

---

# Angels & Mortals: A New Combinatorial Optimization Algorithm\*

Francesc Comellas and Ruben Gallegos

Departament de Matemàtica Aplicada IV, Universitat Politècnica de Catalunya,  
Avda. Canal Olímpic s/n, 08860 Castelldefels, Barcelona, Catalonia, Spain.  
comellas@mat.upc.es

**Summary.** In this paper we present a new optimization algorithm based on the collective behavior of a set of agents which encode the problem to be solved. We compare its performance with a standard genetic algorithm on a classical difficult problem, the  $k$ -coloring of a graph. The results show that this new algorithm is faster and outperforms a standard genetic algorithm for a range of random graphs with different sizes and densities. Moreover, it might be easily adapted to solve other NP-complete problems providing a new efficient tool to deal with difficult combinatorial problems.

## 1 Introduction

An adaptive complex system consists of a large number of interacting units where different processes of learning, change and selection take place; these processes are often driven by information obtained from the environment. In some way, and by mechanisms which we are just starting to understand, these systems discriminate relevant details from random noise and use this information in a collective way. Thus, in an adaptive complex system the global behavior does not depend only on the individual features of its parts, but also on its structure and on the different sort of relations which can be established between them and the environment

A swarm of bees, an ant colony, companies which supply a big city (water, electricity, telephone, etc.), are all complex systems where global patterns emerge from the interaction of a large number of similar elements. Furthermore, this global behavior allows the different systems to attain certain achievements without the presence of an administrative hierarchy or a central control mechanism.

---

\* Research supported by the Ministry of Science and Technology, Spain, and the European Regional Development Fund (ERDF) under projects TIC2001-2171 and TIC2002-00155.

Very often the macroscopic behavior of these adaptive complex systems is studied by introducing equations to describe collectively the microscopic components according to continuous density distributions. However, it has turned out that the opposite is true, the consideration of the individual parts of the system in a discrete way leads to the right explanation of the macroscopic features. This is precisely the approach that Shnerb, Louzon, Bettelheim and Solomon used in a simple model to prove that while the continuum equations would predict extinction of a certain population, the microscopic approach explains the existence of localized subpopulations with collective adaptive properties that allow their survival and further development. Moreover, they show this happens when the relations among the parts occur essentially in two dimensions.

Although we believe that some of these concepts are implicit in certain optimization algorithms, we decided to translate it explicitly into a combinatorial optimization algorithm and test if the "prevalence of life" found by the authors of [11] helps to drive the algorithm towards a quasi-optimal solution of the problem to solve. We call the new method *angels & mortals*, like the game or computer simulation that they use to test the results in their paper. We give a detailed description in Section 3.

The problem which we have considered to test our implementation is the of a graph. This is known to be an NP-complete optimization problem [9] and finding an optimal solution is a computationally hard task, but there exist efficient algorithms to find quasi-optimal solutions as, for example, simulated annealing, genetic algorithms or ant colony based systems. All these general combinatorial optimization methods are used to obtain an acceptable answer in a reasonable time, see [1, 2, 6, 8].

In Section 2 we give a short introduction to the terminology used and a description of the genetic algorithm approach to the problem considered. In Section 3 we describe the motivation and general aspects of our implementation of the *angels & mortals* algorithm. In Section 4, we present the details of the implementations and the results obtained and in Section 5 we discuss briefly the interest of this new combinatorial optimization method and a possible extension of the algorithm to other problems.

## 2 Graph coloring and combinatorial optimization methods

A *proper coloring* of a graph  $G = (V, E)$  is a function from the vertices of the graph to a set  $C$  of *colors* such that any two adjacent vertices have different colors. If  $|C| = k$ , we say that  $G$  is *k-colored*. The minimum possible number of colors for which a proper coloring of  $G$  exists is called the *chromatic number*  $\chi(G)$  of  $G$ . The problem of finding the chromatic number and a proper coloring of a graph is of great interest for its widespread applications in areas such as scheduling and timetabling and particularly in frequency assignment in radio

networks [2, 3, 12]. As many other problems in graph theory, it is an NP-complete problem [9]. Efficient algorithms for this problem are known only for particular graphs, e.g. triangle-free graphs with maximum degree three [4].

When exact methods are not possible, sometimes it is sufficient to obtain an approximate solution with a fast and easy to implement method. This is the case of simulated annealing, genetic algorithms, neural networks, ant colony based systems, etc.

To implement any of these optimization methods we need a way to encode the problem which has to be solved, and a system to quantify the “goodness” of a solution. In the case of  $k$ -coloring, a possible solution may be encoded using a list such that each position is associated to a vertex of the graph and its value to a color. The cost function simply counts the number of times that an edge joins vertices with the same color.

We will compare our technique, *angels & mortals*, with a genetic algorithm because the latter is the method that, although different in concept, has more aspects in common. Both algorithms, for example, consider a set of individuals or population which evolves over time. However in our method the *physical* distance between individuals, and their relation with the environment plays a role which does not exist in a standard genetic algorithm. We would like to emphasize that, for graph coloring problems, other methods, like simulated annealing or ant colony systems, are more efficient than this genetic algorithm or the *angels & mortals* algorithm introduced here. The aim of our paper is to fully describe this new optimization technique and compare it with equivalent methods. Further tests with standard benchmarks will be needed to assess if *angels & mortals* performs better than other algorithms for some specific combinatorial optimization problems.

In a standard genetic algorithm, see [7, 10], the starting point is a collection of possible solutions generated at random, known as *population*. A suitable encoding of each solution in the population is used to compute its *fitness* through a *cost function*. At each iteration a new population, or *generation*, is obtained by *mating* the best of the old solutions with one another. To create the next generation, new solutions are formed through *selection*, *crossover* and *mutation*. In our implementation we rank the individuals according to their fitness and select the best to form a *parent pool* used to obtain a new generation. The solutions that will be considered for crossover are probabilistically selected according to the fitness values from this parent pool. Crossover creates two new *child* solutions from two solutions sampled from this pool. In this way, fitter parents have a better chance of producing children. The process is repeated until a new population with the same size as the original is generated. Children solutions are obtained by interchanging random parts of their parents (i.e. fragments of the corresponding lists). Some randomness is also introduced through the mechanism called *mutation* to ensure that the algorithms avoid getting stuck at local minima. Mutation changes selected parts of a solution (for example a value in the list is replaced

by a new one). The crossover and mutation operations are done with fixed probabilities, thus ensuring that some solutions from the current generation will be kept in the new generation.

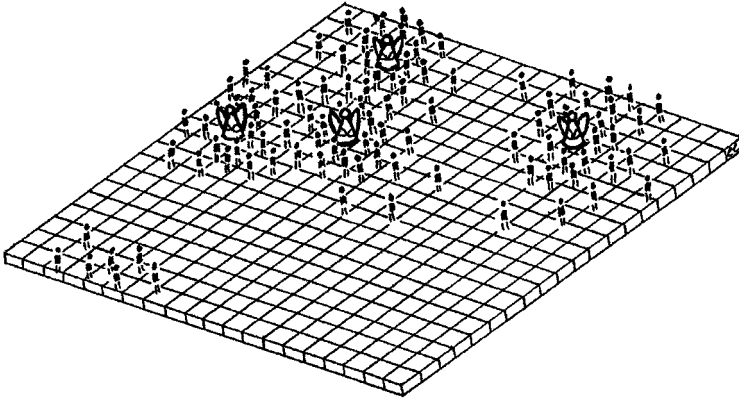
Once a new generation is created, the fitness of all solutions is evaluated and the best solution is recorded. The process is repeated until either the results stabilize or the optimal solution, when it can be identified, is reached.

Therefore the main aspects to decide in the genetic algorithm are the representation of the solutions, the cost function and the crossover and mutation operators. Important parameters are the population size, the size of the parent pool, and the probabilities of crossover and mutation.

### 3 The *angels & mortals* algorithm

The aim of the algorithm is to associate possible solutions for a given problem to the individuals of an artificial world that evolves according to certain rules. Like in other algorithms a fitness is associated to the quality of the solution but in this algorithm the fitness is also tied to the lifespan of the evolving individual.

We have chosen the *angels & mortals* model of Shnerb, Louzon, Bettelheim and Solomon [11] as their research shows that it is a very simple that explains the behavior of complex ecological systems. They distribute randomly a certain number of *mortals* over a torus grid. These individual have a given lifespan and at each clock tick it is reduced by one unit. On the other hand, there are also a few eternal agents, or *angels*, scattered over the same board. The mortals and angels move randomly from cell to cell of the grid. There is one simple rule: when a mortal meets an angel the mortal is cloned to a near place. They wondered what will be the evolution of this world, and the interesting result is that this depends on the way of looking at it. Given average population densities of angels and mortals, it is easy to write an equation that predicts the average death and birth rates. Under some conditions this continuum approach predicts the extinction of the mortals. However a computer exact simulation at the individual level leads to a totally different outcome. Although there is an initial reduction of the mortals population, later this recovers. This contradiction between the continuum and discrete approaches is explained by the adaptive behavior of the mortals. When some of them meet an angel new births take place in its neighborhood and the overall mortal population increases at these sites. The result is clouds of mortals moving around following their angels. Clouds are unstable as they grow, split up and join again, but because of this apparently adaptive behavior, the population of mortals survive. However the individual mortals have no explicit rules other than they duplicate in front of an angel. In that aspect they differ from standard adaptive agents with complex rules embedded in them. Here a nonadaptive individual produces an adaptive global world.



**Fig. 1.** A representation of the world with angels and mortals. After some generations, mortals form clouds that follow the angels.

In our implementation we construct first a toroidal world of  $n \times m$  cells where each cell may be empty or contain an angel or a mortal. A certain number of angels and a larger number of mortals are assigned at random to different cells of this world. We then read the adjacencies of the graph to be colored and the number of colors that the algorithm will consider. The next step consists of generating as many random solutions (lists of colors, such that each list position is associated to a vertex of the graph) as mortals are in the world. The fitness of each solution is calculated (number of edges which join vertices with different colors) and, according to this fitness a is assigned to the corresponding mortal. A better fitness translates into a longer lifespan. A generation consists of moving, if possible, to one of eight near cells each of the angels and mortals, clone a mortal if it happens to be near an angel, decrease one unit the life counter of all mortals and eliminate those mortals that reach zero life and finally mutate all mortals. The reaper also eliminates those mortals that have 40 percent or more of the associated edges joining vertices with the same color. The process is repeated until a solution to the problem is found or a predefined maximum number of generations has elapsed. We discuss now briefly the main operators of this algorithm.

*Mutation.* It is exactly the same operator as in a GA, but while the probability of mutation for a GA is small, in the angels & mortals algorithm there is always mutation. Mutation considers a list position at random and replaces the current color by the best possible color, if this exists. If that is not possible, then no change is performed. It would seem that this sort of mutation, the only source of change in the system, would drive the fitness to a local minimum, but the presence of angels and the cloning process produces enough diversity to avoid minima as different clones of the same mortal can evolve

```

Angels & Mortals Algorithm():
Begin
  Set MaxGenerations;
  Initialize an  $n \times m$  world with  $A$  angels and  $M$  mortals.;
  Associate a random solution to each mortal;
  Assign the lifespan of each mortal accordingly to their solution fitness;
  Repeat Until ( currentGeneration < MaxGenerations ) Do
    Look for the best mortal;
    If (this mortal solution is the global optimum) Then
      Report solution and exit algorithm;
    endif
    Decrease life counters;
    Reap mortals;
    Move randomly all angels and mortals to a neighbor cell;
    Clone mortals near angels ;
    Mutate mortals;
    Recalculate fitnesses and assign new lifespans;
    Increment currentGeneration;
  endDo
End.

```

Fig. 2. A basic version of the Angels & Mortals Algorithm.

very differently and explore other paths of the state space. In the last section we will discuss other variations of the mutation operator.

*Cloning.* When a mortal encounters an angel, the cloning process produces a new mortal and puts it in the first available cell starting by the cell just up and checking clockwise the eight cells around it. This simple rule helps to the survival of the best individuals. Because good individuals have a longer life they have also more chances to meet an angel. We have also tested a variation in which the encounter of a mortal with an angel does produces a clone but increases its lifespan.

*Reaper.* After moving a mortal, cloning it -if he is near an angel- and mutating it, its life is decreased by one unit. When it reaches 0, the individual is removed from the world. We see that the association of fitness to lifespan is crucial for the right convergence of the algorithm and is also directly related with the world size. If life is too long the world could become overcrowded. Set too short a life and the mortal population will disappear. We have introduced also a reaper mechanism that kills an individual which have more than 40 percent of its edges joining vertices with the same color. The reaper helps to improve the performance of the algorithm as there are less individuals to compute and it creates empty cells for clones.

In the next section we will give the values considered for our implementation

## 4 Results

For our study, a large number of instances of the problem to be solved have been generated to test and compare the performances of a standard genetic algorithm and the new *angels & mortals* algorithm. We use random graphs of orders ranging from 30 to 200 vertices and densities of 5%, 15% and 20% (the density of a graph is the ratio between the number of edges that actually has the graph and the maximum number that may contain). For each case 20 simulation runs were performed.

All simulations were programmed in C (less than 500 lines) and executed on a PC (AMD K7 Athlon at 1411 MHz) under Windows Me.

Possible solutions have been coded as lists where each position represents a vertex of the graph and has values 0 to  $k - 1$  according to the color assigned to it. The cost function calculates the number of edges that do not allow in the associated graph a proper coloring and subtracts this number from the size (total number of edges) of the graph.

The main parameters of the genetic algorithms are: *Population*= 200, *parent pool size*= 150, *crossing probability*= 0.9 and *mutation probability*= 0.001.

We have tested two different versions of the *angels & mortals* algorithm.

$A \& M_1$  is a implementation of the Shnerb, Louzon, Bettelheim and Solomon concept, see [11]. It starts with a toroidal  $20 \times 20$  world with 25 angels and 5 mortals. The maximum number of mortals allowed is 200. When this value is reached, the cloning process does not act. The lifespan of a mortal is  $100 \left\lfloor \frac{\text{fitness}}{\text{total number of edges}} \right\rfloor$ . After a mutation the life is modified according to the change in the fitness.

$A \& M_2$  is a simple variation of the algorithm such that initially the population of mortals is set to 200 and there is no cloning. When a mortal encounters an angel, its lifespan is extended by a fixed amount (6 units).

Table 1 shows the results corresponding to graphs of orders 30,50,70 100 and 200 with edge densities of 5%, 15% and 25.

We have performed a wide range of experiments testing different world sizes, mutation operators, parameter values etc. In most cases the algorithm converges similarly or better than the genetic algorithm. In that sense, Table 1 does not represent runs corresponding to the best possible performance of the algorithms but just a complete set of experiments. Furthermore, and as it has been reported in Section 2, there are other methods more suitable for graph coloring problems. We have implemented, for example, a simple version of a algorithm (around 100 lines of code in C language) which finds solutions of a similar quality four to five times faster. In this paper, however, we decided to compare *angels & mortals* with a technique computationally equivalent. More tests considering other combinatorial optimization problems will help to determine the exact role that our new algorithm can play.

**Table 1.**  $k$ -coloring problem for graphs of orders 30,50,70 100 and 200 with edge densities of 5%, 15% and 25 %. Comparative values between *angels & mortals* algorithms and a standard genetic algorithm. (*A&M<sub>1</sub>*: *angels & mortals* with cloning; *A&M<sub>2</sub>*: *angels & mortals* with life extension; GA: genetic algorithm, see text for details).

Vert	Edges	Colors			Time			# Suc.		
		A&M <sub>1</sub>	A&M <sub>2</sub>	GA	A&M <sub>1</sub>	A&M <sub>2</sub>	GA	A&M <sub>1</sub>	A&M <sub>2</sub>	GA
30	22	2	2	3	0.01	0.06	0.01	18	19	20
50	61	3	3	3	0.09	0.03	0.08	20	14	11
70	120	3	3	4	0.19	0.13	0.14	3	10	16
100	247	4	4	4	1.35	0.25	1.14	2	13	1
200	995	9	7	8	2.95	0.80	2.15	3	5	1

Vert	Edges	Colors			Time			# Suc.		
		A&M <sub>1</sub>	A&M <sub>2</sub>	GA	A&M <sub>1</sub>	A&M <sub>2</sub>	GA	A&M <sub>1</sub>	A&M <sub>2</sub>	GA
30	65	3	3	4	0.10	0.04	0.02	4	13	19
50	184	5	5	5	0.44	0.11	1.14	15	20	8
70	362	6	6	7	0.47	0.25	0.30	9	17	7
100	742	8	8	9	1.89	0.38	0.48	2	12	1
200	2985	17	15	16	10.32	1.05	3.86	3	11	8

Vert	Edges	Colors			Time			# Suc.		
		A&M <sub>1</sub>	A&M <sub>2</sub>	GA	A&M <sub>1</sub>	A&M <sub>2</sub>	GA	A&M <sub>1</sub>	A&M <sub>2</sub>	GA
30	108	4	4	5	0.10	0.07	0.06	6	14	11
50	306	6	6	7	0.41	0.20	0.24	5	12	2
70	603	8	8	9	1.10	0.50	0.44	2	4	6
100	1237	12	11	12	1.71	0.45	0.94	7	10	9
200	4975	24	22	23	11.58	1.38	6.46	4	14	6

## 5 Discussion.

The model presented here opens a new range of optimization algorithms based on artificial life and other adaptive systems. The main idea is to associate possible solutions of a problem to individuals of a complex system and let it evolve.

The algorithm designed is very robust. We have tested worlds with sizes from  $15 \times 15$  to  $40 \times 40$ , number of angels from 5 to 30, maximum number generations from 10.000 to 40.000, and a mutation mechanism accepting a decrease in the fitness (similarly to simulated annealing). In all cases the algorithm finds a solution of quality similar to a standard genetic algorithm.



We have restricted ourselves to a very simple model for these first tests and the results presented in the former section are encouraging.

Finally, the *angels and mortals* might be very easily adapted to solve other problems by considering the corresponding cost function and mutation mechanism. As a general conclusion, the results show that the algorithm performs better than a standard genetic algorithm even without fine tuning it. It is simple and easy to implement and can suggest other algorithms based on artificial life systems for which a problem is coded into an individual and the fitness of the corresponding solution is associated to a relevant characteristic of this individual, e.g. the lifespan. Moreover, the *angels and mortals* algorithm presented here, besides finding better solutions, runs much faster than a standard genetic algorithm and it might be also implemented on a parallel computer, thus improving further its performance.

## References

1. Aarts E, Lenstra JK (1997) Local Search in Combinatorial Optimization. Wiley, Chichester New York
2. Abril J, Comellas F, Cortés A, Ozón J, Vaquer M (2000) IEEE Trans Veh Tech 49:1558–1565
3. Allen SM, Smith DH, Hurley S, (1999) Discrete Math 197/198:41-52
4. Bondy JA, Locke SC (1986) J Graph Theory 10:477–504
5. Costa D, Hertz A (1997) J Oper Res Soc 48:295–305
6. Duque-Antón M, Kunz D, Rüber B (1993) IEEE Trans Veh Technol 42:14–21
7. Goldberg DE (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading
8. Galinier P, Hao JK (1999) J Comb Optim 3:379–397
9. Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, New York
10. Holland JH (1992) Scientific American 267:44–50
11. Shnerb NM, Louzoun Y, Bettelheim E, Solomon S (2000) Proc Natl Acad Sci USA 97:10322-10324
12. Smith DH, Hurley S (1997) Discrete Math 167/168:571–582