

The Evolutionary Transition Algorithm: Evolving Complex Solutions Out of Simpler Ones

Tom Lenaerts, Anne Defaweux, and Jano van Hemert

Abstract. Capturing the metaphor of evolutionary transitions in biological complexity, the Evolutionary Transition Algorithm (ETA) evolves solutions of increasing structural and functional complexity from the symbiotic interaction of partial ones. In this chapter we show that the ETA indeed captures this idea and we illustrate this on instances of the Binary Constraint Satisfaction problem. The results make the ETA a promising optimization approach that requires more extensive investigation from both a theoretical and optimization perspective. We analyze here, in depth, some of the design choices that are made for the algorithm. The analysis of these choices provides insight on the plasticity of the algorithm toward alternative choices and other kinds of problems.

1 Introduction

In biology, evolutionary transitions theory [14, 15] provides a generalized explanation of how organisms of increasing complexity may have emerged from the interaction of simpler life forms. The Evolutionary Transition Algorithm (ETA), presented here, captures this metaphor to create structurally and functionally more complex solutions from the combination (interactions) of simpler ones (solutions that solve

Tom Lenaerts

ML-group, Département d'Informatique, Université Libre de Bruxelles,
Boulevard du Triomphe CP212, 1050 Brussels, Belgium
e-mail: tlenaert@ulb.ac.be

Anne Defaweux

COMO, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
e-mail: adefaweux@gmail.com

Jano van Hemert

National e-Science Centre, The University of Edinburgh,
15 South College Street Edinburgh EH8 9AA, UK
e-mail: jano@vanhemert.co.uk

a smaller part of the problem, for example). As such, the goal of this chapter is to show how a particular biological metaphor was transformed into an algorithm with definite potential. Future investigations will attempt to show its relevance for real-world applications.

Apart from its general contribution to the development of nature-inspired algorithms, this new evolutionary algorithm was introduced to address some limitations of the standard genetic algorithm (GA). One of these limitations follows from the dependencies between the variables in the solution representation. In the standard GA, new genotypes are produced by the recombination of existing ones. To avoid the disruption of good building blocks, one needs to ensure that correlated parts of a solution are close to one another in the genotype. For many problems, determining the internal structure of the genotype is a problem on its own since the interdependencies between the variables of the problem are not known. The compositional mechanism implicit to the ETA provides a way to discover the closely related variables of the problem during the evolutionary process while at the same time looking for the solution to the complete problem. An additional motivation follows from the difficulty of scaling the GA to evolve larger solutions for more extensive problems. By taking the compositional approach the evolutionary process can first focus on solving parts of the problem, which can be combined later just like the modular watch constructed by Simon's watchmaker [22].

We show here, using the Binary Constraint Satisfaction problem (BINCSP) as an illustration, that the ETA evolves increasingly complex solutions from the interactions of simpler evolving solutions. The results for BINCSP confirm that the ETA is promising approach that requires more extensive investigation from both a theoretical and practical optimization perspective. Especially decision problems in general [11] and BINCSPs in particular seem to be very well suited for this new evolutionary algorithm. We provide an in depth analysis of the design choices that are made for the algorithm: choices related to the configuration of the initial population, the introduction of a decomposition operator which breaks down more complex solutions into simpler components and the impact of the transition condition on the performance of the algorithm.

This chapter is partitioned as follows. Before discussing the algorithm, which will be explained using BINCSP, a non-exhaustive overview of related work is provided. Afterwards, a set of BINCSP simulations and their results are shown and explained. Apart from a brief overview of earlier results, we provide new results on particular design choices that can be made for the ETA. Finally, a summary is provided and some conclusions are drawn concerning the usefulness and the future of this algorithm.

2 Related Work

There are two ideas behind the compositional search approach: the first idea concerns the use of symbiotic relations to identify good collaborations, and the second idea is concerned with the aggregation of complex solutions from the interaction of partial ones. The related work listed here falls under either one or both of these

categories. It is necessary to note that what we are going to provide is not a full list but a rough picture of the major research work related to the topic. Consequently, certain publications which are very similar to those mentioned here might be missing.

The very first GA-related reference that can be found on the evolution of complete solutions from the combination of partial ones is the work on the messy GA [5, 6, 7]. The messy GA has the structure of a classical GA. When two partial solutions are selected for reproduction in the messy GA, their combined genetic material creates a bigger solution that is defined at a higher level of complexity. Messy GAs therefore have the idea of combining partial solutions. However, this approach differs from the one we propose here: fitness effects caused by the interactions of the partial solutions are not taken into account. Partial solutions are evaluated and selected on their own as in the classical GA. It is during the process of reproduction that combination occurs. Messy GAs therefore lack the idea of a transition that operates on the behavior of good symbionts.

Other approaches focus on the collaboration between partial solutions to construct a full solution for a particular problem. In the Parisian Approach [17, 18] for example, the algorithm intends not to evolve a full solution to a problem but rather a collection of partial solutions that together solve the entire problem. This approach takes an additional step into the direction of compositional evolutionary search. However, the Parisian approach lacks a transition step that merges the set of partial solutions into a full solution. In this respect, the Parisian approach is more related to previous work on multilevel selection [13].

Other approaches introduce an evolutionary “divide and conquer” mechanism like the cooperative co-evolutionary GA [16, 30]. In this algorithm, the problem consists of a collection of sub-problems that can be solved in isolation. Afterwards these partial solutions can be recombined into a full solution for the global problem. The major difference with the ETA resides in the explicit divide and conquer framework. In the cooperative co-evolutionary GA, the process divides the problem into collaborating sub-problems; for each sub-problem, the algorithm evolves solutions that collaborate with one another in order to address the entire problem. The way to divide the problem can become an issue when the problem structure to learn is not trivial or cannot be easily identified.

Finally, a truly compositional approach that uses both the concept of symbiosis and transitions is the Symbiogenetic Model (SEAM) [27, 28, 29]. This model considers a population of partial solutions that interact with one another through the mechanism of symbiosis. In this population, the good symbiotic relations are identified and produce a new partial solution through a transition. This model initially appears very similar to ETA. There is, however, a major difference between SEAM and ETA: SEAM does not generate a succession of populations by reproducing the parent population through the mechanism of fitness proportional selection. It randomly selects pairs of individuals and places them into a symbiotic relation. It then replaces the symbiotic relation by a new individual when it performs better than the parents in isolation. The condition under which this transition occurs also differs from ETA: it uses the concept of pareto optimality with respect to an evolutionary context to define whether a symbiosis performs better or worse than the parents

alone. The result of this approach requires the model to include features about the structure of the problem that needs to be solved like for instance the degree of modularity of the problem. Furthermore, in order to work, it requires the problem to be fully hierarchically decomposable, which covers only a subset of the structured problem class.

Finally, the Hierarchical Genetic Algorithm (HGA) [23] is an algorithm very similar to SEAM. The major difference resides in the fact that the HGA introduces, in its implementation, explicit information concerning the hierarchical nature of the problem that needs to be solved (by hard coding the hierarchical search in the process and therefore disclosing the emergent aspect present in SEAM) and optimizes the sampling approach so that only the best samples are present for evaluation at all time, considerably speeding up the search process.

3 BINCSP: The Illustrating Test Case

Before we explain the algorithm we briefly define the problem that is used to illustrate the technique.

Constraint Satisfaction Problems (CSP) [24] are NP-complete problems that are defined by a set of variables X associated with possible domain values D and a set of constraints C defined on this set of variables that specify which combinations of assignments can or cannot occur. The problem consists of finding an assignment to the whole set of variables from the associated domain values so that all constraints are satisfied. This makes the problem a decision problem [11]. If an assignment is impossible then the corresponding problem is said to be unsolvable. A variant of this problem is the BINCSP, where each constraint is defined on at most two variables. This introduces no restriction on the general form of CSP as every CSP can be rewritten into a BINCSP and vice versa [19].

Let us take as an illustration the following BINCSP: consider a set of six variables: $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ all taking values in $D = \{1, 2, 3\}$. We consider the following set of constraints:

$$C = \{(x_1 \neq x_2), (x_2 \neq x_3), (x_3 \neq x_1), \\ (x_4 \neq x_5), (x_5 \neq x_6), (x_6 \neq x_4), \\ (x_1 = x_4), (x_2 = x_5), (x_3 = x_6)\} \quad (1)$$

This constraints setup consists of 9 binary constraints. Each binary constraints defines a relation on 2 variables. For each pair of variables, only one binary constraint may be defined.

The problem consists of finding the right assignment for the variables so that all these constraints are satisfied. We denote the assignment of one variable $x_i \in X$ with value $d \in D$ by $\langle i, d \rangle$ where i is the index of the variable we consider. Using this notation, we represent the simultaneous assignments of variables x_1, x_2 and x_4 with respective values v_1, v_2 and v_4 as

$$(\langle 1, v_1 \rangle, \langle 2, v_2 \rangle, \langle 4, v_4 \rangle)$$

4 General Description of the ETA

As can be seen in Figure 1, the ETA follows a classical *evaluate-select-reproduce* evolutionary loop. As a GA it works with a limited fixed size population which is evaluated and reproduced at each iteration. Before the evaluation phase, all (free) individuals are paired up. This grouping can be performed in different ways. Here we opted for the most simple form, i.e. random pairing. Once each individual has a partner, a fitness score is assigned based on their own properties and the properties they obtain from the interaction with the other individual. The combination of both properties is referred to as the induced phenotype, which will be explained in more detail later. The fitness score will guide the selection process as in the GA and selected individuals can reproduce in three different ways:

1. They can just reproduce their own genetic information, as in the GA.
2. To maintain interesting links, we also provide the possibility that both individuals and their interaction are reproduced.

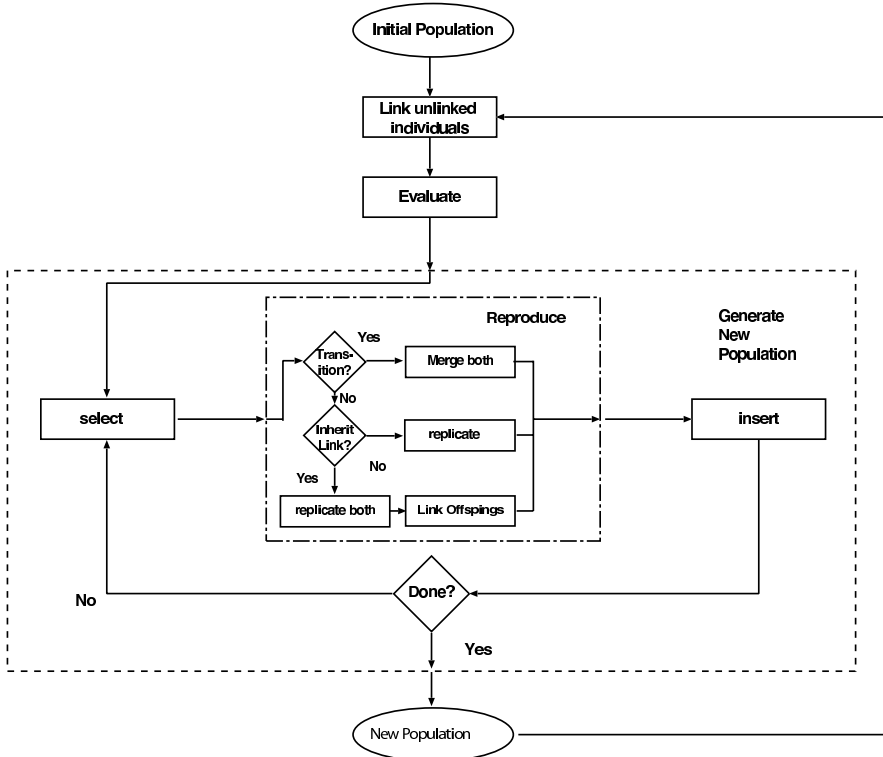


Fig. 1 Schematic overview of the ETA. The algorithm consists of three phases: 1) the pairing up of partial solutions, 2) the evaluations of the interacting solutions and 3) the reproduction of the solutions. Every iteration of these three steps produces a new population.

3. Good symbionts which solve the partial problem completely are reproduced as individuals at a higher level of complexity, meaning that their genetic information is combined into a new individual.

Repeated iterations of this process produce individuals of increasing complexity that solve an increasing proportion of the problem.

The following sections explain each part of the ETA in detail.

4.1 *Representation of Basic Elements and Partial Solutions*

A partial solution for BINCSP is a compound label which is not defined on all variables of the variable set X . A compound label on, for instance, variable x_1, x_3 and x_7 where each variable is instantiated with values d_1, d_3 and d_7 respectively is denoted by

$$(\langle 1, d_1 \rangle, \langle 3, d_3 \rangle, \langle 7, d_7 \rangle).$$

In the current description of ETA, we make a distinction between a *partial solution* and a *solution*: A partial solution for the BINCSP is a compound label which does not assign all variables of the variable set defined by the BINCSP. When all variables are assigned, we obtain a solution to the problem. The underlying idea behind this distinction is that a solution defining a full genotype is the achievement of the compositional search algorithm. We therefore speak of partial solution as long as the algorithm is still in the process of evolving complexity, meaning it is still trying various combinations of partial solutions with the hope of obtaining a full solution to the problem. The representation of a (partial) solution is also called the *genotype*.

In its most basic form, a partial solution should correspond to the assignment of exactly one variable. However, such solutions are meaningless by themselves as each constraint is defined on exactly two variables. As a consequence, they need to be evaluated during their interaction with other individuals.

The fact that basic elements of length 1 have a zero fitness has consequences when considering a fitness-proportional selection model. Indeed, initializing the population with length 1 partial solutions means that none of them can expect a positive fitness, i.e. be selected, without interacting with others. This lead us to think that the most basic units of selection are not of length 1 but rather of at least length 2. In our simulations, we evaluated both scenarios (see simulations section for more details) and observed that the length of the initial partial solutions has little impact on the evolutionary process itself.

4.2 *Interactions and the Induced Phenotype*

In the ETA, pairs of (partial) solutions are bound by *symbiotic relations*. When bound the two solutions exchange information. The result of this exchange is captured in the *induced phenotype* of the solution which can alter positively or negatively the fitness of each of the solutions. We explicitly make the distinction between phenotype and *induced phenotype* to stress that the latter is phenotype produced

from both the genetic information of an individual and the effect of external factors, like other population members and the environment, on the final phenotype.

Since the induced phenotype of an individual is constructed by combining the information contained in its own genotype with the information contained in the genotype of the other members of the symbiotic relation, conflicts in this information may occur. In case of a conflict (i.e. two different values are assigned by the partial solutions to the same variable), a conflict mediation strategy is applied here that randomly chooses one of the possible values from the set of conflicting values.

As explained earlier, interactions are pairwise and created randomly using the population of solutions, i.e. the solution can be linked to any other solution in the population. Alternative strategies, based on multilevel selection [13] are currently under investigation.

Let us briefly illustrate the process by which the induced phenotype is constructed in the context of the BINCSP example. Suppose we have a partial solution s that interacts with another partial solution sp . Both partial solutions are represented through the following genotypes:

$$\begin{aligned} s &= \langle 1, \alpha_1 \rangle, \langle 2, \alpha_2 \rangle, \langle 9, \alpha_9 \rangle \\ sp &= \langle 2, \beta_2 \rangle, \langle 4, \beta_4 \rangle, \langle 9, \alpha_9 \rangle \end{aligned}$$

In these two solutions, we observe that s is well defined for the variables 1, 2 and 9 and sp for the variables 2, 4 and 9. The value assigned to variable 9 is the same for both partial solutions. However, the values assigned to variable 2 are not the same, meaning that the solutions have a conflict to solve for this variable: we need to choose between value α_2 or β_2 . The conflict mediation strategy randomly chooses between one of the conflicting values yielding in our specific example two possible induced phenotypes:

$$\varphi(s, sp) = \langle 1, \alpha_1 \rangle, \left\langle 2, \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} \right\rangle, \langle 4, \beta_4 \rangle, \langle 9, \alpha_9 \rangle$$

where α_2 or β_2 is chosen randomly with equal probability.

After conflict resolution, the resulting phenotype assigns exactly one value for each variable that was assigned in s or sp . This phenotype is then used by the fitness function to evaluate the partial solution. The details concerning the fitness function and the way the partial solutions are evaluated in the context of BINCSP is provided in the following section. Note first that the induced phenotypes of the members of the symbiotic relation do not have to be the same for both partners since the process decides randomly which value to use for each member independently. Thus both s and sp can select α_2 , β_2 or make different choices (s takes β_2 and sp takes α_2 or vice versa).

4.3 Evaluation Functions

We consider two evaluation functions for BINCSP. The first function evaluates the quality of the solution's phenotype with respect to the entire constraints set. The

second function is restricted only to the constraints that are defined on the variables that are present in the partial solution.

Definition 1. The *classical fitness evaluation for the ETA on BINCSP* of a solution s , $f_{et}(s)$ which interacts with a symbiotic partner sp and for which the induced phenotype of the interaction for s with sp is denoted by $\varphi(s, sp)$ is given by following equation:

$$f_{et}(s) = \frac{1}{|C|} \sum_{c \in C} eval(\varphi(s, sp), c) \quad (2)$$

where $eval_c(\varphi, c)$ checks whether the compound label φ satisfies the constraint c or not.

We have slightly adapted the notion of constraint satisfaction as we also need to address partial solutions that have no assignment or incomplete assignment definitions for a given constraint. We therefore suppose that a compound label satisfies a constraint c if:

- it *covers* the constraints, i.e. there are assignments for all the variables comprised in c in the compound label of the partial solution.
- the assignments do not violate the constraints.

This is summarized in the following equation:

$$eval(\varphi, c) = \begin{cases} 1 & \text{if } (\varphi \text{ covers } c) \text{ and } (satisfies(\varphi, c)) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

This fitness value gives an estimation of how the partial solution scores with respect to the entire set of constraints. This means that small partial solutions, even if they are made of good genetic material, cannot receive a high reward value in comparison with larger partial solutions which, by assigning more variables increase their chance of improving their score, even if some of these variables are incorrectly assigned. To obtain an objective value of how a partial solution scores with respect to the sub-problem of constraints that are covered by the partial solution, we therefore also consider a restricted version of this fitness function which limits the evaluation to the covering set of constraints for this partial solution.

Definition 2. The *covering set* of constraints for a given genotype γ on the constraints set C is denoted by $C_{cov}(\gamma)$ and consists of all the constraints of C which are covered by γ :

$$C_{cov}(\gamma) = \{c \in C \mid \gamma \text{ covers } c\}.$$

With this covering set, we can now define the covering fitness function.

Definition 3. The *covering fitness function* of a solution s interacting with sp is denoted f_{cov} and is given by the following equation:

$$f_{cov}(s) = \frac{1}{|C_{cov}(\varphi(s, sp))|} \sum_{c \in C_{cov}(\varphi(s, sp))} eval(\varphi(s, sp), c) \quad (4)$$

We typically use the first evaluation function to guide the evolutionary process, that is, for selecting the best solutions for reproduction, while the second measure is used as an observation measure for deciding whether the interacting partial solutions are merged into one new solution.

The motivation for the use of two different fitness measures is the following: the classical fitness function f_{et} will tend to favor cooperation as bigger partial solutions are more likely to receive a greater fitness than smaller partial solutions and we wish the system to evolve a complete solution to the problem and therefore to favor increase in the length of the solution genotype. The covering fitness does not capture information with respect to the entire problem since partial solutions of any size (bigger than 2) can have maximum fitness, which is 1. This function prevents the system from evolving larger solutions as they will not perform better than smaller partial solutions. This is the reason why we choose the classical fitness for the selection process. However, when it comes to evaluating the quality of a symbiotic relation, the covering fitness contains much more information than the classical fitness. By limiting the evaluation to the problem actually covered by the symbiotic relation, we obtain a measure of how well the relation performs with respect to the sub-problem it addresses. The covering fitness therefore allows us to assess whether a symbiotic relation should be preserved for future generations through the mechanism of transitions.

4.4 Replication and Transitions

When selected for reproduction, a partial solution will have three possible options (examples are based in BINCSPP problem defined in Section 3):

1. Simple replication: In this case, one solution is replicated, that is, its genotype is copied and passed on to its offspring (with possible mutations: the value of variable 1 changed from 1 to 2 in the example below). It occurs when the solution scored sufficiently high to survive one more generation but did not perform good enough within its symbiotic relation to see this relation survive or preserved through a transition.

$$s = \langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 2 \rangle \longrightarrow s' = \langle 1, 2 \rangle, \langle 2, 2 \rangle, \langle 3, 2 \rangle$$

2. Inherit symbiotic link: In this situation, the solution not only replicates itself but also trigger the replication of its symbiotic partner. The symbiotic link that binds both parents is also inherited by their offspring. This reproduction mode is a step toward a transition; it is, however, a reversible step since both individuals still exist independently. In the situation of BINCSPP, we did not implement any special condition for this mode to occur and it can therefore occur randomly at each generation.

$$\left\{ \begin{array}{l} s = \langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 2 \rangle \\ sp = \langle 2, 1 \rangle, \langle 5, 1 \rangle, \langle 6, 3 \rangle \end{array} \right\} \longrightarrow \left\{ \begin{array}{l} s' = \langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle \\ sp' = \langle 2, 1 \rangle, \langle 5, 1 \rangle, \langle 6, 3 \rangle \end{array} \right\}$$

In the example, the covering fitness of the induced phenotype of both individuals (s and sp) is less than 1. As a consequence, no transition occurs. Yet we can replicate both partners and their link to check if we can improve it by choosing, for instance, another value for variable 2. Again, mutation might occur when the individuals are reproduced, as is the case here for variable 3 in individual s .

3. Transitions: This case occurs when the induced phenotype of an individual actually solves the sub-part of the problem completely, i.e. when the covering fitness is equal to 1.0. In this case, the outcome of the symbiotic relation is replicated instead of the original genotype of the solution. This means that the induced phenotype becomes the genotype of the offspring.

$$\left\{ \begin{array}{l} s = \langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle \\ sp = \langle 2, 1 \rangle, \langle 5, 2 \rangle, \langle 6, 3 \rangle \end{array} \right\} \longrightarrow s' = \langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 2 \rangle, \langle 5, 2 \rangle, \langle 6, 3 \rangle$$

In this example, the induced phenotype of s can be for instance:

$$\varphi(s, sp) = \langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 2 \rangle, \langle 5, 2 \rangle, \langle 6, 3 \rangle$$

The covering fitness of $\varphi(s, sp)$ is 1. As a consequence a transition occurs, making the genotype of s' the same as the induced phenotype produced by the interaction of s and sp .

The second reproduction mode might look like an unnecessary step in the entire process as it occurs randomly without any relation to the transition itself. Nevertheless, this mode may be very important for various reasons:

1. Even though this event occurs randomly, it is performed on individuals selected according to some fitness proportionate selection scheme. In this way, only individuals which are potentially better get the chance to reproduce their symbiotic partner. Consequently, it provides the conflict mediation strategy more chances of finding a good combination.
2. It is a way to keep potentially good genetic material in the population as the symbiotic partner might possess the genetic information that actually causes this solution to be selected. In other words, their high fitness is a result of the combination of both partners. Therefore, we need to explore their relationship further.
3. Through a possible error during copying, a nearly good combination may become the right combination. See the examples above.

Further evaluation is required for this part of the algorithm.

The third reproduction mode, the transition, creates solutions of higher complexity: Small genotypes aggregate into larger genotypes that in turn, through interactions with other genotypes, can aggregate into more complex genotypes until a full solution emerges.

5 Evaluation of the ETA on Concrete BINCSP Instances

The objective of the current simulations is to demonstrate that ETA succeeds in building complex solutions that can actually solve BINCSP problems. Since we

will evaluate BINCSP instances of different complexities we may derive the subset of problems on which the ETA performs well. Moreover, we examine here how the ETA behaves when we change certain operations or conditions inherent to the algorithm.

5.1 Simulation Setup

Each simulation has an almost similar setup and uses the same set of BINCSP instances. These instances are randomly generated using RandomCSP package provided by [26] and produce instances with common statistical properties as the one used in [25]. This allows us to compare the ETA with other coevolutionary approaches that were tested on this set (see [2]).

The complexity of the instances created by the tool can be tuned by two parameters p_1 and \bar{p}_2 . We let p_1 and \bar{p}_2 vary from 0.1 up to 0.9 with a step size of 0.2. In Table 1 these instances are classified according to their problem complexity into either the easy or difficult class. For each combination of p_1 and \bar{p}_2 , 25 different random problem instances for a BINCSP of 15 variables, each taking values in a domain of size 15, were generated.

For each of the 25 problem instances, we perform 10 runs (each run using a different random seed). The maximum amount of generations is set to 100000. This means that for each setup of p_1 and \bar{p}_2 , 250 runs are performed in total. This amount of runs for each problem instance should be sufficient to evaluate the algorithm.

The initial population is created such that it contains all partial solutions of length 1, i.e. every variable-value combination is present once. We refer to this population as a *sound population*.

During replication, the genotype may mutate with probability 0.001, i.e. the value of a variable may be altered to some random value in the domain of the variable. Transitions are performed when the covering fitness is 1.0.

The speed of the algorithm is determined as follows: Let e be the number of evaluations in the run r where the optimal solution was found and e_{max} the maximum number of evaluations in the simulation run. We wish to evaluate the speed of the algorithm on a scale of $[0, 1]$ where higher values mean that the algorithm needed less evaluations to find a solution than lower values and the value 0 corresponds to

Table 1 Identification of easy and difficult BINCSP instances

		Tightness (\bar{p}_2)				
		0.1	0.3	0.5	0.7	0.9
Density (p_1)	0.1	easy	easy	easy	easy	difficult
	0.3	easy	easy	easy	difficult	unsolvable
	0.5	easy	easy	difficult	unsolvable	unsolvable
	0.7	easy	easy	difficult	unsolvable	unsolvable
	0.9	easy	difficult	unsolvable	unsolvable	unsolvable

the situation where no solution was found in the given time. We have therefore the following equation for the speed of the run r :

$$speed(r) = \begin{cases} 0 & \text{if no solution found during } r \\ \frac{e_{max}-e}{e_{max}} & \text{if the optimum is found in } e \text{ evaluations} \end{cases} \quad (5)$$

5.2 Observations

We collected different types of data over all the runs:

- The size of the solutions over time: We monitor the maximum and minimum size of the solutions over time, together with the amount of overlap between the symbiotic partners.
- The fitness of the solutions over time: We trace the best fitness, the average fitness, the worst fitness in the population and the fitness of the largest individual (i.e. the individual that achieved the highest level of complexity). We collect the values of the classical fitness as provided by Equation 2 for these observations.
- Success ratio: This gives the ratio of successful runs over the total amount of runs.
- Average number of generations required to reach a complete solution for all BINCSP instances of a particular complexity: Since the population is evaluated exactly once in each generation, this number is strongly correlated to the speed of the algorithm (see Section 5.1).

5.3 Previous Results

In this section, we summarize the results of the simulations of the ETA on BINCSP discussed in [3, 4], for reasons of comparison.

Table 2 shows the success ratio, average number of generations to find a solution and standard deviation in this number for each BINCSP setup. The plots in Figures 2 and 3 show the evolution of fitness and genotype complexity over time for 4 easy and 4 difficult BINCSP instances (see Table 1).

From these simulations the following conclusions were drawn. The results in Figure 2 show that the transition model succeeds in finding a solution for the easy BINCSP as the fitness converges rapidly to 1.0. They also show that, for difficult BINCSP (after over 2000 generations), the algorithm converges to a population that contains a combination of partial solutions that solve over 90% of the constraints but are unable to resolve the conflicts between the interacting partial solutions in order to solve the entire set of constraints.

We observe in Figure 3 that for easy BINCSP, the intersection (reflecting the number of variables that are in common between the two solutions) is small while exploring, which corresponds to a high complementarity. Indeed for the easy BINCSP (left column of Figure 3) we see that initially the intersection between the two solutions is rather small. Once the algorithm begins to converge toward a

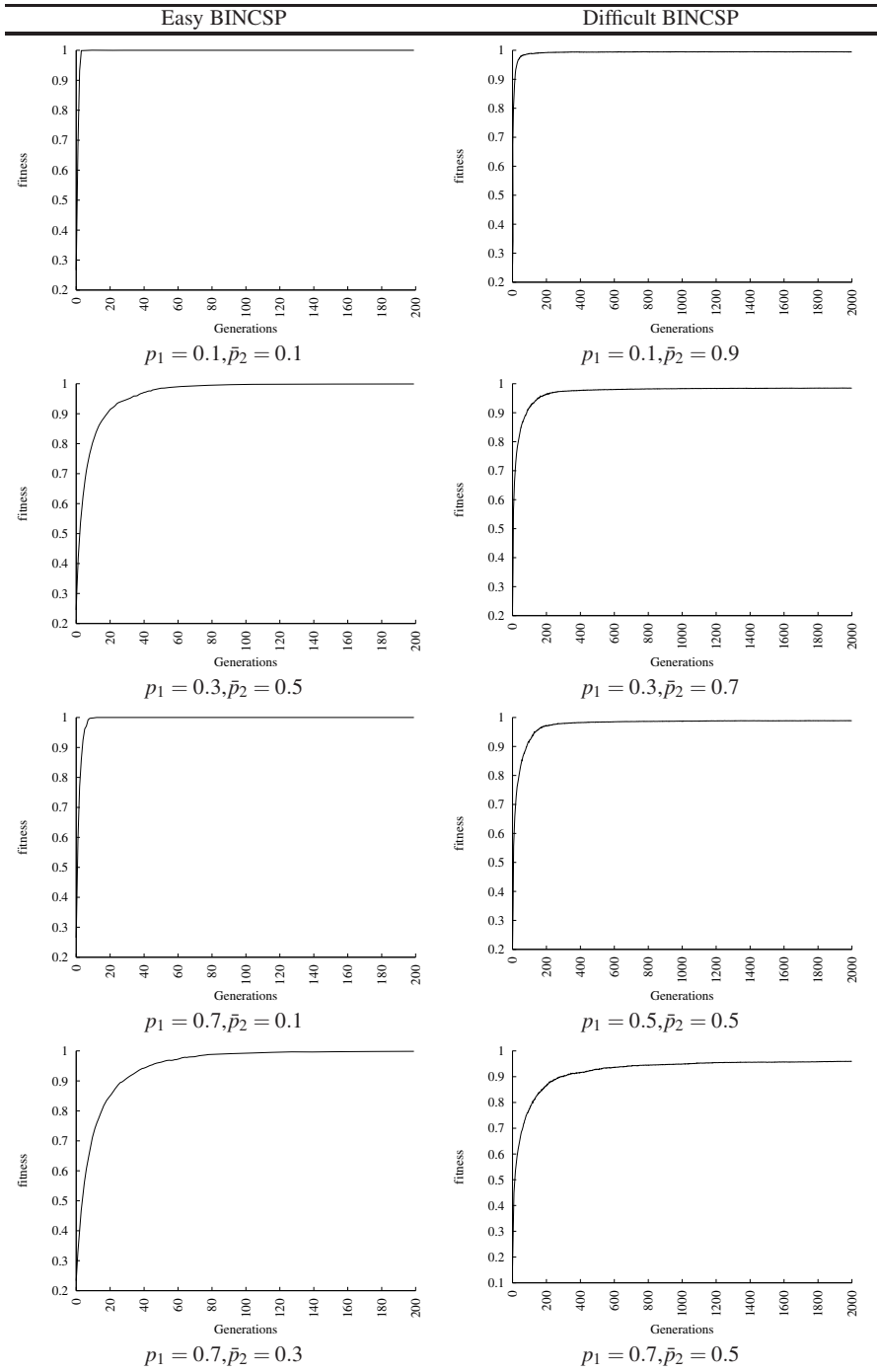


Fig. 2 Evolutionary dynamics of the ETA. All plots show the evolution of the best fitness.

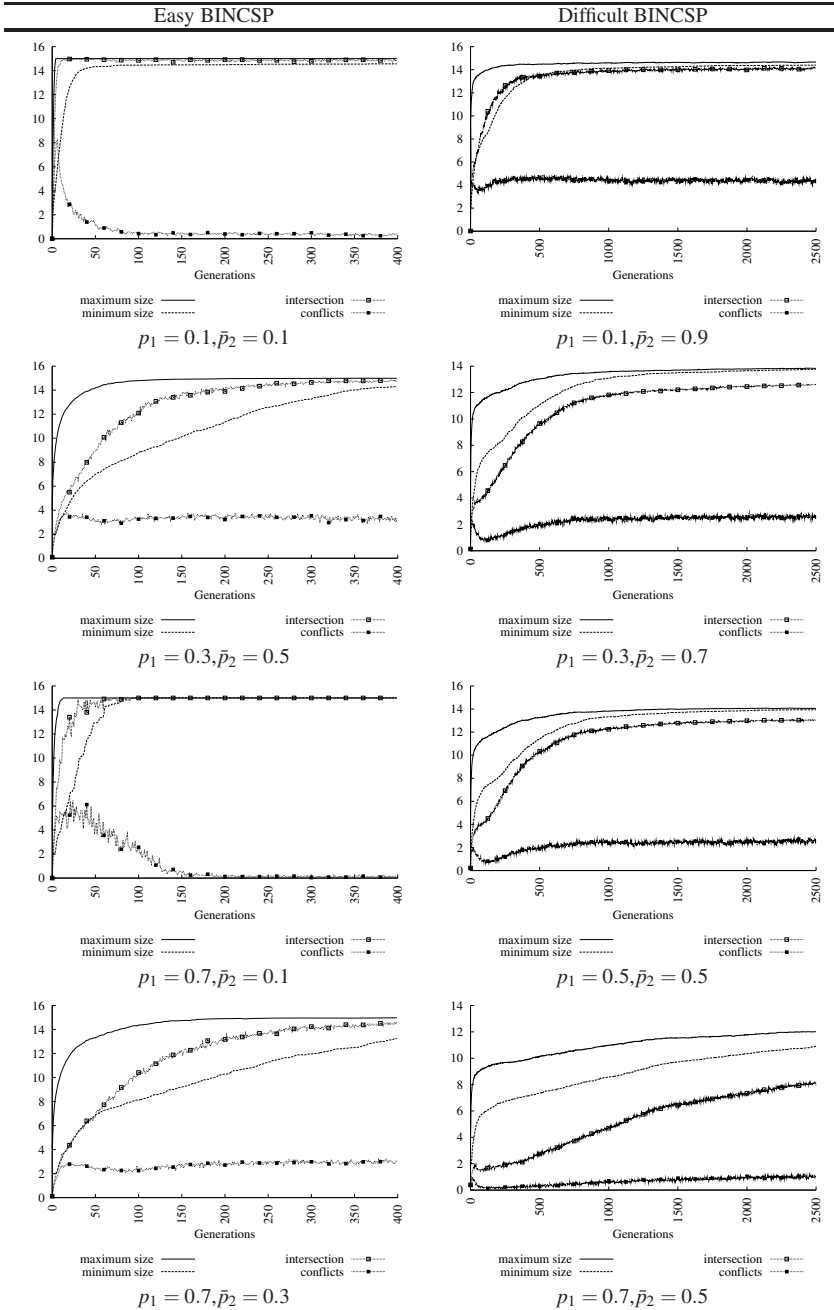


Fig. 3 Evolutionary dynamics of the ETA: All plots show the evolution of the genotype complexity. Each plot contains the maximum and minimum size of the individuals in the population, the average number of variables in common (intersection) and the average number of remaining conflicts (conflicts).

Table 2 Results of the simulations. Each square contains the success ratio, average number of generations, standard deviation and average number of evaluation (between braces).

Density	Tightness					
	p_1	0.1	0.3	0.5	0.7	0.9
0.1	1.0	1.0	1.0	1.0	1.0	0.992
	(4)	(5)	(7)	(13)	(3627)	(3627)
	[0]	[0]	[1]	[3]	[1224]	[1224]
	{900}	{1125}	{1575}	{2925}	{816075}	{816075}
0.3	1.0	1.0	1.0	0.28	-	-
	(5)	(12)	(43)	(11758)	-	-
	[0]	[3]	[25]	[-]	-	-
	{1125}	{2700}	{9675}	{2645550}	-	-
0.5	1.0	1.0	0.54	-	-	-
	(6)	(25)	(11409)	-	-	-
	[1]	[8]	[-]	-	-	-
	{1350}	{5625}	{2567025}	-	-	-
0.7	1.0	1.0	0.29	-	-	-
	(8)	(61)	(8673)	-	-	-
	[1]	[29]	[-]	-	-	-
	{1800}	{13725}	{1951425}	-	-	-
0.9	1.0	0.972	-	-	-	-
	(11)	(2335)	-	-	-	-
	[2]	[4902]	-	-	-	-
	{2475}	{525375}	-	-	-	-

solution, this complementarity decreases and the number of conflicts in the intersection stabilizes. The positive number of the conflicts in this intersection for certain easy BINCSP setups reflects a low but existing variation in the genotype population throughout the evolutionary process. One can also see in the left column that when the population converges completely, all conflicts disappear. In contrast, when $p_1 = 0.3$ and $\bar{p}_2 = 0.5$ (and also $p_1 = 0.7$ and $\bar{p}_2 = 0.3$), there are still alternatives present in the population even though an optimal solution is found.

When we look at difficult BINCSP in the same figure, we observe quite a similar evolution (right column of Figure 3). However, in these cases, the individual size and the intersection (almost) never reaches the maximum number of variables. Since the maximum size is not reached, each genotype needs a symbiotic partner to cover the entire variable set, implying some complementarity to achieve the final solution. Yet the problem is that for these difficult problems it is not sufficient to glue together complementary solutions (see three last plots in the right column of Figure 3). In those cases, the number of variables that are common between the partners in the symbiotic relation is even less than the minimum size of the individuals in the population. This shows that partial solutions seem to agree on certain aspects of the solution yet are divided over the other variables and their assignments that are required to reach the complete solution. So here we have an exploration issue that needs to be solved. The low number of conflicts shows that the ETA evolves complementary partial solutions that specialize in solving two different sub-sets of

the constraints set too. However, the difficulty of solving the entire problem makes it impossible to resolve certain conflicts among these variables - the results show that the evolutionary process is stuck in some local optimum, which may require other mechanisms to resolve the issue.

The results in Table 2 were compared to other co-evolutionary BINCSP approaches in [4]. The ETA performs relatively well compared to problem-specific EA techniques and outperforms them in one specific case: those instances where the constraint network has some kind of modular structure due to the sparseness of the connection between the constraints. Even when the constraints to learn are difficult (high $\bar{p}2$), the ETA conquers these constraints by solving the subproblems separately and aggregating them once they are solved.

In [3] we analyzed the effect of initial population composition on the results of the ETA. We compared a randomly generated population of partial solutions of size two and a sound population (as used here). The results showed that the initial population has little overall effect on the chance of success, which is interesting as it indicates that our algorithm performs well with a small initial population and hence does not need a large and complete population to achieve good results. We illustrated that these differences are indeed not significant when comparing the speed (using Equation 5) of the algorithm.

5.4 Introducing Decomposition in the ETA

The ETA, as described so far, only creates more and more complex solutions. This might result in a dead-end, possibly explaining why a higher success score on the difficult BINCSP instances was not achieved. It could then be useful to decompose these bad solutions back into smaller individuals so alternative composition paths may be followed. The motivation for not including this from the beginning stems from the description of the biological metaphor. Evolutionary transitions are considered as a one-way process [15]. Once a transition occurs, the components of the new individual are forced to collaborate with one another. Defection leads irremediably to the death of the organism together with all its components. So, biologically, individuals cannot decompose back into lower level units and such a situation, when it happens, leads to the extinction of the individual and all its components. In an optimization context however, we are not limited by this restriction.

To introduce decomposition in the evolutionary process, we slightly adapted the ETA described at the beginning of this chapter so that, at each generation, a certain fraction of the population has the possibility of being decomposed into smaller units. These smaller units of selection can then recombine with other units and evolve a new solution which is defined at a higher level of complexity. The deconstruction (decomposition) therefore introduces a new exploration mechanism that tries other combinations of building blocks that had been identified as potentially good units in former symbiotic relations.

This notion of decomposition is, in a certain sense, related to the backtracking process implicit to depth-first and related search methods [8, 20]. Indeed, these deterministic approaches decompose invalid solutions to reconstruct them afterwards

with valid assignments. In the evolutionary approach, the process is no more deterministic but stochastic [11]. However, the idea of decomposing solutions which lead to a dead-end in the optimization process remains.

In this simulation setup, we observe whether the introduction of decomposition, i.e. deconstruction operation, can affect the quality of the results. We first describe the adaptation of the existing algorithm with the introduction of a decomposition mechanism and then we present the simulation setup and results.

5.4.1 Evolutionary Transition Algorithm with Decomposition

ETA with decomposition introduces two new features to the existing framework of the algorithm: the decomposition condition and the decomposition operator. The *decomposition condition* determines when a given solution should decompose into elementary solutions. This condition may trivially be implemented as a random process but may also relate to observations of the evolutionary process (for example the process is stuck in a local optimum, the solution has lasted many generation without resolving its conflicts with the symbiotic partners) or even to problem-specific aspects. The *decomposition operator* may be implemented in different ways to induce different dynamics in the search process. For example, it can decompose one solution all the way back to its elementary units. Such an operator, ensures the presence of elementary solution units during the entire run. Another possibility consists of decomposing one solution back into 2 (or more) partial solutions, reducing the complexity of the given solution by a factor 2 (or more) but without asking the process to rebuild a complex solution with these partial units from scratch.

Given the operator and the condition, the ETA can now be extended. During the replication process, the selected solutions are passed one by one through the reproduction operator. This operator, as before, verifies if the transition condition is met, that is, if the symbiotic relation performs well enough to emerge a new individual at a higher level of complexity. This part of the replication process is unchanged with respect to the classical implementation of ETA. If the transition condition is not met, which means that the selected solution will self-replicate and will possibly replicate its symbiotic relation, the replication process will first check whether the decomposition condition is met before proceeding with self-replication as before. If the decomposition condition is met, then the solution decomposes to a lower level of complexity.

For the simulations, we consider random decomposition. Each selected solution which does not perform a transition may be decomposed with a certain probability. The choice for random decomposition instead of a more problem-or-process related condition is motivated by the idea that we are mainly concerned with the evaluation of the decomposition process. We want to see if decomposition brings something to the system. Random decomposition, not being problem related, offers an objective view of the induced dynamics of the decomposition process.

When the decomposition condition is met, the selected solution will not replicate as a whole but split itself into two or more parts according to the decomposition operator. In our implementation of ETA with decomposition for this set of simulations, we implement a simple decomposition operator where the solution is divided

```

decompose(solution)
for  $pos := 1$  to  $solution.size$  step 1 do
   $allele := solution[pos]$ 
   $prob := 0.5$ 
  if  $offspring_1.size > offspring_2.size$ 
    then
       $prob := prob \times (1 - \frac{offspring_1.size}{solution.size})$ 
    else
       $prob := prob \times (1 + \frac{offspring_2.size}{solution.size})$ 
  if  $random(prob)$ 
    then
       $offspring_1.add(allele)$ 
    else
       $offspring_2.add(allele)$ 

```

Fig. 4 Pseudo-code for the decomposition operation

into two parts of approximately the same size. This strategy means that we do not really backtrack to the former situation consisting of the two partial solutions that were merged into this parent solution. We obtain, however, a decomposition in terms of complexity as the solution goes back to a former level of complexity with respect to this parent solution. The way the two offspring solutions are generated in corresponds to a random process where, on average, each allele of the parent solution is passed on to one of the two offspring solutions with the same probability. The pseudo-code of the decomposition of a solution is given in Figure 4. The two offsprings that are created are then added to the offspring population.

5.4.2 Simulation Setup

The problem setup is the same as before. The goal here is to compare the transition model without decomposition (see Table 2) with a transition model using decomposition. Both models will use initial sound populations as described in the previous simulation. They also share exactly the same parameters setup, not considering the additional *decomposition probability*. To explore the impact of this parameter on the behavior of the process, we consider 3 possible values for this parameter:

- Small probability for decomposition: the value of the decomposition probability scales from 0.001 to 0.05 which means that, at each generation, a very small part of the population will be decomposed.
- Medium probability for decomposition: the value of the decomposition probability scales from 0.15 to 0.35. In this situation, at each generation, a larger part of the population will be decomposed. However, decomposition remains a rare event as the solutions have more chance to remain unchanged than to undergo a decomposition operation.
- High probability for decomposition: the value of the decomposition is set to 0.5 and above. In this situation, at each generation, the population will perform at

least as much decomposition operations as self-replication operations. This situation corresponds to a situation where exploration is preferred over exploitation.

For each setup of the BINCSP and the decomposition probability, we perform 10 runs each with a different seed on the 25 problem instances that were randomly generated. We collect the same information as for the previous simulation.

5.4.3 Simulation Results

In Table 3, we gather results concerning the success ratio and average number of generations to reach a solution for the 5 difficult test cases identified in Table 1. For

Table 3 Comparison of success ratio between different decomposition setups. Each entry consists of three values: 1) the success ratio, 2) the average number of generations and 3) the standard deviation. The upper row corresponds to the situation of a generic transition model operating with initial sound population and no decomposition, the other rows gives the results for different decomposition probabilities (DP). Results better than the generic model are shown in bold.

		$p_1: 0.9$	0.7	0.5	0.3	0.1
		$\bar{p}_2: 0.3$	0.5	0.5	0.7	0.9
DP:						
No	Generic	0.972 (2335) [4902]	0.29 (8673) [-]	0.54 (11409) [-]	0.28 (11758) [-]	0.992 (3627) [1224]
Low	0.001	0.976 (3106) [5077]	0.276 (4654) [1546]	0.54 (7955) [8128]	0.236 (10534) [33000]	0.992 (2803) [2220]
	0.01	0.992 (2476) [2956]	0.308 (8507) [9492]	0.604 (12993) [10729]	0.236 (8230) [8053]	1.0 (2835) [4201]
	0.05	0.984 (2922) [3869]	0.36 (13705) [14214]	0.58 (11879) [7222]	0.3 (9248) [5126]	1.0 (1457) [2407]
	0.15	0.736 (7869) [2166]	0.232 (13168) [11514]	0.3 (13542) [1246]	0.172 (19029) [16504]	0.884 (3003.5) [4857.5]
	0.25	0.648 (9492) [1191]	0.26 (15437) [12867]	0.248 (12890) [3104]	0.156 (10436) [12701]	0.856 (3471) [6290]
0.35	0.736 (8774) [6220]	0.1 (12836) [9810]	0.26 (17794) [13272]	0.096 (18952) [12820]	0.892 (4223) [7425]	
High	0.5	0.38 (18237) [13186]	0.08 (9509) [3147]	0.076 (25637) [20368]	0.048 (25561) [23721]	0.604 (13627.5) [9136]
	0.65	0.004 (46) [0]	0.0 (-) [-]	0.0 (-) [-]	0.0 (-) [-]	0.02 (39) [14]

each test case, runs were performed for different decomposition probabilities scaling from low to high. The success ratios that are printed bold in the table represent the best results achieved for the corresponding problem setup.

We observe that decomposition influences the results of the ETA. It is not a surprise to see that high values for the decomposition probability yield the worst results. Indeed, high decomposition probabilities result in an almost systematic decomposition of solutions. As a consequence, partial solutions do not have the necessary time to perform conflict mediation or produce new levels of complexity. Medium values do not perform well either. We had expected that the significant increase in exploration provided by medium levels of decomposition would have helped the process in finding the good solutions. However, even the medium range values for probability did not score well. Actually, only low values for the probability, that scale from 0.01 to 0.05 have a positive impact on the simulation results. In this situation, the ETA with decomposition is able to match and even outperform the results of the generic ETA. This leads to the conclusion that, if the decomposition does influence the search process, it should remain a rare event like the standard mutation operator.

To conclude this comparison of the generic ETA with the ETA with decomposition, we performed the Wilcoxon rank-test between the speeds (defined in Equation 5) of the generic ETA and the best performing ETA with decomposition on each test case. These tests which are summarized in Table 4 compare the speed of each run to find an optimum between a generic ETA and the best performing ETA with decomposition. It evaluates whether the difference in success ratio we observed in Table 3 between both algorithms is significant or not in the global optimization process. The null hypothesis of these tests is that the average speeds of both algorithms are equivalent. The speed is computed the same way as in Equation 5 (see page 114) and captures the average speed of each run in a measure between 0 and 1 where 0 means that the run did not find any optimum value in the given time while 1 means that the solution was found from the very first generation. Higher values for the speed mean that the run was able to find a solution faster than others.

In Table 4 we see that the null hypothesis cannot be rejected under the classical 5% confidence interval except for the test case $p_1 = 0.9, \bar{p}_2 = 0.3$. We can conclude from this that, in general, the differences in performance we observed between the generic ETA and the best scoring ETA with decomposition in Table 3 are not significant. So, introducing a small fraction of decomposition at each generation during the evolutionary process may increase the expected chance of success, although this

Table 4 Wilcoxon-Rank Test between the Generic ETA and the ETA with Decomposition (for the decomposition probability DP that scored best)

BINCSP problem	DP	p-value
$p_1 = 0.9, \bar{p}_2 = 0.3$	0.01	0.01826
$p_1 = 0.7, \bar{p}_2 = 0.5$	0.05	0.2406
$p_1 = 0.5, \bar{p}_2 = 0.5$	0.01	0.08414
$p_1 = 0.3, \bar{p}_2 = 0.7$	0.05	0.7706
$p_1 = 0.1, \bar{p}_2 = 0.9$	0.05	0.955

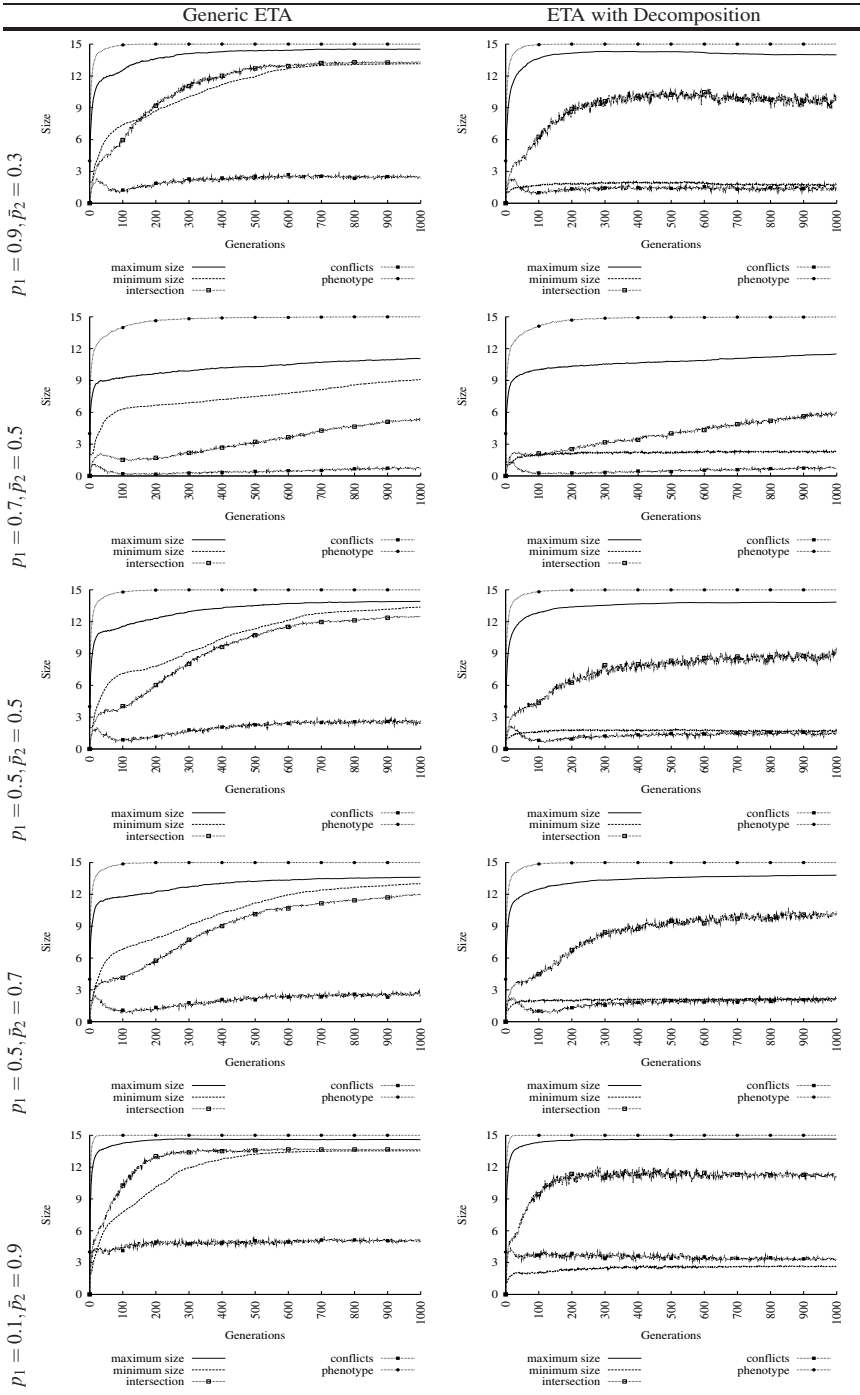


Fig. 5 Comparing size dynamics between generic ETA and ETA with Decomposition

increase is not significant with respect to the dynamics one obtains with a generic ETA without decomposition.

In Figure 5, we illustrate that the ETA with decomposition techniques has a relatively equivalent dynamics for the sizes when compared to a generic ETA. However, decomposition guarantees the survival of smaller building blocks as we can observe that the average minimum size of the ETA with decomposition remains below 3 during the evolutionary process, meaning that the smallest unit of selection is on average of length 2 while for generic ETA the minimum size increases slowly during the process and tends to converge toward the maximum size. This capacity to keep lower level building blocks during the process enforces exploration through combination of these units during the entire process and therefore, through the mechanism of conflict mediation, yields better results as it provides the system with a mechanism to escape local peaks.

5.5 *The Relevance of a Good Transition Condition*

All the simulation results discussed so far used the covering fitness to determine whether a transition occurred or not. This problem-specific approach will now be compared to a process of random transitions. The motivation for this study is twofold. First of all, problem-specific transition conditions may be difficult to determine. This issue is not new: in the context of learning classifier systems (LCS) [12, 10], the problem of finding a good mechanism to evaluate partial solutions is known as the credit assignment problem [9]. It is clear that given the problems on which one wants to apply the ETA, one will spend some time in determining how to decide on the quality of a partial solution. The difference with the assignment problem in LCS is that we do not need to address how the success of a bunch of classifiers has to be distributed over all classifiers since two partial solutions are merged as soon as they are considered to be adequate, which means they are no longer independent partial solutions.

If random transitions show good performance, a broader class of problems could be addressed with evolutionary transitions as well. Our second motivation is to compare the behavior of both setups and observe how significant the impact of a problem-specific transition condition is on the overall dynamics and success ratio. The random results form a baseline against which we can compare the results of problem-specific transition functions.

5.5.1 **Simulations Setup**

Our simulation uses the same setup as before. It uses a sound population and performs a total of 250 runs (10 seeds, 25 instances) on different types of BINCS. In the case of a random transition condition, each selected solution will perform a transition with a certain probability (the Transition Probability TP) whose distribution law is uniform. We consider three possible setups for this:

1. A low transition rate (0.2): this means that each selected solution makes its symbiotic relation permanent with a relatively low probability. At each generation, approximately 20 % of the population will perform a transition.
2. A medium transition rate (0.5): approximately 1 solution out of 2 will increase in complexity.
3. A High transition rate (0.8): only 20 % of the population does not perform a transition at each time step.

The random transition condition does not require the symbiotic relation to be successful to see an increase in the genotype complexity. Therefore, all of these rates are more likely to induce a relatively fast emergence of maximum size genotypes in the population. This means that after several generations, the evolution toward the solution only occurs through recombination of existing solutions by means of symbiosis¹. In these simulations, we wish to observe whether the observed transitions perform significantly better than random based transitions.

5.5.2 Simulation Results

In Table 5, we show the results obtained for a set of difficult BINCSPP instances. The results show that the choice of transition condition has quite a significant impact on the performance of the algorithm. It is not surprising that the generic ETA outperforms the random transition ETA. The performance of the generic ETA in comparison to the random transition ETA is, however, particularly dominant. Table 6 illustrates that this difference is significant for all test cases and setups of *TP*. Random transitions produce a fully defined genotype too quickly. As a consequence, the ETA can only evolve a solution through the recombination operation provided by

Table 5 Comparison of the success ratio between an ETA that uses a problem-specific transition condition and ETAs that perform transition on a randomly (with the Transition Probability *TP*)

		$p_1: 0.9$	0.7	0.5	0.3	0.1
		$\bar{p}_2: 0.3$	0.5	0.5	0.7	0.9
TP:						
Low	0.2	0.972	0.29	0.54	0.28	0.992
		(2335)	(8673)	(11409)	(11758)	(3627)
		[4902]	[-]	[-]	[-]	[1224]
Med.	0.5	0.464	0.064	0.136	0.036	0.804
		(6936)	(10431)	(12624)	(2011)	(3917)
		[7774]	[5725]	[11080]	[2310]	[3484]
High	0.8	0.612	0.08	0.196	0.072	0.868
		(7248)	(2546)	(6256)	(20503)	(4244)
		[8446]	[4169]	[5708]	[26279]	[4529]
		0.0	0.0	0.0	0.0	0.0
		(-)	(-)	(-)	(-)	(-)
		[-]	[-]	[-]	[-]	[-]

¹ Although similar to uniform cross-over in GAs, this recombination mechanism differs in the sense that recombination occurs prior to selection while GAs perform recombination after selection.

Table 6 Comparisons tests between the different random based transition and the generic ETA. The values in each column correspond to the p-value of the Wilcoxon-Mann-Whitney ranking test.

	0.2	0.5	0.8
$p_1 = 0.9, \bar{p}_2 = 0.3$	< 0.0001	< 0.0001	-
$p_1 = 0.7, \bar{p}_2 = 0.5$	< 0.0001	< 0.0001	-
$p_1 = 0.5, \bar{p}_2 = 0.5$	< 0.0001	< 0.0001	-
$p_1 = 0.3, \bar{p}_2 = 0.7$	< 0.0001	< 0.0001	-
$p_1 = 0.1, \bar{p}_2 = 0.9$	< 0.0001	< 0.0001	-

the symbiotic relation. It ignores one important step of compositional search: First conquer the lower level of complexity before tackling higher levels of complexity.

Figures 6, 7, 8 and 9 show the evolution of the fitness, maximum size, minimum size and conflicts of each algorithm on the difficult test cases.

The same observation can be made for each test case; High values of TP lead to the situation where the worst average fitness is obtained. This means that the high probability of transition prevents selection from performing its task well (i.e. distinguishing the good symbiotic relations from the bad ones). This observation is confirmed by the conflicts' curves in the figures. We observe that the highest value of TP yields the highest number of conflicts at the beginning of the evolutionary process (which results from premature transitions). Furthermore, this number of conflicts soon tends to become zero. This means that the evolutionary process converges

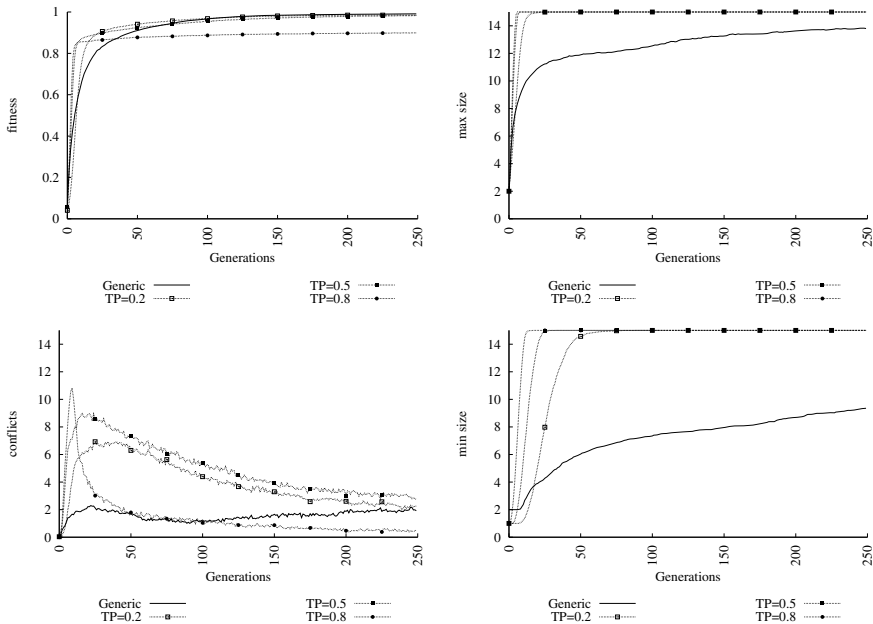


Fig. 6 Comparison between random base transition and observed transition (generic ETA). $p_1 = 0.9, \bar{p}_2 = 0.3$.

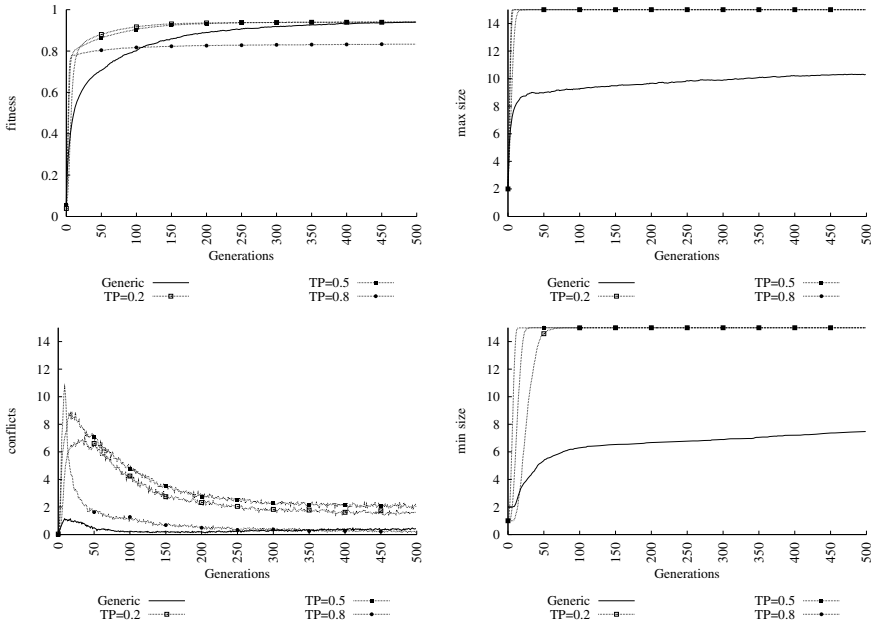


Fig. 7 Comparison between random base transition and observed transition (generic ETA). $p_1 = 0.7, \bar{p}_2 = 0.5$.

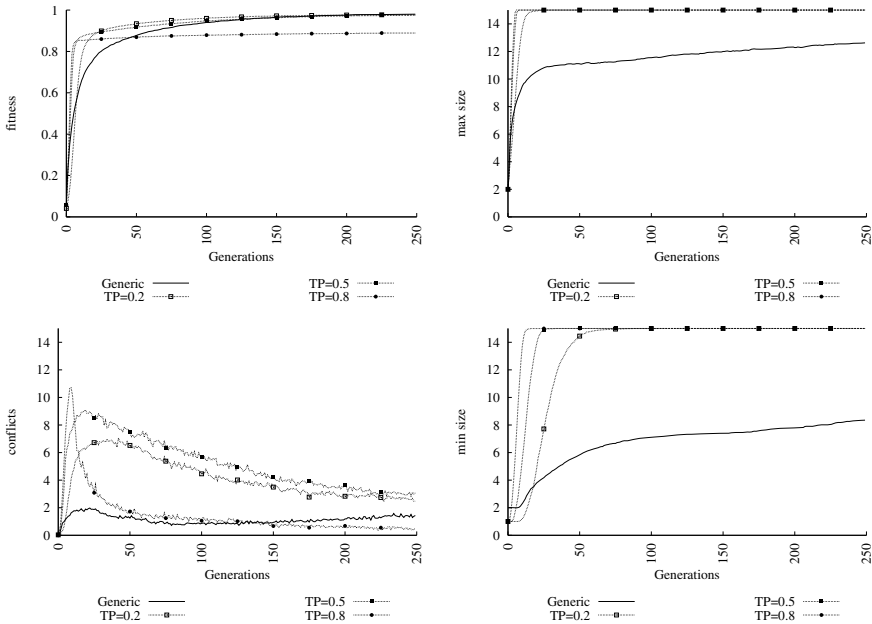


Fig. 8 Comparison between random base transition and observed transition (generic ETA). $p_1 = 0.5, \bar{p}_2 = 0.5$.

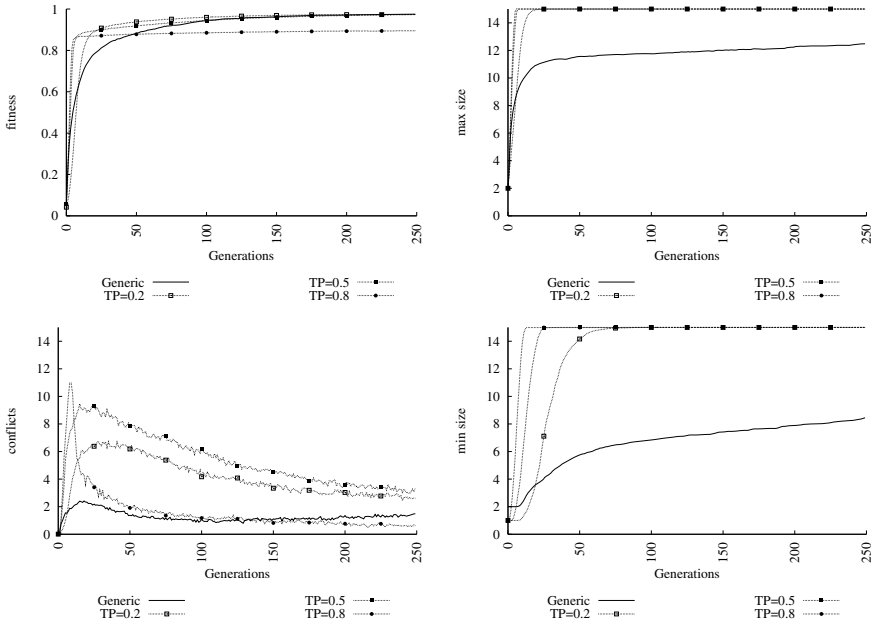


Fig. 9 Comparison between random base transition and observed transition (generic ETA). $p_1 = 0.3, \bar{p}_2 = 0.7$.

toward a sub-optimal solution and is lacking of the necessary exploration to evolve a solution to the problem. The other TP values (0.2 and 0.5) perform much better, achieving similar average fitness levels as the generic ETA. The conflicts' curves with scores above the generic case imply that the population is able to maintain sufficient diversity and therefore, some exploration is still active. However, a brief look at the evolution of the maximum and minimum sizes reveals that for all the random based transition instances, full genotypes are produced from the very beginning of the process (in less than 100 generations) and the population soon contains only fully defined genotypes (as the minimum size also converges to a fully defined genotype). Conversely, in the generic ETA, we observe that the maximum and minimum sizes remain at lower complexity levels and that different levels of complexity are present in the population (as the maximum size differs from the minimum size). This means that random-based transitions lose the efficiency inherent in the compositional approach and that these processes have to, relatively early in the evolutionary process, rely only on recombination of fully defined genotypes to evolve a solution.

6 Conclusion

In this chapter, we introduced an algorithm that mimics evolutionary transitions from biology and tackles evolutionary compositional search. We applied the ETA

on a test case which has many practical applications [20, 21]: the Binary Constraints Satisfaction Problems. We wished to evaluate how ETA performs on a challenging test case and simulated the ETA on randomly generated problem instances for various setups of the BINCSP. These instances scale from easy to difficult depending on the parameters p_1 and \bar{p}_2 [25]. We extensively analyzed the relevance of certain design decisions that were made for the ETA. From the simulation results, the following conclusions can be drawn.

First of all, ETA succeeds in compositionally building complex solutions by aggregating partial solutions through the mechanism of symbiotic relations and transitions. In the case where the system fails to evolve a fully defined genotype for the given problem instances, it is able to evolve two different partial genotypes that together through, their symbiotic relation, yield a full representation for a solution. Yet, this fully defined solution is sub-optimal. A possible reason explaining why ETA failed to evolve complete solutions for the difficult problem instances may reside in the random nature of the instances themselves. These instances are randomly generated, which means that the constraints network is, most likely, totally unstructured and that the resulting epistasis (correlation between the variables) is high. ETA, by its compositional nature, is more likely to work fine on structured instances where modularity in the problem is more apparent. This requirement for structured problem instances is confirmed when we observe the performance of ETA with other co-evolutionary approaches on BINCSP [1]. The fact that the ETA performs well on structured problems makes it promising for real-world applications, since most of them tend to be structured in some way. The study of the impact of the initial population setup on the outcome of ETA demonstrated little differences in the results [3]. This means that modeling the initial population setup is not necessarily an issue and can be kept relatively simple (for example, an initial small population of random assignments).

Concerning the impact of the introduction of decomposition techniques to the ETA, we showed that the outcome of the algorithm is positively influenced when decomposition is a rare event (only a very small fraction of the population should see the solutions being decomposed). Even though it may improve the success ratio, a statistical comparison of the best performing decomposition ETA with the generic ETA showed that, in general, there are no significant differences in terms of speed.

To analyze the quantitative relevance of a problem-specific transition condition, we re-examined all previous BINCSP results using a random (performance-independent) transition condition. We conclude that the choice to perform a transition on a symbiotic relation should be considered thoroughly and should be related to the nature of the problem to be solved as we observed that the ETA with problem-specific transition condition significantly outperformed any setup of the ETA with random condition.

Regarding the evaluation of the performance of ETA on structured problems, the ETA will be evaluated on hierarchically structured problems like the Hierarchical if-and-only-if function [27]. Moreover, we are currently investigating the performance of the ETA on combinatorial optimization problems where one does not only need

to find an assignment of values to variables but also find the optimal assignment. These new studies should provide additional understanding on the applicability of the ETA for real-world applications.

References

1. Defaweux, A., Lenaerts, T.: Evolutionary transitions in sequence complexity; a proof of concept. In: Proceedings of the Annual Machine Learning Conference of Belgium and The Netherlands, Brussels, Belgium, pp. 38–45 (2004)
2. Defaweux, A., Lenaerts, T., van Hemert, J.: 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, vol. 3630, pp. 342–352. Springer, Heidelberg (2005)
3. Defaweux, A., Lenaerts, T., van Hemert, J., Parent, J.: Complexity transitions in evolutionary algorithms: Evaluating the impact of the initial population. In: Proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, UK, pp. 2174–2181 (2005)
4. Defaweux, A., Lenaerts, T., van Hemert, J., Parent, J.: Transition models as an incremental approach for problem solving in evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference, Washington DC, USA, pp. 599–607 (2005)
5. Goldberg, D.E., Deb, K., Clark, J.H.: Genetic algorithms, noise, and the sizing of populations. *Complex Systems* 6, 333–362 (1992)
6. Goldberg, D.E., Korb, B., Deb, K.: Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems* 3, 493–530 (1989)
7. Goldberg, D.E., Korb, B., Deb, K.: Messy genetic algorithms revisited: Studies in mixed size and scale. *Complex Systems* 4, 415–444 (1990)
8. Golomb, S., Baumert, L.: Backtrack programming. *Journal of the ACM* 12, 516–524 (1965)
9. Greffentette, J.: Credit assignment in rule discovery systems based on genetic algorithms. *Machine Learning* 3, 225–245 (1988)
10. Holmes, J., Lanzi, P.L., Stolzmann, W., Wilson, S.W.: Learning classifier systems: New models, successful applications. *Information Processing Letters* 82, 23–30 (2002)
11. Hoos, H.H., Stützle, T.: *Stochastic Local Search: Foundations and Applications*. Morgan-Kaufmann/ Elsevier (2005)
12. Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.): *IWLCS 1999*. LNCS, vol. 1813, p. 243. Springer, Heidelberg (2000)
13. Lenaerts, T.: *Different Levels of Selection in Artificial Evolutionary Systems: Analysis and Simulation of Selection Dynamics*. PhD thesis, Department of Computer Science, Vrije Universiteit Brussel, Belgium (2003)
14. Maynard Smith, J., Szathmáry, E.: *The Major Transitions in Evolution*. W.H. Freeman, San Francisco (1995)
15. Michod, R.: *Darwinian Dynamics: Evolutionary transitions in Fitness and Individuality*. Princeton University Press, Princeton (1999)
16. Potter, M.: *The Design and Analysis of a Computational Model of Cooperative Coevolution*. PhD thesis, Department of Computer Science, George Mason University, USA (1997)

17. Raynal, F., Collet, P., Lutton, E., Schoenauer, M.: Individual gp: an alternative viewpoint for the resolution of complex problems. In: Banzhaf, W., et al. (eds.) *Proceeding of the Genetic and Evolutionary Computation Conference*, Orlando, Florida, USA, pp. 974–981 (1999)
18. Raynal, F., Collet, P., Lutton, E., Schoenauer, M.: Polar ifs + parisian genetic programming = efficient ifs inverse problem solving. *Genetic Programming and Evolvable Machines Journal* 1(4), 339–361 (2000)
19. Rossi, F., Dhar, V.: On the equivalence of constraint satisfaction problems. In: Aiello, L.C. (ed.) *Proceedings of the 9th European Conference on Artificial Intelligence*, Stockholm, Sweden, pp. 550–556 (1990)
20. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs (1995)
21. Salido, M.A., Garrido, A., Barták, R.: Introduction: special issue on constraint satisfaction techniques for planning and scheduling problems. *Engineering Applications of Artificial Intelligence* 21(5), 679–682 (2008)
22. Simon, H.A.: *The Sciences of the Artificial*, 3rd edn. MIT Press, Cambridge (1969)
23. Thierens, D., de Jong, E.D., Watson, R.A.: On the complexity of hierarchical problem solving. In: *Proceedings of The Genetic and Evolutionary Computation Conference*, Washington DC, USA, pp. 1201–1208 (2005)
24. Tsang, E.P.K.: *Foundations of Constraint Satisfaction*. Academic Press, London (1993)
25. van Hemert, J.: *Application of Evolutionary Computation to Constraints Satisfaction and Data Mining*. PhD thesis, Universiteit Leiden, Netherlands (2002)
26. van Hemert, J.: *RandomCSP Freely* (2002), <http://freshmeat.net/projects/randomcsp/>
27. Watson, R.: *Compositional Evolution: Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis*. PhD thesis, Brandeis University, USA (2002)
28. Watson, R.A., Pollack, J.B.: Symbiotic combination as an alternative to sexual recombination in genetic algorithms. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) *PPSN 2000*. LNCS, vol. 1917, pp. 425–434. Springer, Heidelberg (2000)
29. Watson, R.A., Pollack, J.B.: A computational model of symbiotic composition in evolutionary transitions. *Special Issue on Evolvability Biosystems* 69(2-3), 187–209 (2002)
30. Wiegand, P.: *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University, USA (2004)