

# On the ability of Swarms to compute the 3-coloring of graphs

Blanca Cases, Carmen Hernandez, Manuel Graña, Alicia D'Anjou<sup>1</sup>

<sup>1</sup>Grupo de Inteligencia Computacional  
www.ehu.es/ccwintco  
UPV/EHU

## Abstract

Swarm Intelligent Systems are computational models of the spatial evolution of populations explaining it as a global behavior emerging from locally controlled movements, which are guided by decisions taken on the basis of local information. The increase of agents' cognitive capabilities, endowing them with memory and with the ability of selecting the rules of movement depending on an internal state allows the application of Self-Organizing Particle Systems (SOPS) to heuristic problem solving. Our focus in this work is on the complex emergent behavior arising when endowing the individuals with another elementary cognitive ability: the perception of the affinity of another individual. Each individual agent perceives other individual agents as friends or enemies. The first class is attractive while the second is repulsive. Metaphorically, the first class is associated with amity, security and comfort while the second is associated with danger, enemies and things to avoid. This local individual perception produces the emergence of teams and classes at a global level. This behavior produces a spatial distribution that can be interpreted under the appropriate metaphor as solving a particular computational problem. Applying this metaphor, we have found empirically that Self-Organizing Particle Systems can be designed to perform the task of 3-coloring graphs with the same precision as the Brélaz coloring heuristic, which is the best greedy heuristic known for this purpose.

## Introduction

Emergent cooperation in biological systems is a central concept in Artificial Life from its very beginnings, researching for the ways in which a whole population of simple organisms is able to collectively perform a task (Nitschke, 2005). The natural social phenomena that inspired artificial life systems are swarms (Eberhart et al., 2001): flocking birds, fish schools, ant colonies, hives, or the pursuit and evasion behavior of predators and preys. The principal idea that underlies these works is the design of biologically inspired models, analyzing the emergence of collective behavior in terms of the local decision rules that govern the action of agents.

Emergent cooperation interest is not restricted to the domain of Artificial Life. Distributed Artificial Intelligence

(Russell and Norvig, 1995) has gone in the direction of developing multi-agent systems able to solve problems by its collective behavior.

Swarm Intelligence (also called Self-Organizing Particle Systems (SOPS)) elements are agents geographically situated in a virtual environment. The emergent behaviors of interest for researchers are the ones showing collective navigation abilities or spontaneous clustering. These interests remain invariant from the first works of Reynolds (Reynolds, 1987, 1999) in computer graphics animation or the early applications to the navigation of teams of robots (G. et al., 2006; Lerman et al., 2001). On the other hand, Distributed Artificial intelligence is more focused in the local mechanisms of logical reasoning and conflict resolution over abstract spaces for knowledge representation.

Behind the approaches of Distributed Artificial Intelligence and Artificial Life to the design and simulation of (biologically inspired) intelligent social systems, Theoretical Computer Science has developed mathematical tools for the complexity analysis of collective emergent behavior. This field of Grammar Systems (Csuhaaj-Varju et al., 1994) deals with a mathematical theory of agent cooperation arising from communication protocols modeled as grammars. Grammar Colonies (Kelemen and Kelemenov, 1992; Kelemenová and Csuhaaj-Varjú, 1994) is a development in the framework of Grammar Systems closely related with Swarm Intelligence. It has been proved that a society of individuals equipped of a grammar generating finite languages are able to generate context dependent languages as if they possessed a collective "mind".

Recent research trends in Swarm Intelligence go towards the convergence of Artificial Life and Artificial Intelligence applying Self-Organizing Particle Systems to Problem Solving, by means of a mechanism that is basically the same used in Grammar Colonies: endowing each agent with a finite state machine that governs its inner flight rules depending on the current state. Adding a short term memory of visited positions is enough to design a system of two competing teams that collect minerals from some deposits transporting them to their respective homes (Rodríguez and Reg-

gia, 2004). This approach is called “designing for computing” in the self-design individual swarm-like agents whose problem-solving collective capabilities are proportional to the size of the population.

The research question guiding this work is the following one: what are the minimal cognitive capabilities that allow the emergent behavior of Swarms to solve NP-complete problems, such as the classical ones dealt with by classical and heuristic Artificial Intelligence algorithms, without mediating an explicit knowledge representation. We show in this paper that the simple distinction between friends (we) and enemies (them) in a population of boids is enough to produce an emergent behavior that can be interpreted as solving the problem of graph coloring. And they do it with a performance comparable to that of “traditional” algorithms.

We use the swarm metaphor to model the graph coloring problem as follows: agents correspond one-to-one with the nodes of the proposed graph. The graph topology defines the agent affinities as follows: the agents whose nodes are directly connected are “enemies”, agents whose nodes are at graph distance<sup>1</sup> 2 are “friends”. Agents are attracted to friends while try to fly from or to avoid enemies. The colors for the graph coloring correspond to specific attraction spatial regions. All agents are attracted to stay in these regions. Figure 2 shows the virtual space where the boids are moving around. The graph coloring solution is given by the distribution of the agents over the color attraction regions. When all the agents are placed in one of the color attraction space regions, the system configuration can be interpreted as defining a a complete coloration of the graph. When some agents are outside these regions, the system configuration corresponds to a partial solution to the coloration problem. To shake the system out from local optimal configurations corresponding to partial solutions, the agents are endowed of an aggressive instinct that allows them to overcome fear or repulsion to the enemies and to try to displace them from the privileged space regions.

We found that SOPS perform the task of 3-coloring graphs with comparable and sometimes better precision than the Brélaz coloring heuristic (Weisstein, 2008), which is the best greedy heuristic known for this purpose.

In the following sections we will first introduce Reynolds model. Next section describes in detail our metaphor to model the solution of the NP-complete graph 3-coloring problem through the swarm behavior. Next section summarizes some analytical results of the approach, trying to shed some light on the problem of determining the minimal cognitive capabilities that may have the agents to solve the problem of 3-coloring. Then we give some computational experiment results over a sample of hard colorable graphs. We end up with some discussion, conclusion and venues for further research.

<sup>1</sup>Graph distance means the length of the shortest path between two nodes. Unconnected nodes have infinite graph distance.

## Description of Reynold’s model

The idea of emulating of the movements and behaviours of societies of living beings from simple local rules that steer the individuals, giving rise to more complex global behaviours is a growing field of research, with applications in quite different domains. (Reynolds, 1987, 1999) was one of the pioneers in the simulation of the flight of flocks of birds.

According with Reynolds, each individual exhibits a very simple behaviour that is specified by a few simple rules that guide them to get along with the collective motion of the flock. The global behavior of the flock emerges from these individual decisions. We will stick to the birds metaphor, so that in the following, we call boids to the agents that compose a flock.

Each boid is aware of an spatial region around it, its neighbourhood. Given a set of  $n$  boids, the steering rules for  $i$ -th boid  $b_i$ , at time instant  $t + 1$  are defined as a function of the position  $p_j$  and the velocity  $v_j$  of the neighbouring boids at the previous instant  $t$ . The set of boids dwelling inside the neighborhood of the  $i$ -th boid is denoted:

$$\partial_i = \partial(b_i) = \{b_j | dist(p_i, p_j) < \theta\}$$

where  $dist$  is the euclidean distance. Let  $|\partial_i|$  denote the number of boids in the neighbourhood.

The steering basic rules, used in our model, are the classical of Reynolds model: alignment, separation, and cohesion. Combining these rules, the flocking birds are able to flight co-ordinately avoiding collisions. The flocking rules for the boid  $b_i$  are formalized as follows:

- **Separation:** steer to avoid crowding local flockmates.

$$v_s = - \sum_{b_j \in \partial_i} (p_j - p_i)$$

- **Cohesion:** steer to move toward the average position  $c_i$  of local flockmates

$$v_c = c_i - p_i \text{ where } c_i = \frac{1}{|\partial_i|} \sum_{b_j \in \partial_i} p_j$$

- **Alignment:** steer in the direction of the average heading of local flockmates.

$$v_a = \frac{1}{|\partial_i|} \sum_{b_j \in \partial_i} v_j - v_i$$

Together with the elementary steering rules, the seek and flee rules, and the rules that define the behaviors of attraction towards friends and evasion from enemies.

- **Seek and flee:** seek attempts to steer a vehicle so that it moves toward a static goal. Here  $\|p\|$  denotes the norm of

a vector  $p$ , and  $f_{\text{maxvelocity}}$  is a non-negative parameter that limits the norm (the length in the Euclidean distance) of vector  $v_{\text{seek}}$ .

$$v_{\text{seek}} = v_{\text{goal}} - v_i \text{ where}$$

$$v_{\text{goal}} = \frac{p_i - p_0}{\|p_i - p_0\|} \times f_{\text{maxvelocity}}$$

Flee velocity is defined simply as the opposite of seek,

$$v_{\text{flee}} = -v_{\text{seek}}$$

- **Pursue and evasion.** These rules are a generalization of the seek and flee rules with the only difference of that the goal is non-static, and the boid moves toward or escapes from some of the flock mates.

### Self-organizing particle systems for 3-coloring of graphs

Reynolds model is the basis of Self-Organizing Particle Systems (SOPS) for problem solving, introduced by (Rodríguez and Reggia, 2004). Following the authors, SOPS self-organization refers to the fact of that global behaviour emerges from the concurrent local interactions between particles in such a way that the whole population acts like an organism. Their contribution consists in incrementing the cognitive capabilities of agents by a Finite State Machine (FSM) controlling the boid flight rules. In this way, the system improves past SOPS, in which the boids do not have a significant intelligence. This is an interesting approach in that it represents an overture to Distributed Artificial Intelligence from the perspective of Artificial Life.

The paper by Rodríguez y Reggia presents a model formed of two teams of agents exploiting mineral deposits, carrying minerals to their respective homes. The agents have the following types of knowledge:

- Geographical: inherited from the Reynolds's model.
- Social: being able to recognize the neighbours as belonging to a category: friend or foe.
- Internal: the set of rules that control the movement of the agent, which are represented in a FSM, and a stack of visited positions.

From the point of view of Theoretical Computer Science, each individual has internally the elements that define a pushdown automata, which are able to generate flight trajectories in the virtual space. This enhancement significantly increases the power of SOPS to solve classical NP-complete problems. We focus on the 3-coloring of graphs.

Let  $G = (V, E)$  be a graph (Borodin et al., 2005), composed of a set of vertices  $V$  and a set of non-oriented edges  $E \subseteq V^2$ . A  $k$ -coloring of  $G$  is a function that maps

each vertex in  $V = \{b_1, \dots, b_n\}$  to a colour in the set  $C = \{1, 2, \dots, k\}$  in a way such that two connected vertex have different colours. The problem can be formulated alternatively as minimizing the number of nodes that are badly coloured and even the number  $k$  of colours used. The chromatic number of a graph is the minimal number of colours needed for a coloration. The problem of  $k$ -coloring is NP-complete for all  $k \geq 3$ . We have selected 3-coloring as a benchmark problem because of the great difficulty that presents in spite of the simplicity of its formulation.

The representation proposed here is composed of a board of dimensions  $X_{\text{max}}, Y_{\text{max}}$  closed, as a torus, and  $k \leq 4$  goals inscribed in a regular polygon representing each goal a colour in the set  $C = \{1, 2, 3, 4\}$ . Goals are static and attract the individuals with a pseudo-gravitational force.

A graph  $G$  represents a population of flocking birds. Each node  $b_i$  is an agent (a boid) whose initial position and velocity are drawn at random from a uniform distribution defined over the board. The graph represents the social network of the population. Two nodes  $i, j \in V$  are enemies iff they are connected in the graph  $G$ , that is  $(i, j) \in E$ .

Hence, from the point of view of boid  $b_i$ , the set of boids actually inside its neighbourhood is partitioned into two subsets: the set of Enemies  $\partial_{E_i}$  and the set of friends, with an Amity relationship,  $\partial_{A_i}$ .

$$\partial_i = \partial_{E_i} \cup \partial_{A_i}$$

where  $\partial_{E_i} = \{b_j : (b_i, b_j) \in E\}$ . The Amity relationship is defined as being the "enemies of my enemies":

$$\partial_{A_i} = \{b_k : \exists j (b_i, b_j) \in E \wedge (b_j, b_k) \in E \wedge (b_i, b_k) \notin E\}$$

### Velocity parameters

The model of 3-coloring has been implemented in Matlab 7. The velocity of each boid depends on three strengths that modulate its current velocity:

- **The Neighbourhood strength:** determined by a **Radius** around the boid with default **weight**= 1.0, pushing the boid toward the friends and away from enemies.
- **The Goal strength:** The seek strength, that attracts the boids toward the goals. The agent can seek for all the goals or only for the nearest goal with a default **weight**= 1.0 in both cases.
- **The Attack strength:** if a boid can not reach a Goal or be sharing it only with friends after a number of steps it becomes "despaired" and attacks the enemies driving them from their positions:
  - **Internal attack:** displaces at random enemies occupying the same Goal it is lying in.
  - **External attack:** It is outside all of the Goals, since it has enemies inside all of them. The agent selects at random a Goal to head to and displaces an enemy from it.

A Matlab implementation of the process can be obtained from the authors and will be made public at <http://www.ehu.es/ccwintco/>. The magnitude of the velocity vector is globally limited by a parameter **Limitation of velocity** such that all the boids move with the same length step, set by default to 1.0.

**The Social velocity** We call Social velocity to the boid velocity component due to its repulsive and attractive interactions with the boids actually inside its neighbourhood. It is the sum of two components:

- **Enemity velocity:** taking as input the neighbouring enemies  $\partial_{E_i}$  we calculate a velocity

$$v_{E_i} = -w_a^{E_i} \times v_a^{E_i} + w_s^{E_i} \times v_s^{E_i} + w_e^{E_i} \times v_e^{E_i}$$

where  $w$ 's represents the strengths and  $a$  means alignment,  $s$  is separation and  $e$  evasion. This is the velocity term that corresponds to the alignment in opposite direction to enemies, separating and moving away from them.

- **Amity velocity:** taking as input the neighbouring friends  $\partial_{A_i}$  calculates a velocity

$$v_{A_i} = w_a^{A_i} \times v_a^{A_i} + w_c^{A_i} \times v_c^{A_i} + w_p^{A_i} \times v_p^{A_i}$$

where  $p$  means the behaviour of pursuing its friends. There exists a strength of alignment, cohesion or pursuing the boids that compose the amity group.

**The Goal seeking velocity** This term models the need to get a colour for the node. In some experiments we do not activate it (see application interface in figure 2). For the purpose of solving the graph 3-coloring, we restrict the model to have at most 4 goals. The coordinates  $(X, Y)$  of the goals are given as the vertices of a regular polygon. The goals have influence inside a Goal Radius, and they attract the boids with a strength of 1.0 by default. Depending of the settings selected in the interface, the Goal velocity term is defined as:

- If the velocity to the nearest goal has been selected as a velocity parameter, the goal velocity is:

$$v_{Goal} = \frac{p_i - g_0}{\|p_i - g_0\|} \times f_{limitvelocity}$$

where  $g_0$  is the position of the goal nearest to  $p_i$ .

- In the case of all the goals were selected,

$$v_{Goal} = \frac{\sum_{m=1}^4 p_i - g_m}{\left\| \sum_{m=1}^4 p_i - g_m \right\|} \times f_{limitvelocity}$$

Being  $g_m, m \leq 4$ , the positions of the goals.

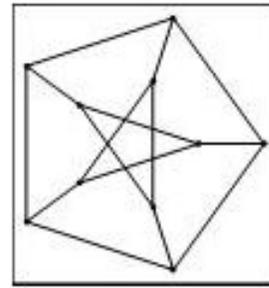


Figure 1: Petersen's Graph is 3-colorable. The system obtains a correct coloration with probability close to 1.0 for small graphs.

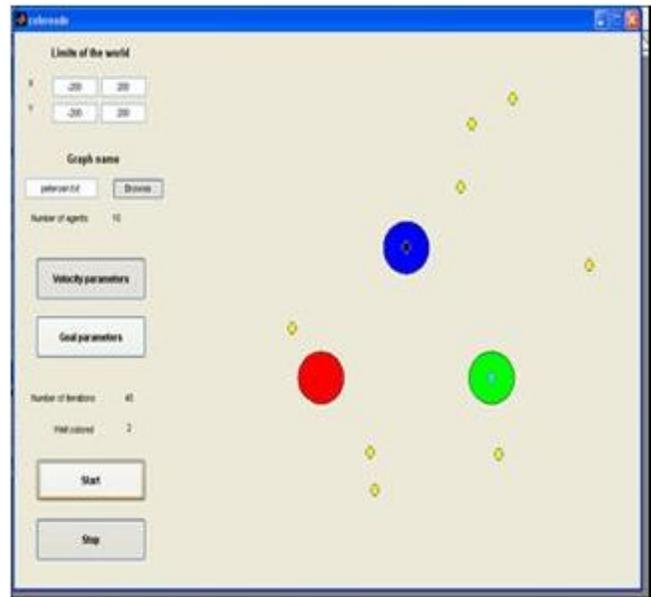


Figure 2: An snapshot of a run of 3-coloring SOPS on Petersen's graph.

## Graph Coloring

The Matlab implementation allows to set the limits of the world and to load a file encoding a graph. Figure 2 shows an instant in the 3-coloring of Petersen graph, displayed in figure 1. This graph has 10 nodes. The interface shows an animation where the boids, which are represented as yellow small circles initially distributed at random, move toward the goals producing in this way a coloration.

## Attack behaviour

The results produced by the application of the boids swarms to 3-colorable small graphs, like Petersen's graph, is successful in the almost all of the cases when:

- The boid neighbourhood radius extends to the whole virtual world.

- The boids are attracted only to the nearest goal (instead of all the goals).

Without the attack mechanism, and using the default values for the Goal Radius and its strength of attraction, the system always converges either to an optimal configuration with all the boids situated inside the goals or to a sub-optimal one, with few boids wandering around of the nearest goal. This last situation occurs whenever the graph is non 3-colorable. Once a boid reaches a goal, it remains inside forever, being the goal a sink for agents' trajectories.

To shake the system away from (local minima) configurations that do not solve the coloring problem, we propose the Attack behavior, incorporated in a rule in the following deadlock conflict configurations:

- Internal: At least two enemies are situated in the same goal.
- External: An agent is wandering outside the goals because it has enemies in all of them.

To model attack we give to the agents an internal counter of the degree of "desperation" or "dissatisfaction" of the agent in a conflictive situation. Agents in a goal have an increasing degree of satisfaction over time. Whenever an agent enters in a conflict, the satisfaction level decreases as time goes, until the counter reaches a value below a given threshold, the aggressive behavior is activated and the boid attacks.

The attack consists in selecting randomly an enemy in conflict which is less desperate than the aggressor (its level of dissatisfaction is greater than the satisfaction of the assailant agent). The boid under attack is expelled from the goal and the aggressor takes its place. We have introduced a noise term in the velocity that helps to generate mildly erratic trajectories for wandering agents.

### Modelling agents as Finite State machines

Following the pattern of the proposal by (Rodríguez and Reggia, 2004), we present our SOPS model of 3-coloring in a top-down manner.

First, the introduction of a satisfaction counter can be represented by a FSM as the one shown in fig. 3, being  $k$  the maximum value for satisfaction and 0 the minimum. We can represent the whole automaton of the figure as a single state with dash border labelled with the level  $s$  of satisfaction.

In figure 4 we give the FSM specification of a boid. We have abbreviated the states with satisfaction of level  $i$  simply as  $S > 0$ .

Initially, the agents are wandering starting at a randomly drawn position and with maximum level of satisfaction. If the agent falls within the area of influence of a goal, the Goal seek behaviour is activated and the boid tends to remain inside it. The boids only come out from the goal if they are involved in conflicts that make the satisfaction decrease.

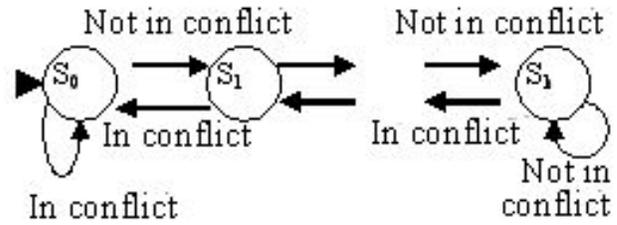


Figure 3: State  $S_0$  means dissatisfaction while states  $S_i$  where  $0 < i \leq k$  represent Satisfaction of level  $i$ .

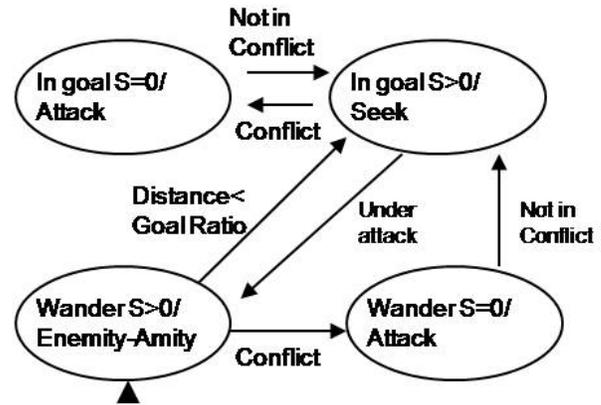


Figure 4: the FSM for a boid

If the agents suffers an attack, it passes to the wandering state, looking for a new goal.

If the agents reaches the state of desperation,  $S = 0$ , it attacks displacing another agent of the world and incrementing its satisfaction so long as conflicts disappear.

To obtain a faster convergence, we apply a cascade coloration strategy. Therefore, the execution of the program has two stages : First, the system attempts to find a 4-coloration of the graph situating 4 goals in the world. Once a coloration is obtained or after the maximum allowed time (1500 iterations) is elapsed, the second stage starts, eliminating the less populated goal. The individuals newly freed wander to seek a new goal until a 3-coloring is reached or the limit number of iterations (in this case 3500) are completed.

This procedure of cascading coloration is based on known works in reaction-diffusion particle systems (Turk, 1991) and is a way to extend the problem of 3-coloring of graphs. To find the chromatic number of a graph, i.e. the minimal number of colours that are necessary for a coloration is sufficient to start the process of colouring successively the graph

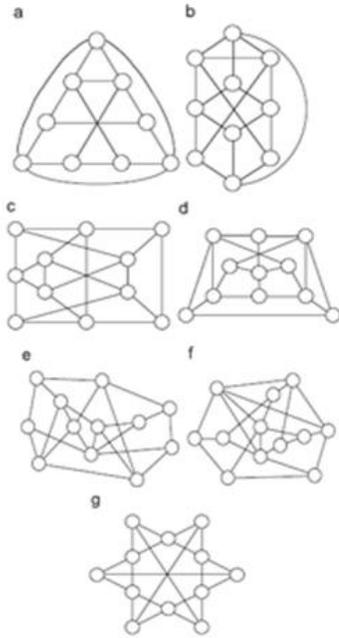


Figure 5: Several instances of hard 3-coloring graphs

with  $k, k - 1, k - 2, \dots$  colours until a minimal successful number of colours is reached.

### Benchmarking Experiments: a comparison to Brélaz heuristic

The problem of 3-coloring of graphs has a very simple formulation but it is very difficult to solve. In 1979 Steinberg (Borodin et al., 2005) formulated a conjecture: every planar graph without 4 and 5-cycles is 3-colorable. In the last years, important advances has been made in the direction of proving Steinberg's conjecture. However, the problem of 3-coloring is NP-complete and the current research efforts focus on heuristics that may give good approximations to a global optimal solution in polynomial time. The best known heuristic for graph coloring is the Brélaz algorithm (Weisstein, 2008; Galinier and Hertz, 2006), which is a greedy algorithm that proceeds as follows: it performs first the coloring of the nodes with greater degree (number of connected nodes) and more constrictions (saturation), giving to each node the first available colour.

We have made some experiments to verify that SOPS algorithm is at least as precise as Brélaz algorithm, meaning that the chromatic number given by SOPS is less or equal than Brélaz chromatic number. It is well known that Brélaz heuristic works deficiently with some hard configurations for the 3-coloring (Mizuno and Nishihara, 2008). These authors present a graph building algorithm to construct hard coloring graphs. It performs graph embedding to combine basic hard configurations, given in fig. 5, into bigger hard

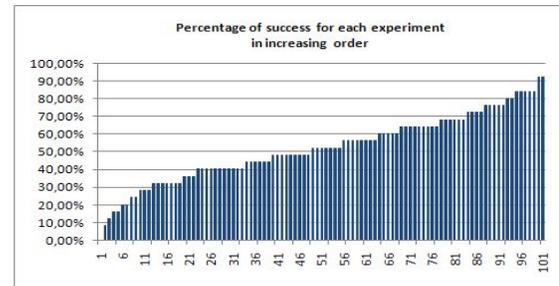


Figure 6: Success of SOPS 3-coloring

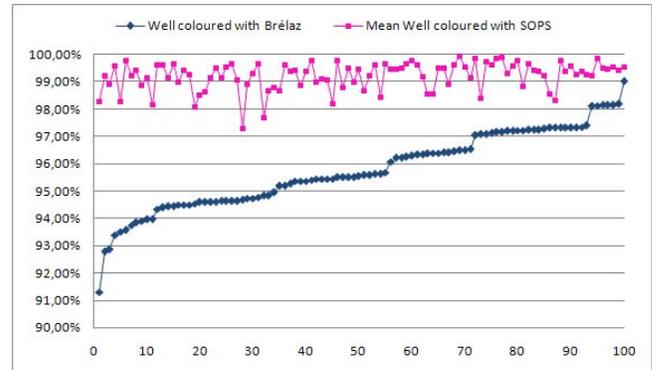


Figure 7: Percentage of well colored nodes for Brélaz versus Self-Organizing Particle Systems over the sample of benchmark hard graphs

graphs. For all of these graphs the Brélaz heuristic gives 4 as chromatic number, while all of them are 3-colorable. A sample of 100 graphs obtained from 10 random embeddings of the basic configurations were generated. For each graph, we have executed 25 runs of the SOPS algorithm registering the best configuration (we call this an experiment): Each run ends either when a 3-coloring solution is reached (success) or when 5000 iterations are completed in cascade.

The average results over all the experiments are: Mean number of nodes (boids): 110, Mean number of iterations: 3761, and Average of succeeding runs: 51%. In fig. 6 we have ordered the experimental graphs by the percentage of succeeding trials. This figure may serve as a model of the accumulative probability distribution of our algorithm obtaining a successful coloration over the sample of hard 3-colorable graphs. Note that if one execution of our algorithm obtains a 3-coloring, that constitutes a proof that the graph is 3-colorable. Note also that the Brélaz algorithm algorithm is deterministic, so that repeated trials have no sense for it.

For another look into relative performance we consider the following: if a node gets color number 4, then it is badly colored in Brélaz coloration. In SOPS, we register for each run the minimum number of individuals outside all the goals as the best configuration, and we say that this is the num-

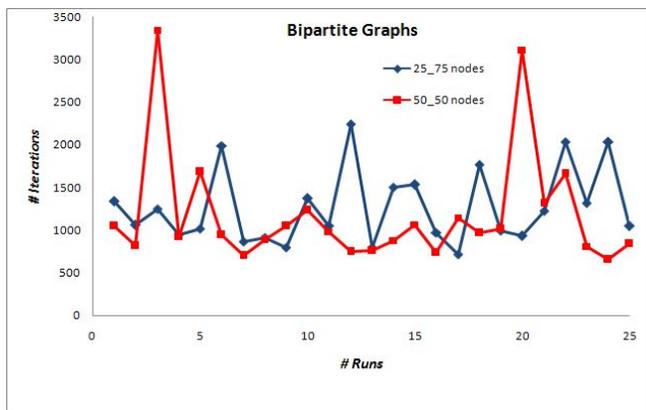


Figure 8: Number of iterations of 25 runs on the bipartite graphs

ber of bad coloured nodes for that execution. In each experiment (25 runs) the mean is taken. Figure 7 shows the percentage of well colored nodes for the whole sample. It can be appreciated that SOPS is always very close to the 100% well colored nodes, while some instances of Brélaz coloration are very poor. In order to discover if there exists a correlation between the variables, the sample has been ordered by increasing values of Brélaz algorithm. A correlation Pearson coefficient of 0.30 has been found and in consequence, correlation does not exist between the results. In average, Brélaz algorithm colorates well the 95,82% of the nodes with a standard deviation of 1.45%, while SOPS reaches a mean of the 99.17% and standard deviation 0.60%.

It is well known that Brélaz algorithm needs two colours for a bipartite graph, being particularly efficient in this case. To show that SOPS solves also correctly these problems, we have selected two complete bipartite graphs of 100 elements: the first with two classes of 50-50 nodes and the second with 25-75. In the 25 runs of each graph, the run was successful in both cases, being successful the 100% of the times. Regarding computing time measures, the mean number of iterations for graph 25-75 was 1266 and the minimum length of a successful run was 715. For graph 50-50 the average final step was 1172 being the minimum 654. Figure 8 shows the distribution of the number of iterations on the 25 runs of the SOPS algorithm for this graph.

## Discussion

We have designed and implemented a Self-Organizing Particle System that may be interpreted as solving the graph coloring problem. We addressed the problem of 3-coloration of graphs, but the cascading procedure of coloration presented before makes the extension to  $k$ -colorations be an immediate consequence. We chose the problem of 3-coloring graphs because of the important open questions around the problem: it is NP-complete and Steinberg's conjecture is giving

arise an important research nowadays (Borodin et al., 2005).

A recent biologically inspired approach to this problem has used the ant colony optimisation approach (Dowland and Thompson, 2008), but we do not know of any other attempt to solve the problem using flocking birds. Their approach that identifies an individual in a population to a whole coloration of the graph, that is a tuple  $(u_1, \dots, u_n)$  where  $u_i$  is the colour of node  $i$ , losing in this way the biological inspiration in favour of cognitive abstraction. On the other hand, our approach to the coloration of graphs is mainly geometrical, attending to the representation of the nodes of a graph as a flocking bird situated geographically. The solution to the graph coloring emerges from the whole population configuration, which means a great economy of representation, and of computational power needed to implement the approach. The geometrical approach can be a source of experimentation and inspiration to improve sequential algorithms and heuristics for 3-coloration, which is important from the point of NP-completeness.

Second, we do not proceed in the direction of creating a model of colouring graphs from an existing model. Our aim was the research of the behaviour arising from endowing the individuals in a swarm with another elementary cognitive ability: the perception of the affinity of another individual. The individual perceives another individual as belonging to We or to Them. The first class is attractive while the second is repulsive. The first class is associated with amity, security and comfort while the second is interpreted as danger, enemies and things to avoid. We found that amity-enemy dynamics allows to model the solving process for coloring graphs, and not the other way around.

The third contribution of this paper has to do with the complexity of swarms, understood as the complexity of the behaviour of the emergent super-organism with respect to the computational capabilities of individuals. This work has been made in the last years in the field of theoretical computer science (Csuhaaj-Varju et al., 1994; Kelemen and Kelemenov, 1992; Kelemenová and Csuhaaj-Varjú, 1994). We have attempted to discover the lowest computational capabilities of individuals that allows the swarm to perform a coloration of a graph. Revisiting the work of Rodriguez and Reggia (2004) may lead a strong theoretical basis for further developments in the convergence with grammar systems.

The experimental results on hard coloring graphs with known chromatic number 3, show that the proposed approach can be very effective and competitive with state of the art algorithms. The Brélaz algorithm is the common benchmark algorithm. Our approach improves on it over a sample of hard graphs.

## Acknowledgements

The MEC partially supports this work through grant DPI2006-15346-C03-03. The Basque Government has

awarded a *Grupo de Investigación Universitario* award to the Computational Intelligence Group.

## References

- Borodin, O. V., Glebov, A. N., Raspaud, A., and Salavatipour, M. R. (2005). Planar graphs without cycles of length from 4 to 7 are 3-colorable. *J. Comb. Theory Ser. B*, 93:303–311.
- Csuhaj-Varju, E., Kelemen, J., Paun, G., and Dassow, J., editors (1994). *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*. Gordon and Breach Science Publishers, Inc., Newark, NJ, USA.
- Dowland, K. A. and Thompson, J. M. (2008). An improved ant colony optimisation heuristic for graph colouring. *Discrete Applied Mathematics*, 156:313 – 324.
- Eberhart, R., Shi, Y., and Kennedy, J. (2001). *Swarm Intelligence*. Morgan Kaufmann, 1 edition.
- G., B., D., P., and S., N. (2006). Distributed coordination of simulated robots based on self-organisation. *Artificial Life*, 12, n. 3:289–311.
- Galinier, P. and Hertz, A. (2006). A survey of local search methods for graph coloring. *Comput. Oper. Res.*, 33:2547–2562.
- Kelemen, J. and Kelemenov, A. (1992). A grammar-theoretic treatment of multiagent systems. *Cybernetics and Systems. An International Journal*, 23(6):621–633.
- Kelemenová, A. and Csuhaj-Varjú, E. (1994). Languages of colonies. *Theoretical Computer Science*, 134(1):119–130.
- Lerman, K., Galstyan, A., Martinoli, A., and Ijspeert, A. (2001). A macroscopic analytical model of collaboration in distributed robotic systems. *Artificial Life*, 7:375–393.
- Mizuno, K. and Nishihara, S. (2008). Constructive generation of very hard 3-colorability instances. *Discrete Applied Mathematics*, 156(2):218–229.
- Nitschke, G. (2005). Emergence of cooperation: State of the art. *Artificial Life*, 11(3):367–396.
- Reynolds, C. (1999). *Steering Behaviors for Autonomous Characters*. <http://http://www.red3d.com/cwr/papers/1999/gdc99steer.html>.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21:25–34.
- Rodríguez, A. and Reggia, J. A. (2004). Extending self-organizing particle systems to problem solving. *Artificial Life*, 10(4):379–395.
- Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, New Jersey, USA.
- Turk, G. (1991). Generating textures on arbitrary surfaces using reaction-diffusion. *SIGGRAPH Computers Graphics*, 25(4):289–298.
- Weisstein, E. (2008). Brelaz’s heuristic algorithm.