodes. *Drops* erode the ground (they reduce the altitude of nodes) or deposit the sediment (increase it) as they move. The probability of the drop to take a given edge instead of others is proportional to the gradient of the down slope in the edge, which in turn depends on the difference of altitudes between both nodes and the distance (i.e. the *cost* of the edge). At the beginning, a flat environment is provided, that is, all nodes have the same altitude. The exception is the destination node, which is a hole (the *sea*). Drops are unleashed (i.e. it *rains*) at the origin node/s, and they spread around the flat environment until some of them fall in the destination node. This erodes adjacent nodes, which creates new down slopes, and in this way the erosion process is propagated. New drops are inserted in the origin node/s to transform paths and reinforce the erosion of promising paths. After some steps, good paths from the origin/s to the destination are found. These paths are given in the form of sequences of decreasing gradient edges from the origin to the destination. Several improvements are applied to this basic general scheme (see [8, 10]).

Compared to a related well-known evolutionary computation method, *Ant Colony Optimization*, RFD provides some advantages. On the one hand, local cycles are not created and reinforced because they would imply an *ever decreasing cycle*, which is contradictory. Though ants usually take into account their past path to avoid repeating nodes, they cannot avoid to be led by pheromone trails through some edges in such a way that a node must be repeated in the next step.[2] However, *altitudes* cannot lead drops to these situations. Moreover, since drops do not have to worry about following cycles, in general drops do not need to be endowed with *memory* of previous movements, which releases some computational memory and reduces some execution time. On the other hand, when a shorter path is found in RFD, the subsequent reinforcement of the path is fast: Since the same origin and destination are concerned in both the old and

[2] Usually, this implies either to repeat a node or to *kill* the ant. In both cases, the last movements of the ant were useless.

the new path, the difference of altitude is the same but the distance is different. Hence, the edges of the shorter path necessarily have higher down slopes and are immediately preferred (in average) by subsequent drops. Finally, the erosion process provides a method to avoid inefficient solutions because sediments tend to be cumulated in blind alleys (in our case, in *valleys*). These nodes are filled until eventually their altitude matches adjacent nodes, i.e., the valley disappears. This differs from typical methods to reduce pheromone trails in ACO: Usually, the trails of *all* edges are periodically reduced at the same rate. On the contrary, RFD intrinsically provides a *focused* punishment of bad paths where, in particular, those nodes blocking alternative paths are modified.

When there are several departing points (i.e. it rains at several points), RFD does not tend in general to provide the shortest path (i.e. river) from each point to the sea. Instead, as it happens in nature, it tends to provide a tradeoff between quickly gathering individual paths into a small number of main flows (which minimizes the total size of the formed *tree* of tributaries) and actually forming short paths from each point to the sea. For instance, meanders are caused by the former goal: We deviate from the shortest path just to collect drops from a different area, thus reducing the number of flows. On the other hand, new tributaries are caused by the latter one: By *not* joining the main flows, we can form tailored short paths from each origin point. Solving the problem considered in this paper, `MDV`, requires encouraging the latter of both behaviors. Interestingly, we can make RFD tend towards either of these choices by setting a single parameter. In particular, if we reduce the erosion caused by *high flows*, then the incentive of drops to join each other is partially reduced, and thus each drop tends to follow its own shortest path (see [9] for details). Thus, adapting RFD to `MDV` is straightforward.

3 Problem Definition

In this section we briefly describe the problem to be solved in this paper, `MDV` (*Minimum Distances tree in a Variable-cost graph*). A formal definition of the problem and a proof of its NP-completeness can be found in [10]. `MDV` can be stated as follows. Let us consider a variable-cost graph, i.e. a graph where costs are assigned to edges, and the cost of traversing a given graph edge depends on the path traversed so far. That is, if we traverse e *after* following a path σ then the cost of adding e to the path is $c_{e,\sigma}$; in general, we have $c_{e,\sigma} \neq c_{e,\sigma'}$ for any other path σ'. Besides, let us consider a subset of graph nodes required to be covered, called *origin* nodes, and a specific node called *destination* node. Then, the goal of `MDV` is constructing a tree in this graph such that the addition of the costs of paths connecting origin nodes with the destination node through the tree is as small as possible. Let us note that nodes not being origin nodes or the destination node may be included in the tree or not, depending on whether they are useful to minimize the cost of paths or not.

The previous minimization problem appears in any engineering domain where a set of origins must be joined with a destination d in such a way that distances from each origin to d are minimized (in order to improve the *quality of service*, QoS). Let us suppose that we wish to construct a local area network for a new complex of facilities in such a way that computers in all offices and buildings are connected to the company central server. A *tree topology* is chosen for the network due to the ease to further extend such a topology with new switches/routers/networks without needing to change networking devices (see the role of the tree topology in networking in e.g. [5, 6]). Improving the QoS implies minimizing distances from each node to the central node. Let us note that using variable-cost graphs allows to consider that the cost of an edge (in terms of QoS loss) depends on the *origin* of the path we have traversed before reaching the edge. For instance, the QoS of an edge denoting a *fast* but *not very reliable* wire is not high for transmitting data coming from a node typically running real-time applications (in this case, meeting some minimal speed in *all* situations is critical). However, the QoS of this edge may be high for transmitting data coming from an e-mail server (a high bandwidth is good for transmitting a huge amount of e-mails, though it is acceptable if the connection is rarely down).

MDV allows to represent the dependencies of edge costs on the path traversed so far as follows. We assume that the cost of a path of edges $e_1, \ldots, e_n$ from a given origin node o to a given destination node d depends on the evolution of a *variable* through the path. Initially, a value v_o is assigned to this variable at node o. Then, the cost added to the path due to the inclusion of edge e_1 is an amount depending on v_o. After traversing e_1, the value of the variable is updated to a new value v_1. Next, the cost of adding e_2 to the path depends on v_1. After taking e_2, the value of the variable is updated again, and the process continues so on until we obtain the whole cost of the path $e_1, \ldots, e_n$. Thus, an MDV problem instance consists of a directed graph, a set of nodes considered as *origin* nodes, the *destination* node, the *initial value* given to the variable at each origin node, the way in which the variable value *changes* its value at each edge, and the way in which the cost of each edge depends on the variable value (for a formal definition, see [10]).

4 Implementation of the Hybrid Method and Experimental Results

In this section we introduce the RFD-ACO hybrid method and we present some experimental results. In order to obtain the best characteristics of ACO and RFD, we combine them in the following manner. Instead of using ants and drops to traverse the nodes of the graph, we create a new entity called *ant-drop* that contains all the attributes of the ant as well as all the attributes needed for defining a drop. This implies that the ant-drop consumes more memory resources than an ant or a drop, though many attributes are common to both entities indeed. When the ant-drop has to decide its new destination, it takes into account the pheromone trails deposited in all outgoing edges as well as the altitude gradients of these edges. The probability p_{ij} of going from a node i to another node j through edge ij is calculated as follows: First, we compute the probability p_{ij}^{ant} an ant would have to take edge ij, as well as the probability p_{ij}^{drop} a drop would have to take such an edge. Both probabilities are computed as if the method were pure ACO or pure RFD, respectively (i.e. taking into account only pheromone trails and only altitudes, respectively). Then, p_{ij} is calculated by taking into account the relative *weight* we give to each method in our hybridization. Let w_{ACO} be the weight of the ACO method and w_{RFD} be the weight of RFD where $w_{ACO} + w_{RFD} = 1$. The probability for an ant-drop to take the edge ij is $p_{ij} = p_{ij}^{ant} \cdot w_{ACO} + p_{ij}^{drop} \cdot w_{RFD}$. After moving, pheromone values and altitudes are updated as each individual method would do.

Let us note that the relative weights of each method, i.e. w_{RFD} and w_{ACO}, do not need to be fix along the execution. On the contrary, we start the execution considering $w_{RFD} = 1$ and next this value is slowly reduced until 0. Then, the value is slowly raised again up to 1. Though we observed that considering fix values of w_{RFD} and w_{ACO} provides worse results than fluctuating like this, we also observed that different patterns of temporal fluctuation provide very similar results.

After the hybrid method is executed for some time, we start a post-processing for constructing the solution of MDV, that is, a tree. We consider any origin node and we do as follows: We select the outgoing edge with highest p_{ij}, we add this edge to the solution tree, and we traverse the edge to reach the node it leads to. We do the same for this node, and so on until the destination is reached. Next we take another origin node and we do the same until we reach a node already included in the tree (and we join the branch to the current tree, thus introducing a tributary) or the destination is reached. We do the same for all origin nodes and, finally, a tree leading all origin nodes to the destination will be given.

Next we report some experimental results. We compare the results obtained by using an ACO method, the solutions given by RFD, and the solutions found by the hybrid method (from now on, HYB). All experiments were performed in an Intel Core Duo T7250 processor with 2.00 GHz. Problem instances are graphs with 100, 150, and 200 nodes. These graphs were randomly generated in such a way that the cost of the edges, which depends on the actual value of the variable, is always between 1 and 10. Variables can take up to 5 possible values. For each edge and each variable value, the cost of traversing the edge for this value, as well as a new variable value after traversing the edge, were randomly generated. In particular, features such as monotonicity or injectivity were not required in these correspondences.

In order to report solutions that are not biased by a single execution, each algorithm was executed thirty times for each graph. Table 1 summarizes the best solution found by each method among the 30 executions, as well as the arithmetic mean. As we can see, for all graphs the hybrid method obtains the best solutions, considering both mean results and best results. Regarding the time needed to obtain good solutions in specific executions, Figure 1 and Figure 2 show the cost of the best solution found by each algorithm for each execution time (in seconds), where the input of the three algorithms was the graph with 100 and 200 nodes, respectively. As we can see, the hybrid method obtains the best solution after

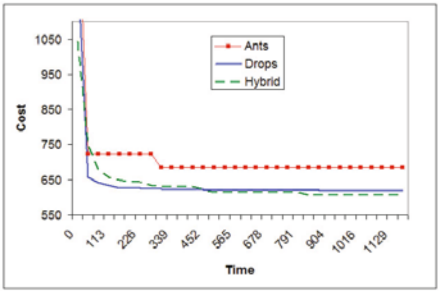


Fig. 1. `MDV` results for a 100 nodes graph

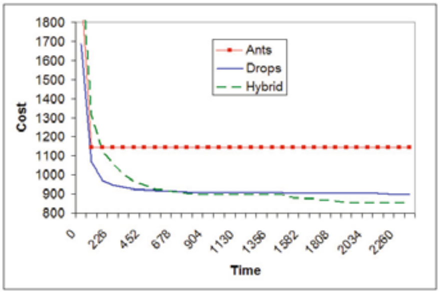


Fig. 2. `MDV` results for a 200 nodes graph

Table 1. Summary of results

Method	Graph size	Best result	Arithmetic mean
RFD	100 nodes	593.75	615.66
ACO	100 nodes	589.11	614.25
HYB	100 nodes	563.59	585.15
RFD	150 nodes	666.33	684.59
ACO	150 nodes	702.68	738.03
HYB	150 nodes	636.88	655.98
RFD	200 nodes	858.83	873.12
ACO	200 nodes	993.81	1082
HYB	200 nodes	848.56	870.98

some time passes. However, it is a little bit slower than RFD because of the extra costs needed to handle the information of both ACO and RFD.

The improvement of the solutions quality in the hybrid method is due to the way in which this method analyzes the search space. By combining both methods, the hybrid method avoids getting stuck in solutions that are a local optimum for one method but not for the other one. The alternation in the weight of each method along time allows each method to be dominant for some time. Though both methods are based on driving entities in a different way, the underlying evolutionary computation model is related. Hence, the mechanics of both methods are partially compatible, which in turn allows both methods to partially collaborate in the formation of solutions.

5 Conclusions and Future Work

In this paper we have presented an hybrid evolutionary computation approach where two previous methods, RFD and ACO, are mixed. In brief, the resulting method consists in driving entities in terms of both some absolute values (pheromone trails) and some derivative values (the differences of altitudes). As a benchmark problem, we have considered MDV, an NP-complete problem where RFD provides better solutions than ACO and takes equal or smaller times to provide them. Interestingly, despite of considering a problem where one of the methods dominates the other, the hybrid method makes an improvement with respect to *both* individual methods. In particular, the inclusion of the ACO approach into the RFD scheme allows the resulting hybrid method to improve the *quality* of solutions. This is particularly remarkable, because previous experiments showed that, in other more balanced problems such as the Traveling Salesman Problem (TSP), the strong point of ACO with respect to RFD was *not* the quality of final solutions (which was lower), but the smaller time needed to construct reasonable solutions.

The mixture of RFD and ACO has proved its usefulness for solving MDV, which is a problem where a direct combination of the features of both methods would not be expected to be an improvement (ACO neither provides better solutions nor is faster than RFD for MDV). Thus, we think that the hybrid approach is particularly promising to solve other problems where a combination of features of both methods would be an improvement indeed (such as TSP). Hence, our next step will be applying the hybrid method to TSP.

References

1. Dorigo, M., Gambardella, L.M.: Ant colonies for the traveling salesman problem. BioSystems 43(2), 73–81 (1997)
2. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics, Part B 26(1), 29–41 (1996)
3. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
4. De Jong, K.A.: Evolutionary computation: a unified approach. MIT Press, Cambridge (2006)
5. Perlman, R.: An algorithm for distributed computation of a spanningtree in an extended lan. In: Symposium on Data communications, SIGCOMM 1985, pp. 44–53. ACM, New York (1985)
6. Peterson, L.L., Davie, B.S.: Computer Networks: A Systems Approach, 3rd edn. Morgan Kaufmann, San Francisco (2007)
7. Rabanal, P., Rodríguez, I.: Testing restorable systems by using RFD. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009. LNCS, vol. 5517, pp. 351–358. Springer, Heidelberg (2009)
8. Rabanal, P., Rodríguez, I., Rubio, F.: Using river formation dynamics to design heuristic algorithms. In: Akl, S.G., Calude, C.S., Dinneen, M.J., Rozenberg, G., Wareham, H.T. (eds.) UC 2007. LNCS, vol. 4618, pp. 163–177. Springer, Heidelberg (2007)
9. Rabanal, P., Rodríguez, I., Rubio, F.: Finding minimum spanning/distances trees by using river formation dynamics. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217, pp. 60–71. Springer, Heidelberg (2008)
10. Rabanal, P., Rodríguez, I., Rubio, F.: Applying river formation dynamics to solve NP-complete problems. In: Chiong, R. (ed.) Nature-Inspired Algorithms for Optimisation. SCI, vol. 193, pp. 333–368. Springer, Heidelberg (2009)

Landmark Navigation Using Sector-Based Image Matching

Jiwon Lee and DaeEun Kim

Biological Cybernetics Lab,
School of Electrical and Electronic Engineering,
Yonsei University,
Seoul 120-749, Corea (South Korea)
{jiwonlee,daeeun}@yonsei.ac.kr

Abstract. Many insects return home by using their environmental landmarks. They remember the image at their nest and find the homeward direction, comparing it with the current image. There have been robotic researches to model the landmark navigation, focusing on how the image matching process can lead an agent to return to the nest, starting from an arbitrary spot. According to Franz's navigation algorithm, an agent estimates the changes of image for its own movement, and evaluates which directional movement can produce the image pattern most similar to the snapshot taken at the nest. Then it finally chooses the best image-matching direction. Based on the idea, we suggest a new navigation approach where the image is divided into several sectors and then the sector-based image matching is applied. It checks the occupancy and the distance variation for each sector. As a result, it shows better performance than Franz's algorithm.

Keywords: landmark navigation, bio-inspired navigation, biorobotics, image matching.

1 Introduction

Many insects and animals can return home accurately by using their own navigation system, after exploring to find the food. It is well known that desert ants, fiddler crabs, and honeybees use the path integration algorithm, also called dead reckoning [1, 2]. For path integration, the current moving speed and the angular information with the reference, for example, light compass are measured and integrated all through traveling, and the conclusive direction is selected for returning home. Without any help of a specific external landmark, the animals and insects can implement the path integration for their navigation.

However, the insects use both landmark navigation and path integration because it is difficult to arrive home accurately by using only the path integration algorithm [3]. For landmark navigation, the visual environment near the nest is used for homing as the information of the goal position.

Various models to explain how the insects navigate with environmental landmarks have been suggested [4–7]. According to the snapshot model [7], an agent

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 432–439, 2011.

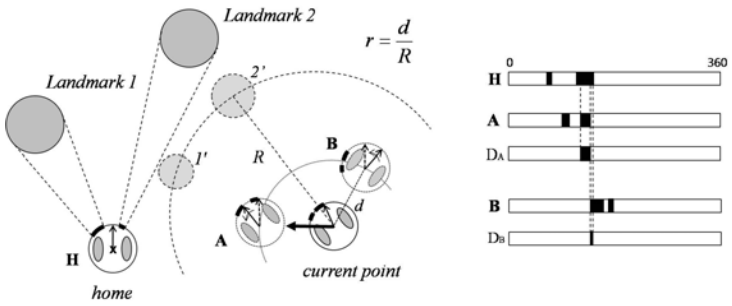


Fig. 1. Estimation process in the image matching method. D_A: dot product result of $H \cdot \text{image}(A)$ where H is the home snapshot and A is one estimated position, D_B: dot product result of $H \cdot \text{image}(B)$ where B is another estimated position; R is the landmark distance and d is the moving distance of an agent.

remembers the snapshot image at its nest and compares it with new images at the current location. Then it chooses the best matching direction with the snapshot. Afterwards, many different navigation models have been introduced. Average landmark vector model (ALV) draws a vector for each landmark, and the agent recognizes the visual environment as a summation of the local vectors [8]. The agent goes towards the direction of the difference between the two vectors, one landmark vector at the current position and the other landmark vector at the nest. The algorithm easily finds appropriate direction for any position, but if one small landmark is missing by noise, the navigation direction is significantly influenced.

The image matching algorithm suggested by Franz et al. [9] includes the estimation process of new predictive images at possible movements. An agent considers all possible directions from the current position and each direction implies a new predictive image. This image is compared with the snapshot image at the nest. The best matching direction is the direction that the agent chooses. For this approach, it is assumed all landmarks are in the equal distance far away from the agent in order to generate the predictive image for each directional movement. Fig. 1 shows an example of this image matching method. If there is no reference compass, all possible head directions of the agent should be considered to estimate the best image matching direction. In spite of the equidistance assumption, it was shown that the approach can guide the agent to the nest when the nest is surrounded by landmarks. However, an agent may misjudge the homing direction, depending on the moving distance and estimation of the landmark distance. Also if large landmarks are available, the matching process tends to choose the direction biased by large landmarks.

In this paper, we suggest a new navigation algorithm, called sector-based approach. The omnidirectional snapshot image is divided into several sectors. Predictive images are generated at the current position as Franz showed, and the predictive images and the snapshot image are compared sector by sector. Each sector evaluates the occupancy state of landmarks and the landmark distance

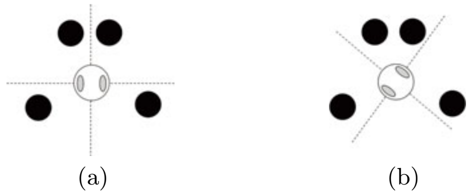


Fig. 2. Sector division; the visual environment surrounded by four landmarks is divided into four non-overlapping sectors (a) honeybee heads for the front (b) The orientation of the bee changes the sector occupancy (modified from [5])

for both the home snapshot and the predictive image. Then the parameters are compared each other and the agent chooses the direction with the best matching score. We will first introduce our sector-based method in details and demonstrate the simulation experiments in the robotic environment.

2 Method

In this paper, we propose a new navigation algorithm with sector-based image matching. Anderson [5] used sector-based analysis to unravel the honeybee's navigation. The visual information is divided to separate parts, called sectors. To compare the two images, or estimate how close they are, the two images are divided into sectors, respectively. Then each sector in a image is compared with a sector in the other image, one by one. Here, occupancy and distance are the factors evaluating the matching degree between a pair of sector images. The occupancy checks if at least one landmark is found and the distance factor evaluates the distance difference between objects, if a sector is occupied. The distance can be estimated by the bearing angles of objects. By comparing sector by sector for the two images, an agent can determine the matching level of the two images.

The sector-based analysis calculates the matching score over a pair of images, and Anderson [5] applied it to see the distribution of the matching score around the honeybee's home. However, how an agent chooses the movement direction has not been handled. In this paper, we use the sector-based image matching process to select the next movement based on the image snapshot. If predictive images are generated for possible directional movements from the current position with equidistance assumption, each predictive image is compared with the snapshot image at the nest, using the sector-based image matching. Here, an agent plans to head towards a place that is similarly surrounded by the landmarks.

In Fig. 2(a), the visual environment is divided into four sectors. Four landmarks surround the robot and when the robot heads towards the front, the robot recognizes that all sectors are occupied, since each sector includes one landmark image.

When the robot turns its head direction to the left as shown in Fig. 2(b), two front landmarks belong to the same sector, and thus the occupancy changes

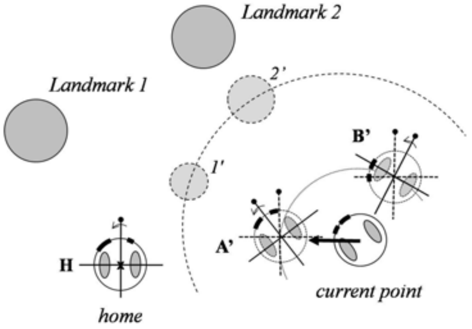


Fig. 3. Estimation process in the sector-based matching method

from 1 1 1 1 to 1 0 1 1 (sectors are counted from the head direction). Also, we can represent the distance for each occupancy as given below:

	(a)	(b)
Occupancy	1 1 1 1	1 0 1 1
Distance	a b b a	b 0 b a

The matching score between the two images is calculated as follows:

$$\text{Matching score } \boldsymbol{m_i} = \text{ Occupancy matching} \times (1 - \text{Distance difference}), \qquad (1)$$

where there are N viewpoints (head directions). The degree of distance difference is normalized into the range [0, 1]. The matching score m_i, for $i = 1, ..., N$ for each head direction is calculated, and the maximum score becomes the representative value at a specific point. It is called a local maximum f_j (Equation (2)).

$$f_j = \max_i m_i, \quad j = 1, \ldots, M, \qquad (2)$$

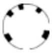
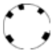
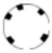
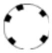
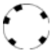
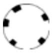
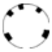
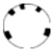
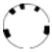
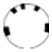
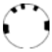
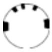
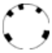
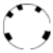
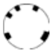
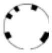
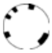
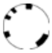
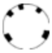
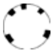
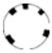
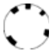
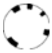
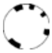
where M is the number of possible moving directions to build predictive images. Then we finally estimate the best matching direction, the j^*-th angular direction, which equals the homing direction for the agent to choose.

$$j^* = \arg\max_j f_j \qquad (3)$$

Using Franz's predictive model with equidistance assumption, the robot tries to estimate the matching score for possible directional movements. The matching score f_j is evaluated for each direction. The robot chooses the best matching direction by finding the global maximum value. Fig. 3 illustrates this estimation process, where predictive images are generated for each directional movement, and images are obtained with the assumption that all landmarks are in the same distance far away from the current position, and each moving distance for all directions is fixed.

Table 1 shows how the images are predicted at the current position. The robot can move to all directions from the current position $P(500, 450)$, and the robot

Table 1. Predictive images for possible movements (four directions, north, south, east, west); robot's current position at P(500,450) when home is at H(500,500)

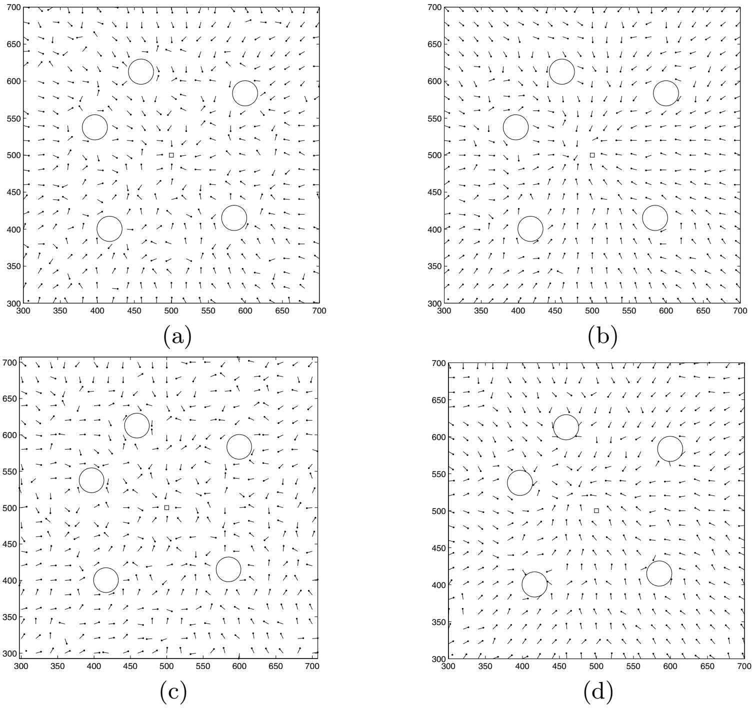
Direction	P image	Ratio 0.1	0.2	0.3	0.4	0.5
North						
South						
East						
West						

predicts possible images distorted from the snapshot image (P image in Table 1) at the current position. These images are compared with the snapshot at home. If the robot moves further to the south, the landmark images in the northern area becomes closer. The ratio $\rho = d/R$ is obtained from the moving distance d and the landmark distance R; we assume $R = 100cm$ in this paper. Thus, the larger ratio means the larger movement in the preferred direction.

The sector-based approach follows the basic scheme of Franz's image matching process, and needs to consider all possible head directions to find the best matching direction. If the reference compass is available, the algorithm becomes simpler. Here, the sector-based approach estimates the matching score, sector by sector, rather than pixel by pixel comparison. It can show more robust navigation for noisy environments.

3 Experiments

We tested our sector-based image matching and Franz's pixel-based image matching. We assumed that five cylindrical landmarks are in the environment, and the home position is at (500,500), as shown in Fig. 4. A robot is supposed to return home after exploration. According to our sector-based approach, the robot predicts images for all possible directions and then compares the predictive images and the snapshot at the nest. Sector-based comparison between the two images determines the best matching direction. Here, we assumed that there are 72 directions available (5 degree resolution) and four non-overlapping sectors are investigated. By equidistance assumption, the robot thinks all landmarks are in the same distance of $R = 100cm$. Fig. 4(a)-(b) shows the vector map where the arrow represents the best matching direction that the robot chooses. When the robot is placed at an arbitrary position, it ultimately returns home by applying the sector-based matching repeatedly. The ratio $\rho = d/R$ also influences the performance. The high ratio $\rho = 0.5$ triggers more consistent directional movement



(a)

(b)

(c)

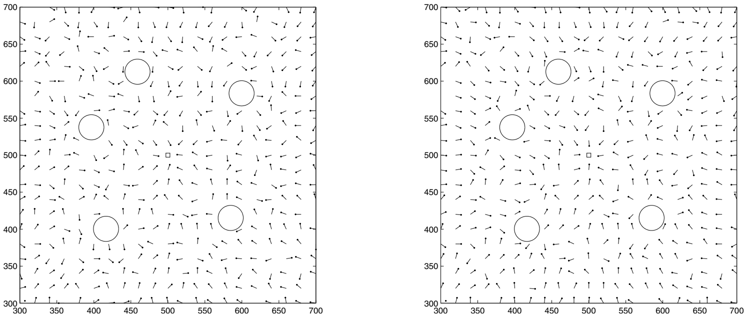
(d)

Fig. 4. Vector map; 72 moving directions and 72 head directions were tested and (500,500) is the home position (a) $\rho = 0.1$ with sector-based approach (b) $\rho = 0.5$ with sector-based approach (c) $\rho = 0.1$ with Franz's method (d) $\rho = 0.5$ with Franz's method

to the nest, as shown in Fig. 4(b). With Franz's method, there is a catchment area where the robot can enter the nest. Especially in the boundary area between objects, it shows low performance.

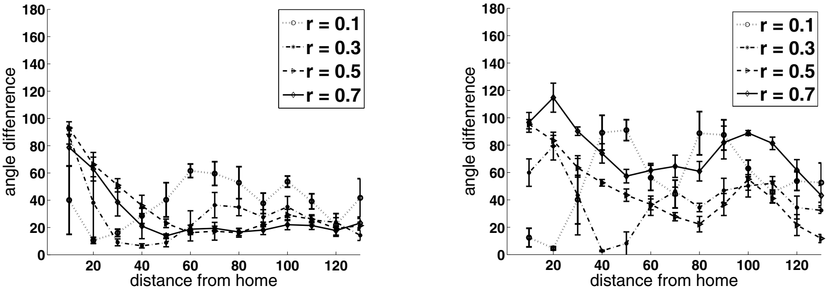
In our experiments, no reference compass is given. The robot should decide the next directional movement purely depending on image information. Thus, the robot tries to match the images with all possible head directions. Here, we tested varying number of head directions. The new head direction implies a new viewpoint over the landmarks as shown in Fig. 2. The occupancy and distance difference for each sector change depending on the head direction. Fig. 5 shows that more viewpoints lead the robot to follow the homeward direction easily.

For more detailed comparison between Franz's and the sector-based matching method, we calculated the angular errors between the estimated direction and the desired direction. Here, the desired direction is the direction in the direct route from a given position to the nest position (500,500). The best matching direction has been calculated for a number of samples within a given distance from the nest, and the direction is compared with the desired direction. Fig. 6(a) shows that the error tends to decrease at far distances, and the distance ratio influences the performance. The sector-based approach shows much better



(a) (b)

Fig. 5. Vector map by sector-based algorithm ($R = 100$, $\rho = 0.1$, 72 moving directions); (a) 6 head directions tested (b) 36 head directions tested



(a) (b)

Fig. 6. Comparison between sector-based approach and Franz's approach; the performance is measured with the angular difference between estimated direction and the desired direction (a) varying ratios and distances with the sector-based approach (b) varying ratios and distances with the Franz's approach

performance than Franz's approach. In Fig. 6(b), there is a large fluctuation of performance depending on the robot position, and the distance ratio also significantly influences the performance, since the ratio specifies the next position to be checked. High ratio in an area around the nest yields a wrong decision of moving direction. Pixel-based matching may not be effective unless we know prior information of the distance from the nest. In contrast, the sector-based approach shows consistently good result for large distances, while the performance degrades in the area close to the nest. In spite of the equidistance assumption, the sector-based approach can guide the robot to homeward direction under a large range of distance ratios. It is more efficient and robust than Franz's approach.

4 Conclusion

In this paper, we develop a new image matching approach, called sector-based landmark navigation. We follow the predictive image matching suggested by

Franz et al. [9], where the snapshot image at the current position is converted into a collection of predictive images depending on the directions which the robot moves in. All the predictive images are compared with the snapshot image at the nest, sector by sector. This sector-based image matching significantly improves the performance of trajectories to return home from an arbitrary spot. We showed the sector-based image matching is significantly better than Franz's pixel-based matching method. In addition, the sector-based approach allows to acquire a desired performance level under a low resolution of images.

Here, we handled only four sectors for the image matching process. We can consider varying number of sectors, which can affect the efficiency of the proposed algorithm. Furthermore, how the sectors are divided may be another factor to influence the performance. For example, if there is an overlap between neighbor sectors, it can possibly improve the navigation performance for noisy environments. We leave those studies for the future work.

If the reference compass is available, the sector-based algorithm becomes more efficient, since we do not need to check all the head directions to match images. It will substantially reduce the computation time of the algorithm. For the future work, we can compare the sector-based approach and Franz's approach when the reference compass is given.

Acknowledgments. This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (No.2009-0080661).

References

1. Collett, M., Collett, T.: How do insects use integration for their navigation? Biological Cybernetics 83, 245–259 (2000)
2. Zeil, J., Hemmi, J.: The visual ecology of fiddler crabs. Journal of Comparative Physiology 192, 1–25 (2006)
3. Collett, M., Colett, T.S., Bisch, S., Wehner, R.: Local and global vectors in desert ant navigation. Nature 394, 269–272 (1998)
4. Collett, T.S., Land, M.F.: Visual spatial memory in a hoverfly. Jourcal of Comparatice Physiology 100, 59–84 (1975)
5. Anderson, A.M.: A model for landmark learning in the honeybee. Jourcal of Comparatice Physiology. A 114, 335–355 (1977)
6. Wehner, R., Ber, F.R.: Visual spatial memory in desert ants, cataglyphis bicolor (hymenoptera: Formicidae). Experientia 35, 156–157 (1979)
7. Cartwright, B., Collett, T.: Landmark learning in bees. Journal of Comparative Physiology. A 151, 521–543 (1983)
8. Moller, R.: Insect visual homing strategies in a robot with analog processing. Biological Cybernetics 83, 231–243 (2000)
9. Franz, M.O., Scholkopf, B., Mallot, H.A., Bulthoff, H.H.: where did i take that snapshot? scene-based homing by image matching. Biological Cybernetics 79, 191–202 (1998)

A New Approach for Auto-organizing a Groups of Artificial Ants

Hanane Azzag and Mustapha Lebbah

LIPN-UMR 7030
Université Paris 13 - CNRS
99, av. J-B Clément - F-93430 Villetaneuse
{Hanane.Azzag,Mustapha Lebbah}@lipn.univ-paris13.fr

Abstract. We present in this paper a new combined clustering algorithm based on two biomimetic models : artificial ants and self-organizing map (SOM). We describe the main principles of our method that aims at auto-organizing a group of homogeneous ants (data's). We show how these principles can be applied to the problem of data clustering.

1 Introduction

The initial goal of this work is to build a new clustering algorithm and to apply it to several domains. Data clustering is identified as one of the major problems in data mining. Popularity and different variations linked to the clustering problem have given birth to a several methods of resolution [1]. These methods can both use heuristic or mathematics principles. We propose in this paper a new approach which builds a tree-structured clustering of the data. This method simulates a new biological model: the way ants build structures by assembling their bodies together. Ants start from a point (support) and progressively become connected to this point, and recursively to these firstly connected ants. Ants move in a distributed way over the living structure in order to find the best place where to connect. This behavior can be adapted to build a tree from the data to be clustered. To simulate the started point (support), we provide an original initial organization of the data (ants) by using another biological model which has inspired researchers for more than several years. Self-organizing map (SOM) is biomimetic method introduced by Kohonen and inspired from human neural network [2]. The SOM method is an unsupervised method which auto-organizes data sets in a 2D grid (map) and which uses a local neighborhood function to preserve the topological properties of the real model.

The numerous abilities of ants have inspired researchers for more than ten years regarding designing new clustering algorithms [3][4]. The initial and pioneering work in this area is due to Deneubourg and his colleagues [3]. The authors have been interested in the way real ants sort objects in their nest by carrying them and dropping them in appropriate locations without a centralized control policy. The next step toward data clustering has been done in [4]. These authors have adapted the previous algorithm by considering that an object is a datum

G. Kampis, I. Karsai, and E. Szathmáry (Eds.): ECAL 2009, Part II, LNCS 5778, pp. 440–447, 2011.

and by tuning the picking/dropping probabilities according to the similarities between data. These ants based algorithms inherit from real ants interesting properties, such as the local/global optimization of the clustering, the absence of need of a priori information on an initial clustering or number of classes, or parallelism. Furthermore, the results are presented as a visualization, a property which is coherent with an important actual trend in data mining called "visual data mining" where results are presented in a visual and interactive way to the domain expert.

2 From Real to Artificial Ants : 'AntTree Algorithm'

2.1 Biological Inspiration

The self-assembly behavior of individuals can be observed in several insects like bees or ants for instance (see a survey with spectacular photographs in [5]). We are interested here in the complex structures which are build by ants. We have specifically studied how two species were building such structures, namely the Argentina ants *Linepithema humiles* and ants of gender *Oecophylla* and that we briefly describe here [6][7]: these insects may become fixed to one another to build live structures with different functions. Ants may thus build "chains of ants" in order to fill a gap between two points. These structures disaggregate after a given time.

2.2 Artificial Ants

For our classic algorithm called AntTree [8] each ant $a_i, i \in [1, N]$ represents one data d_i. An ant a_i is moving over the support denoted by a_0 or over an other ant denoted by a_{pos} (see the ants colored in gray on figure 1). The similarity measure used will be defined in section 3.1. For instance we call it $\mathrm{Sim}(i, j)$. The main principles of our deterministic algorithm are the followings: at each step, an ant a_i is selected in a sorted list of ants (we will explain how this list is sorted in the following section) and will connect itself or move according to the similarity with its neighbors. While there is still a moving ant a_i, we simulate an action for a_i according to its position (i.e. on the support or on another ant). In the following we consider that a_{pos} denotes the ant or the support over which the moving ant a_i is located, and a^+ is the ant (daughter) connected to a_{pos} which is the most similar to a_i (see figure 1).

For clarity, consider now that ant a_i is located on an ant a_{pos} and that a_i is similar to a_{pos}. As will be seen in the following, when an ant moves toward another one, it means that it is similar enough to that ant. So a_i will become connected to a_{pos} provided that it is dissimilar enough to ants connected to a_{pos}. a_i will thus form a new sub-category of a_{pos} which is as dissimilar as possible from the other existing sub-categories. For this purpose, let us denote by $T_{Dissim}(a_{pos})$ the lowest similarity value which can be observed among the daughters of a_{pos}. a_i is connected to a_{pos} if and only if the connection of a_i decreases further this value. The test that we perform consists in comparing a_i to the most similar

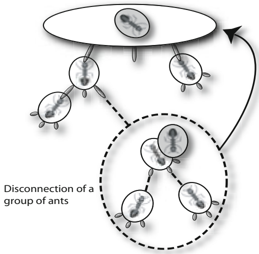


Fig. 1. Disconnection of ants

ant a^+. If these two ants are dissimilar enough ($Sim(a_i, a^+) < T_{Dissim}(a_{pos})$), then a_i is connected to a_{pos}, else it is moved toward a^+. Since this minimum value $T_{Dissim}(a_{pos})$ can only be computed with at least two ants, then the two first ants are automatically connected without any tests. This may result in "abusive" connections for the second ant. Therefore the second ant is removed and disconnected as soon as a third ant is connected (for this latter ant, we are certain that the dissimilarity test has been successful). When this second ant is removed, all ants that were connected to it are also dropped, and all these ants are placed back onto the support (see figure 1). This algorithm can thus be stated, for a given ant a_i, as the following ordered list of behaviour rules:

R_1 (no ant or only one ant connected to a_{pos}): a_i connects to a_{pos}

R'_1 (2 ants connected to a_{pos}, and for the first time):

1. Disconnect the second ant from a_{pos} (and recursively all ants connected to it)
2. Place all these ants back onto the support a_0
3. Connect a_i to a_{pos}

R_2 (more than 2 ants connected to a_{pos}, or 2 ants connected to a_{pos} but for the second time):

1. let $T_{Dissim}(a_{pos})$ be the lowest dissimilarity value between daughters of a_{pos} (i.e. $T_{Dissim}(a_{pos}) = Min\ Sim(a_j, a_k)$ where a_j and $a_k \in$ {ants connected to a_{pos}}),
2. If a_i is dissimilar enough to a^+ ($Sim(a_i, a^+) < T_{Dissim}(a_{pos})$) Then a_i connects to a_{pos}
3. Else a_i moves toward a^+

When ants are placed back on the support, they may find another place where to connect using the same behavior rules. It can be observe that, for any node of the tree, the value $T_{Dissim}(a_{pos})$ is only decreasing, which ensures the termination and convergence of the algorithm. One must notice that no parameters or predefined thresholds are necessary for using our algorithm: this is one major advantage, because ants based methods often need parameter tuning, as well as clustering methods which often require a user-defined similarity threshold.

3 Artificial Ants and Self-Organizing Map

To avoid data sorting and to provide an initial organization of the ants, we propose in this work to hybridize the artificial ant algorithm with a clustering method which uses the self organizing map. This clustering approach named AntTree-N-W (N: neighborhood, W: ward distance) has the advantage to present a topological organization of data in 2D map. In addition it permit us to cluster a large number of data. Self-organizing maps provide an initial organization of ant on the grid, where each group of ants are represented by an ant "prototype". Until this point, hybrid model seems to be standard, but in our case we define a measure of similarity between ant prototype taking into account their proximity on the grid "map". Below we present a global definition of SOM map.

Self-organizing maps are increasingly used as tools for visualization, as they allow projection over small areas that are generally two dimensional.

The basic model proposed by Kohonen consists on a discrete set $\mathcal{C}$ of cells called map. This map has a discrete topology defined by undirected graph, usually it is a regular grid in 2 dimensions. We denote p the number of cells. For each pair of cells (c,r) on the map, the distance $\delta(c, r)$ is defined as the length of the shortest chain linking cells r and c on the grid. For each cell c this distance defines a neighbor cell; in order to control the neighborhood area, we introduce a kernel positive function $\mathcal{K}$ ($\mathcal{K} \geq 0$ and $\lim\limits_{|x|\to\infty} \mathcal{K}(x) = 0$). We define the mutual influence of two cells c and r by $\mathcal{K}(\delta(c, r))$. In practice, as for traditional topological map we use smooth function to control the size of the neighborhood as $\mathcal{K}(\delta(c, r)) = \exp(\frac{-\delta(c,r)}{T})$. Using this kernel function, T becomes a parameter of the model. As in the Kohonen algorithm, we decrease T from an initilal value T_{max} to a final value T_{min}. Let $\mathcal{R}^d$ be the euclidean data space and $\mathcal{A} = \{\mathbf{d}_i; i = 1, \ldots, N\}$ a set of observations, where each data (ant) $\mathbf{d}_i = (d_i^1, d_i^2, ..., d_i^n)$ is a continuous vector in $\mathcal{R}^n$. For each cell c of the grid, we associate a referent vector $\mathbf{w}_c = (w_c^1, w_c^2, ..., w_c^j, ..., w_c^d)$ of dimension n (referent ant). We denote by $\mathcal{W}$ the set of the referent vectors. The set of parameter $\mathcal{W}$, has to be estimated from $\mathcal{A}$ iteratively by minimizing a cost function defined as follows :

$$\mathcal{J}(\phi, \mathcal{W}) = \sum_{\mathbf{d}_i \in \mathcal{A}} \sum_{r \in C} \mathcal{K}(\delta(\phi(\mathbf{d}_i), r))||\mathbf{d}_i - \mathbf{w}_r||^2 \tag{1}$$

where ϕ assign each observation $\mathbf{z}$ to a single cell in the map $\mathcal{C}$. In this expression $||\mathbf{z} - \mathbf{w}_r||^2$ is square of the Euclidean distance.

The cost function (1), is minimized using an iterative process with two steps.

1. Assignment step which leads to the use of the following assignment function:

$$\forall \mathbf{d},\ \phi(\mathbf{x}) = \arg\min_c \left(||\mathbf{d} - \mathbf{w}_c||^2\right)$$

2. Optimization step the referent ant vector $\mathbf{w}_c$ as the mean vector as:

$$\mathbf{w}_c = \frac{\sum_{\mathbf{d}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{d}_i)))\mathbf{d}_i}{\sum_{\mathbf{d}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{d}_i)))},$$

3.1 A New Similarity Measure

At the end of learning, SOM provide a partition of p groups. This partition will be denoted by $\mathcal{P} = \{P_1, \ldots, P_c, \ldots, P_p\}$. Each subset P_c is associated to a referent ant $\mathbf{w}_c \in \mathcal{R}^n$. This partition is used as input of AntTree-N-W algorithm. In order to take into account the ant topology provided by self-Organizing Map, we define a new measure between ants group. Connecting two "referent"ant , infer merging the associated group of ants. The most widely considered rule for merging groups in continuous space may be the method proposed by Ward [9] which is defined as follows:

$$Dist_W = \left(\frac{n_c n_r}{n_c + n_{c_r}}\right) ||\mathbf{w}_c - \mathbf{w}_r||^2 \tag{2}$$

where n_c and n_r represent respectively the size of groups P_c and P_r.

It is necessary to weight the Ward measure by the loss of inertia with value that measures the topological modification after merging two subsets. We propose to quantify this topological modification by as follows:

$$\sum_{u \in C} K(\delta((c,r),u))) \text{ where } \delta((c,r),u) = \min\{\delta(c,u), \delta(r,u)\}$$

This quantity allows to quantify the topological modification, but do not allow to take into account the referent proximity on the map. Thus we propose to subtract value that measures this proximity as follows:

$$\begin{aligned} Dist_{N-W} = &\left(\sum_{u \in C} K(\delta((c,r),u)))\right) \frac{n_c n_r}{n_c + n_{c_r}} ||\mathbf{w}_c - \mathbf{w}_r||^2 \\ &- K(\delta(c,r))(n_c + n_r)||\mathbf{w}_c - \mathbf{w}_r||^2 \end{aligned} \tag{3}$$

This measure is composed with two terms. The first term computes the inertia loose after connecting P_c and P_r. The second term brings subsets corresponding to two referent neighbors on the map, in order to maintain topological order between subsets. Small proximity between two neighbor c and r infers small $\delta(c, r)$, thus the neighbor function $K(\delta(c, r))$ becomes high. Hence, the second term reduces the first term depending on the neighbor function. It's obvious that for null neighborhood our measure computes only Ward criterion. The criterion we proposed allows to obtain a regularized Ward criterion, and this regularization is obtained with the topological order used. Finally the new measure defines a dissimilarity matrix which take into account the inertia loss and the topological order. AntTree has the advantage to have a low complexity (near the $n\ log(n)$) [8]. Those times will be further reduced since AntTree will be applied on the referent ant provided by the SOM algorithm. Thus, the tree structured obtained is as the best representative of referent ant set $\mathcal{W} = \{\mathbf{w}_1, ..., \mathbf{w}_p\}$ (with the new similarity measure ([3])). Each node parent of the tree is more representative of their node son. So the algorithm, AntTree-N-W can be described as follows:

- **Input:** $\mathcal{W} = \{\mathbf{w}_1, ..., \mathbf{w}_p\}$, set of referents ant which are positioned on the map using SOM algorithm,
  - Compute the new distance defined in (expression 3),
  - Building the tree using AntTree algorithm,
- **Output:** Tree structure of referents

The tree obtained provides a clustering $\mathcal{P} = \{P_1 ..., P_s\}$ where the value s represent the number of clusters founded (each sub-tree represent one cluster) provided by AntTree-N-W using map position of referent ants

4 Validation

We have evaluated and compared our algorithms on a set of 19 databases (table 1). Art1 to Art6 are artificial and have been generated with Gaussian and uniform distributions. The others are real and extracted from the machine learning repository [10,11]. To evaluate the quality of clustering using map position, we adopt the approach of comparing the results to a "ground truth". We use the clustering accuracy for measuring the clustering results. The index is classification rate, usually named purity measure which can be expressed as the percentage of elements of the assigned class in a cluster.

We compare our map clustering process AntTree-N-W with classic version of AntTree. We also compare the impact of using the new measure of similarity. We thus compare AntTree-N-W with AntTree-W using classic Ward index (expression 2). Concerning the number of cluster, we observe a clear improvement when we use AntTree-N-W even if there are cases where the number of found cluster is real bad.

Table 1. Databases used in the experimentation

Datasets	Cl_R	d	N	Datasets	Cl_R	d	N
Atom	2	3	800	Tetra	4	3	400
Ring	2	3	1000	Twodi.	2	2	800
Dem-c.	2	2	600	WingN.	2	2	1016
Engyi.	2	2	4096	Art1	4	2	400
Glass	7	9	214	Art2	2	2	1000
Hepta	7	3	212	Art4	2	2	200
Lsun	3	2	400	Art5	9	2	900
Pima	2	8	768	Art6	4	8	400
Target	6	2	770				

Table 2. Comparison between Hierarchical Clustering of map using Ward criterion and a new measure. The number following the good classification rate (purity rate) indicates the number of subsets provided by map clustering.

Datasets/ %	AntTree-W	AntTree-N-W	AntTree
Atom (2)	85.87 (5)	99.9 (7)	83.5(8)
Ring (2)	97.8 (6)	81.5 (5)	71.5(6)
Demi-c. (2)	58.833 (2)	72.67 (4)	74 (3)
Engyt. (2)	74.14 (5)	88.04 (7)	95.1 (10)
Glass (7)	38.32 (5)	59.81 (6)	45.33 (9)
Hepta (7)	43.4 (4)	43.4 (4)	72.6 (6)
Lsun (3)	55 (3)	93 (5)	94(4)
Pima (2)	67 (5)	72.4 (5)	66.67 (8)
Target (6)	83.25 (5)	94.42 (6)	81.81 (9)
Tetra (4)	62.5 (3)	81.75 (5)	93.5 (3)
Twodi. (2)	100 (4)	100 (5)	99.87 (7)
WingN. (2)	95.67 (3)	87.11 (5)	93.60 (6)
Art1 (4)	50.5 (4)	84.75 (4)	76.5 (8)
Art2 (2)	94.9 (4)	97.7 (4)	97.9 (6)
Art4 (2)	100 (3)	100 (5)	98 (4)
Art5 (9)	31.78 (4)	50.33 (6)	51.55 (8)
Art6 (4)	24.25 (2)	78.75 (4)	93 (4)

Table 2 lists the classification accuracy obtained with different methods; the purity rate results were provided. Using a new measure $Dist_{N-W}$, we observe that the results are generally better than the clustering map using traditional ward measure. Taking into account the topological order improves significatively the results. With our hybrid model (with the new measure 3) no index is necessary to obtain the optimal partition.

5 Conclusions and Perspectives

We have presented in this paper a new model for data clustering which is inspired from the self-assembly behavior observed in real ants. This approach is competitive with the precedent standard method which proves the interest to hybridize AntTree with SOM method. The first results that we have obtained are encouraging, both in terms of quality and processing times. The self-assembly model introduces a new ants-based computational method which is very promising. There are many perspectives to study after this series of results. The first consists on approving the similarity measure which seems to be an important tool in clustering map process. As far as clustering is concerned, we would like to introduce some heuristics in order to evaluate their influence on the results, like removing clusters with a small number of instances.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)
2. Kohonen, T.: Self-organizing Maps. Springer, Berlin (2001)
3. Goss, S., Deneubourg, J.-L.: Harvesting by a group of robots. In: Varela, F., Bourgine, P. (eds.) Proceedings of the First European Conference on Artificial Life, Paris, France, pp. 195–204. Elsevier Publishing, Amsterdam (1991)

4. Lumer, E.D., Faieta, B.: Diversity and adaptation in populations of clustering ants. In: Cliff, D., Husbands, P., Meyer, J.A., Stewart, W. (eds.) Proceedings of the Third International Conference on Simulation of Adaptive Behavior, pp. 501–508. MIT Press, Cambridge (1994)
5. Anderson, C., Theraulaz, G., Deneubourg, J.L.: Self-assemblages in insect societies. Insectes Sociaux 49, 99–110 (2002)
6. Lioni, A., Sauwens, C., Theraulaz, G., Deneubourg, J.-L.: The dynamics of chain formation in oecophylla longinoda. Journal of Insect Behavior 14, 679–696 (2001)
7. Theraulaz, G., Bonabeau, E., Sauwens, C., Deneubourg, J.-L., Lioni, A., Libert, F., Passera, L., Solé, R.-V.: Model of droplet formation and dynamics in the argentine ant (linepithema humile mayr). Bulletin of Mathematical Biology (2001)
8. Azzag, H., Guinot, C., Venturini, G.: Data and text mining with hierarchical clustering ants. Swarm Intelligence in Data Mining, 153–189 (2006)
9. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. advanced reference series:Computer Science (1988)
10. Blake, C.L., Merz, C.L.: Uci repository of machine learning databases. Technical report, University of California, Department of information and Computer science, Irvine, CA (1998), ftp://ftp.ics.uci.edu/pub/machine-learning-databases
11. Ultsch, A.: Clustering with SOM: U*C. In: Proc. Workshop on Self-Organizing Maps, Paris, France, pp. 75–82 (2005), http://www.uni-marburg.de/fb12/datenbionik/

Author Index