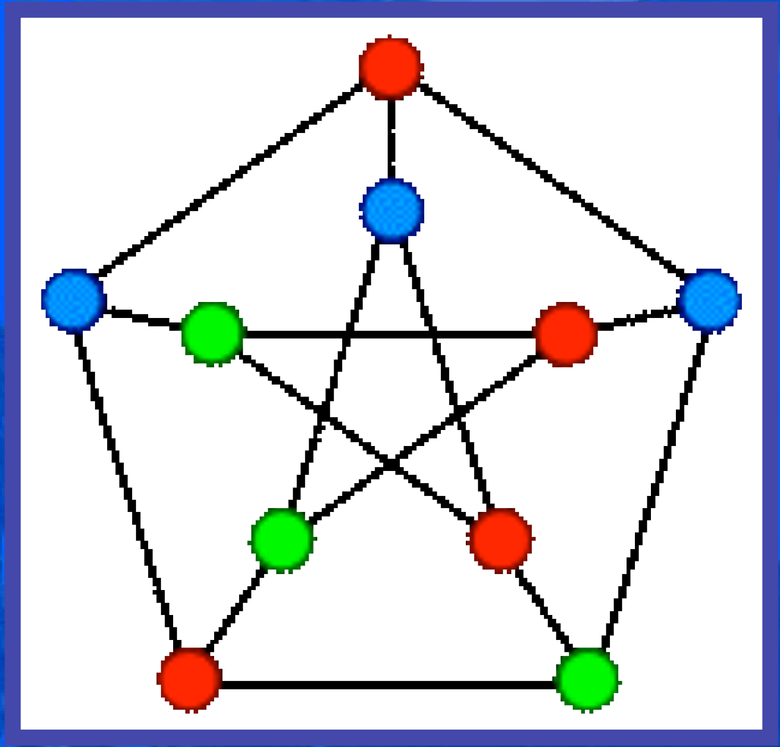
Grupo de Inteligencia Computacional UPV/EHU
http://www.ehu.es/ccwintco

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

Blanca Cases,
Carmen Hernandez,
Manuel Graña,
Alicia D'Anjou

Graph coloring problem

- **Application of swarm intelligence to the k-coloring of graphs.**
- **Minimize the number of neighboring vertices with the same color.**
- **NP complete for k≥3.**
- **Important problems in graph theory.**



Approaches to graph coloring

- Theoretical computer science: **P=NP?**
- Artificial intelligence: **heuristic search.**
- Computational Intelligence and Artificial Life:
  - Searches for non deterministic algorithms that solve the problem efficiently in polynomial times.
  - Complex systems: Neural networks, swarm intelligence, cellular automata…
  - Biological inspiration increments expressiveness of programming languages.
  - Orientation to graph drawing techniques

Self-Organizing Particle Systems (SOPS)

- Computational models of the navigation of swarms.
- As a behavior emerging from locally controlled movements.
- Based on decisions taken on local information.
- Approaches to swarm intelligence:
  - Steering behaviors: Reynolds .
  - Stirmergy: Ant colony optimization solves the travel salesman problem as an emergent behavior.
  - Optimization of abstract functions:
    - Self-organization: Particle swarm optimization (PSO).
    - Self-organisation+Evolution=Stochastic diffusion search

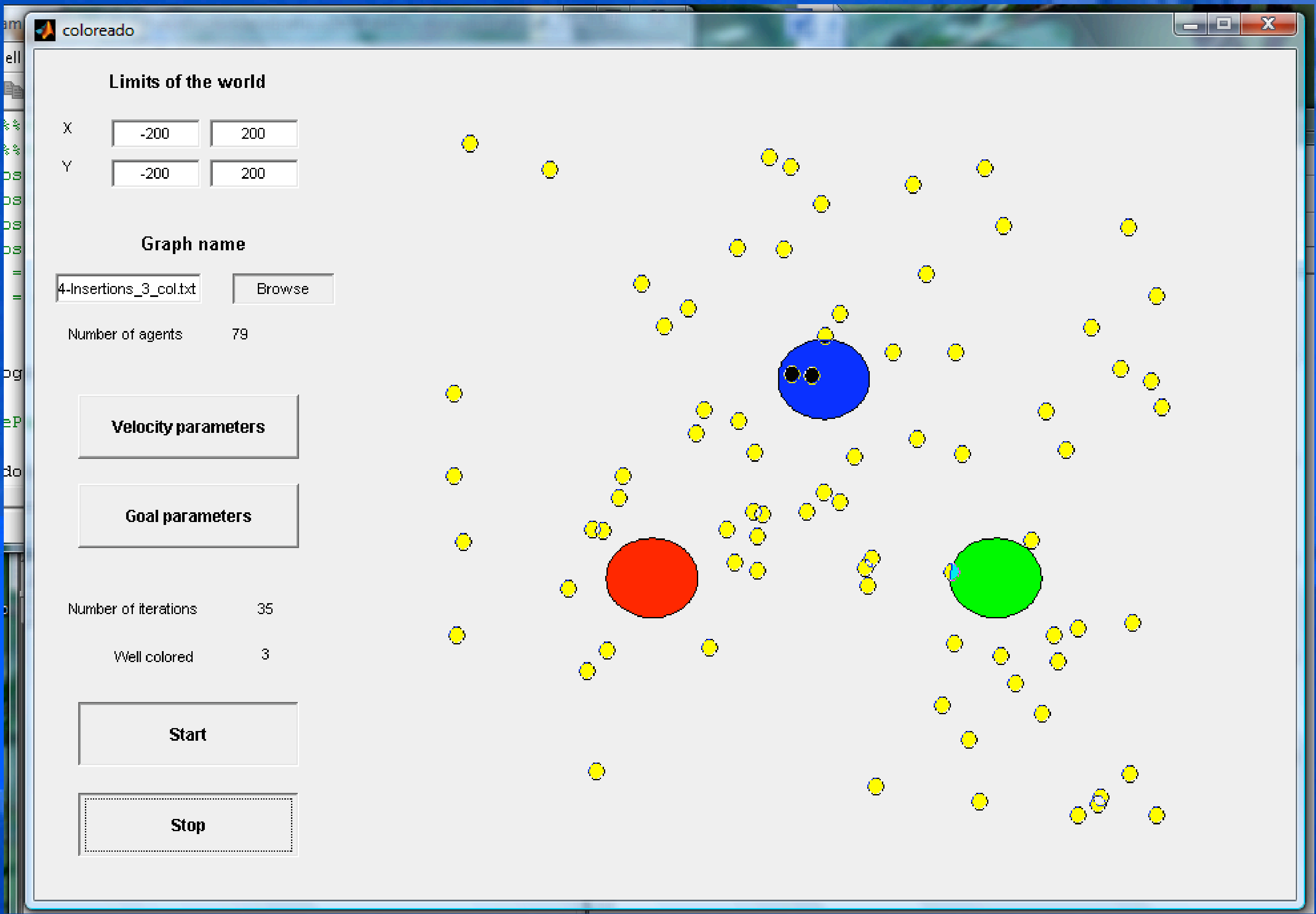
Looking for biological inspiration

- **General methods of optimization lose biological inspiration when representing some combinatorial problems.**
- **Steering behaviors are the simplest approach to swarm intelligence.**
  - Perception: position and velocity.
  - Action: change position.

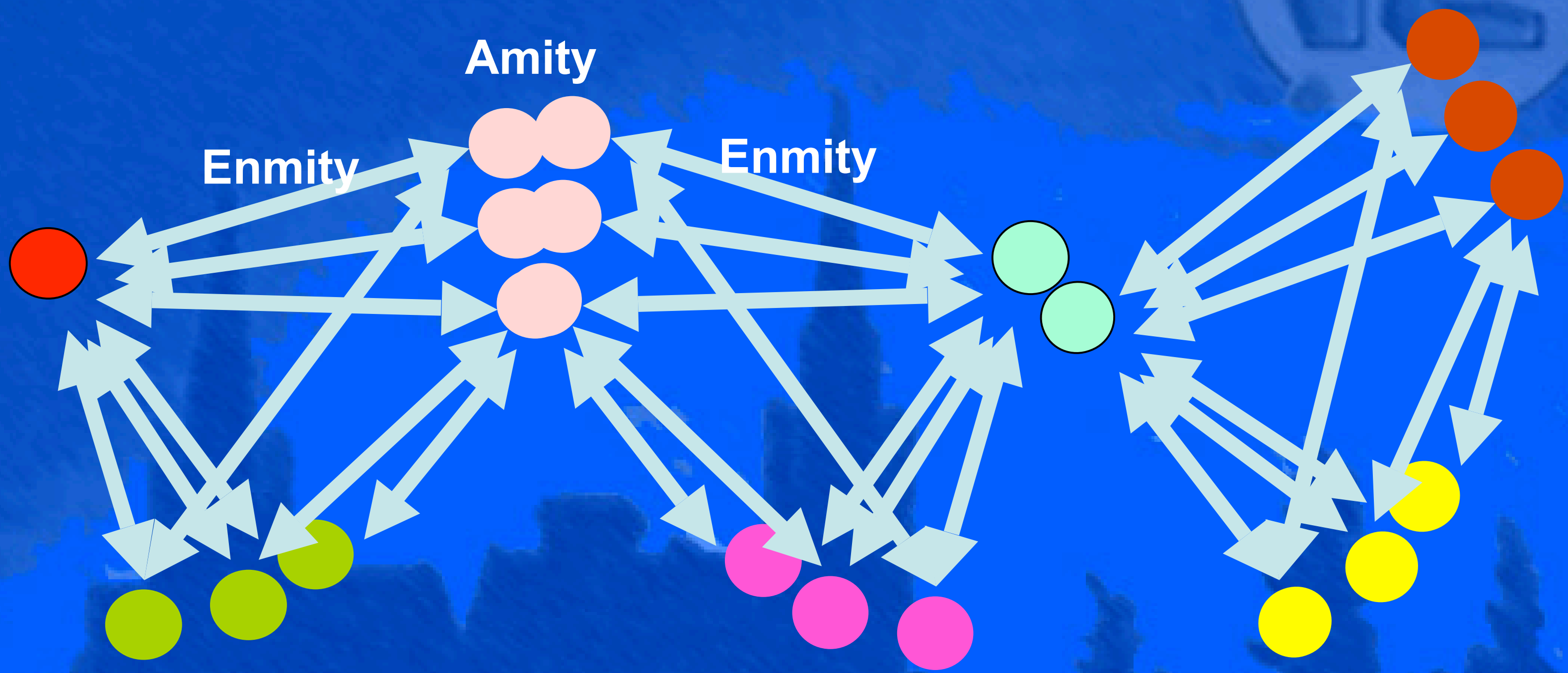
Hypothesis

- Steering behaviors could be useful to find graph drawing algorithms that solve the problem of k-coloration:
  - Toric 2D world.
  - Agents are navigating nodes.
  - Each color is a goal that attracts agents.
- The steering behaviors of pursue-evasion are enough to represent coloration problem:
  - Edges represent the relation of enmity.
  - The enemies of the enemies are friends.
  - An agent evades enemies and pursues friends.
- Attack as a mechanism to avoid suboptimal solutions and ensure convergence if a solution does exist.
- For the sake of simplicity, center in k=3.
- The SOPS algorithm proposed will improve the correctness of the results obtained by Brélaz algorithm.

Matlab application



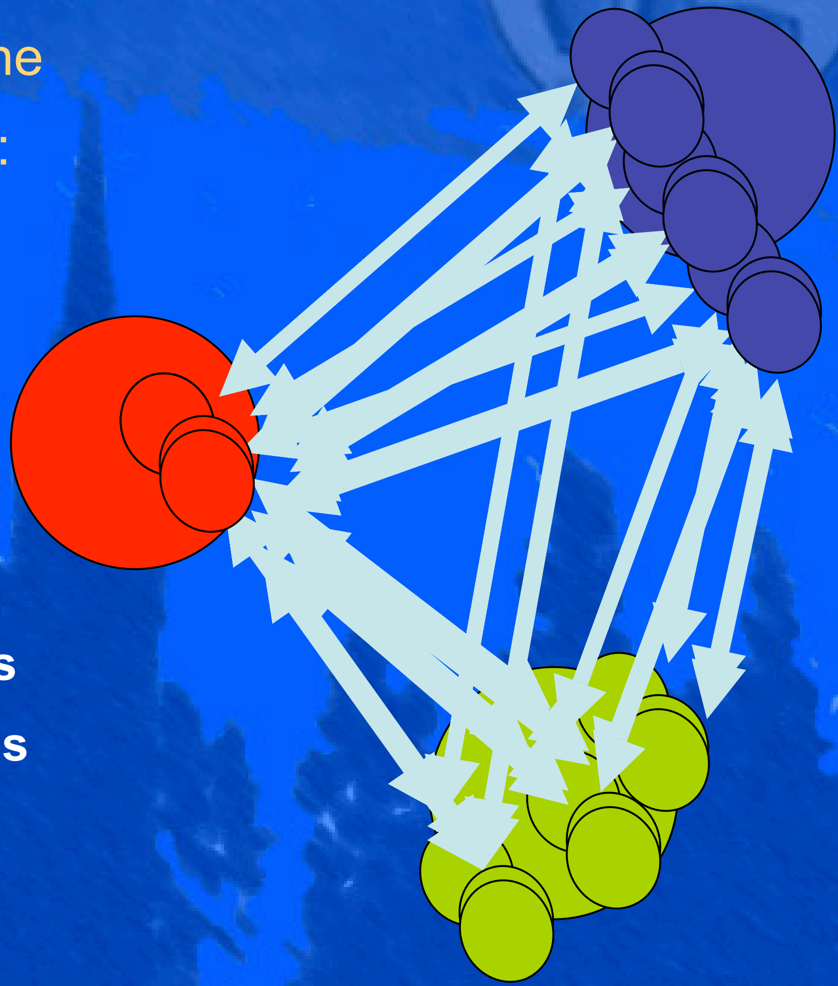
The dynamics of pursue-evasion:



- Components: separation, cohesion and alignment
- If each cluster were a color obtains a consistent coloration
- More clusters than the chromatic number of the graph

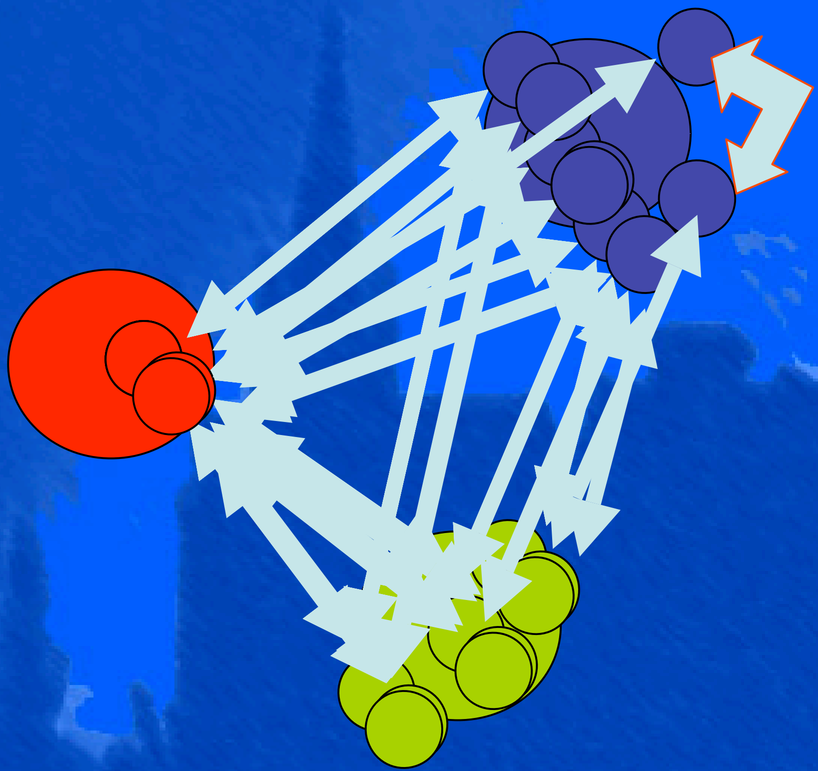
Seek toward goals

- Without the attack mechanism the system always converges either:
  - **To an optimal configuration.**
  - **To a sub-optimal one:**
    - **with few nodes wandering around of the nearest goal.**
    - **with few nodes in the middle.**
  - **A sub-optimal configuration is reached whenever the graph is non 3-colorable**

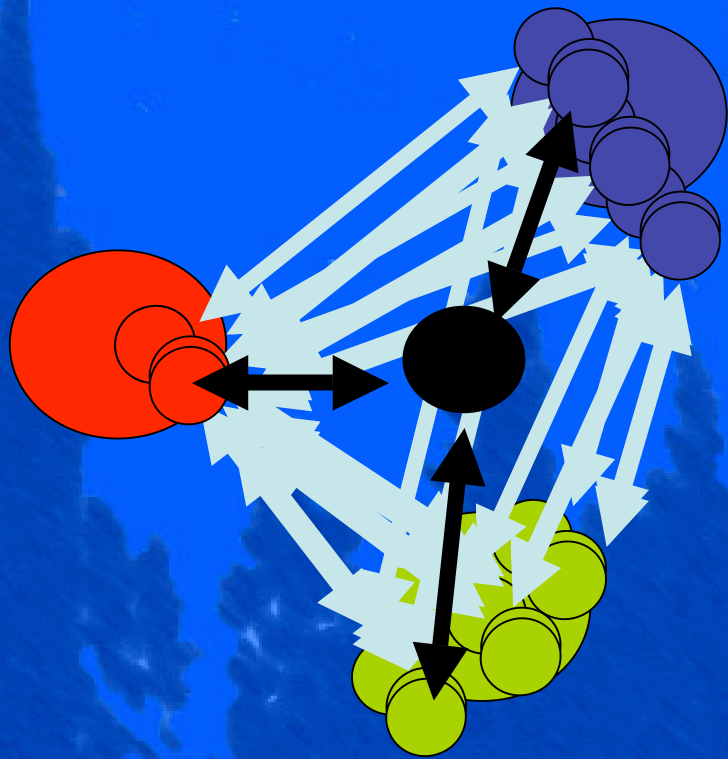


Attack: flee the goals

- **Internal conflict:**
  - Enemies in the same goal



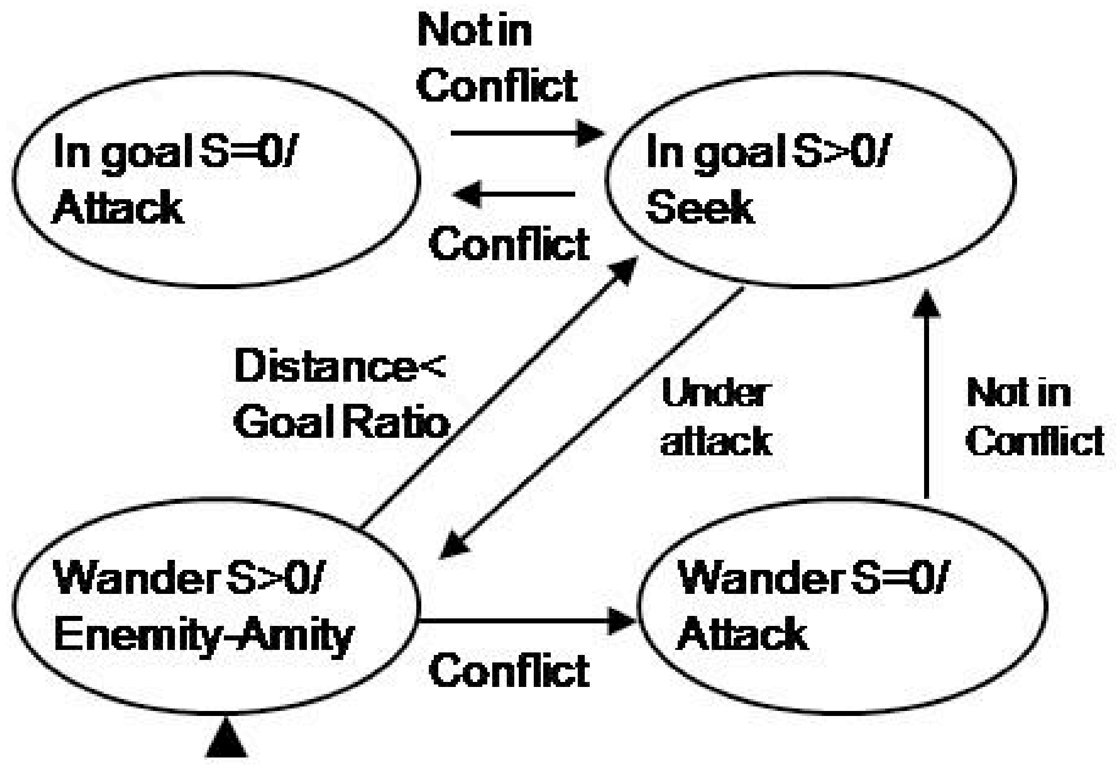
- **External conflict:**
  - Enemies in all the goals



The rule of attack

- 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.
- In a conflict, the satisfaction level decreases until the counter reaches a value below a given threshold.
- In this case, the aggressive behavior is activated and the node attacks.
- The attack consists in selecting randomly an enemy in conflict which is less desperate than the aggressor.
- The node under attack is expulsed 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.

The FSM of individuals



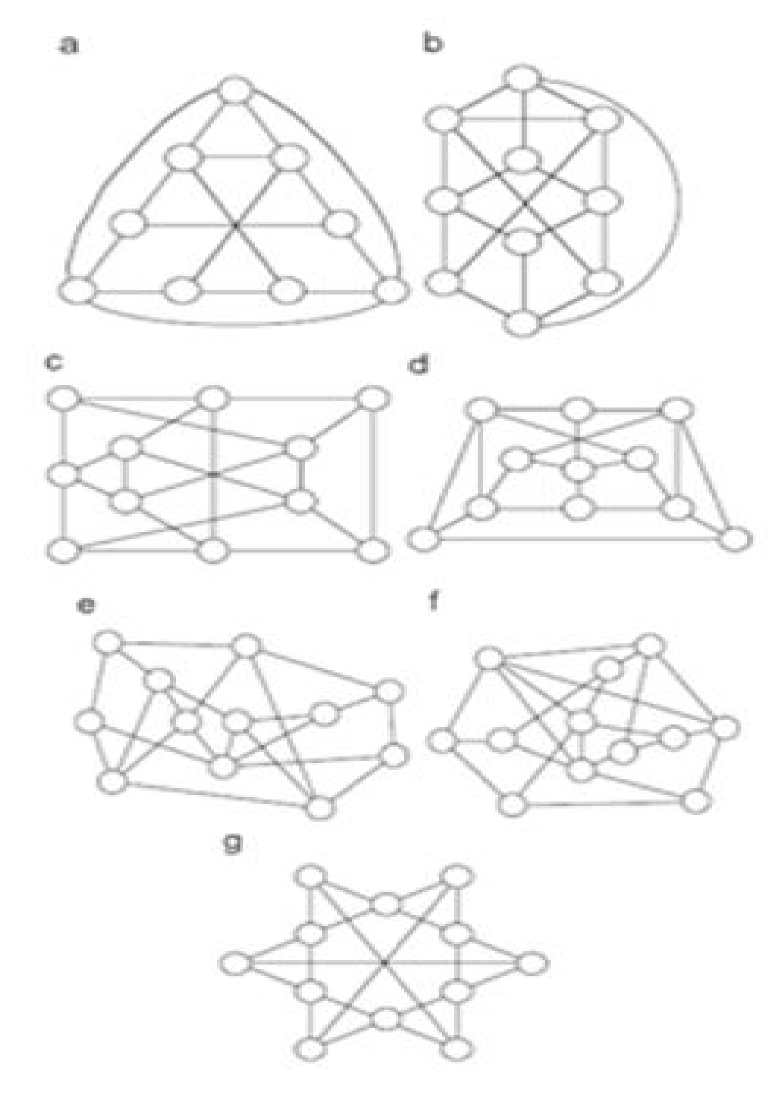
Benchmarking Experiments: a comparison to Brélaz heuristic

- **The problem of 3-coloring of graphs has a very simple formulation but it is very difficult to solve.**
- **Conjecture of Steinberg (1979) open: every planar graph without 4 and 5-cycles is 3-colorable.**
- **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.**

Obtaining the sample

- **Good results for benchmark experiments.**
- **There is a class of graphs that are hard for 3-coloration by Brélaz algorithm Mizuno, K. and Nishihara, S. (2008).**
- **Constructive generation of very hard 3-colorability instances.**
- **Algorithm implemented in Wolfram Mathematica.**

Hard graphs for 3-coloration



- **Embedding: combine graphs to obtains new ones.**
- **The grades of all nodes are 3 or 4.**
- **For all of them the Brélaz chromatic number is 4 while all of them are 3-colorable**
- **We generated a sample of 100 graphs**
- **Each graph a mean of 110 nodes by 10 random embeddings from basic graphs.**

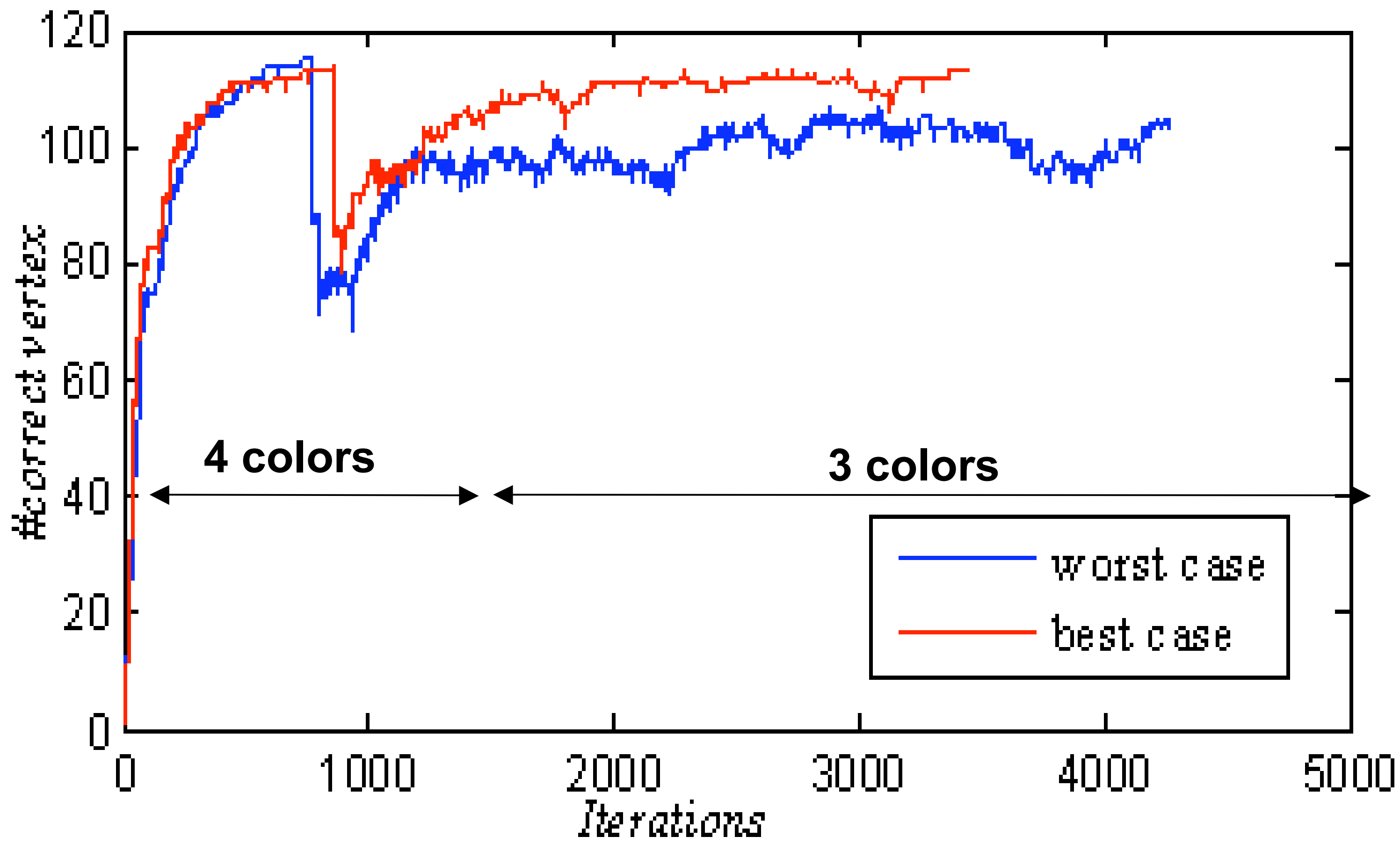
Experimental constrains

- **An experiment consists in:**
  - 25 runs of one graph.
  - **Registering the best configuration for each run:** the one that has minimum number of individuals in conflict.
  - Each run ends either when a 3-coloring solution is reached (success).
  - or when 5000 iterations are completed in cascade.
  - The number of iterations is registered.
- **Parameters are set in the default values.**

Cascade coloration strategy

- To obtain a faster convergence.
- The execution of the program has two stages :
  - First, the system attempts to find a 4-coloration (maximum 1500 iterations)
  - Second, eliminate 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)
- To find the chromatic number of a graph, is sufficient to start the process of coloring successively the graph with colors k, k-1,…

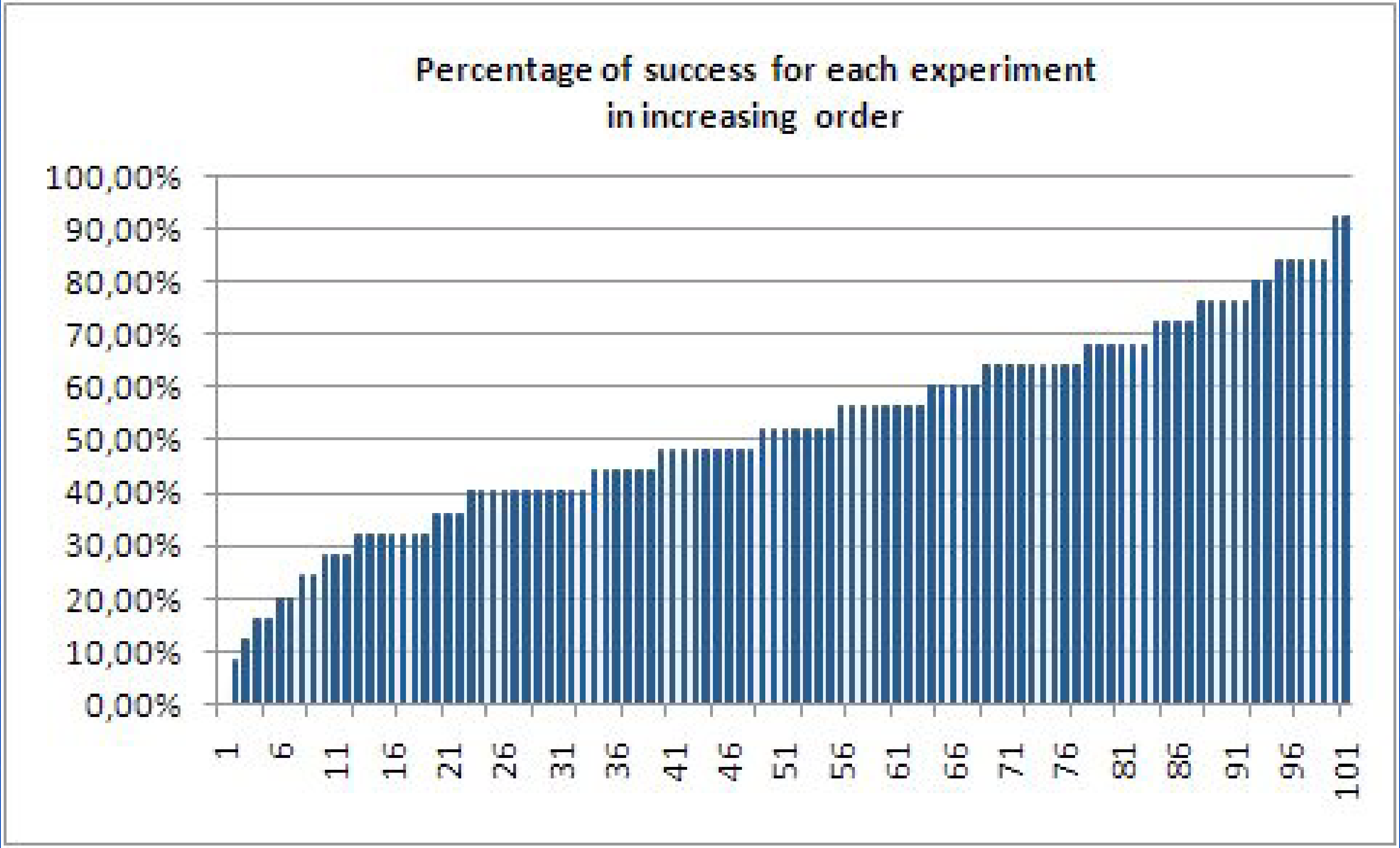
Number of correct colored vertices in cascade iteration of one experiment



Results over 100x25=2500 runs

- Mean number of boids: 110
- Mean number of iterations: 3761
- Average of succeeding runs: 51%

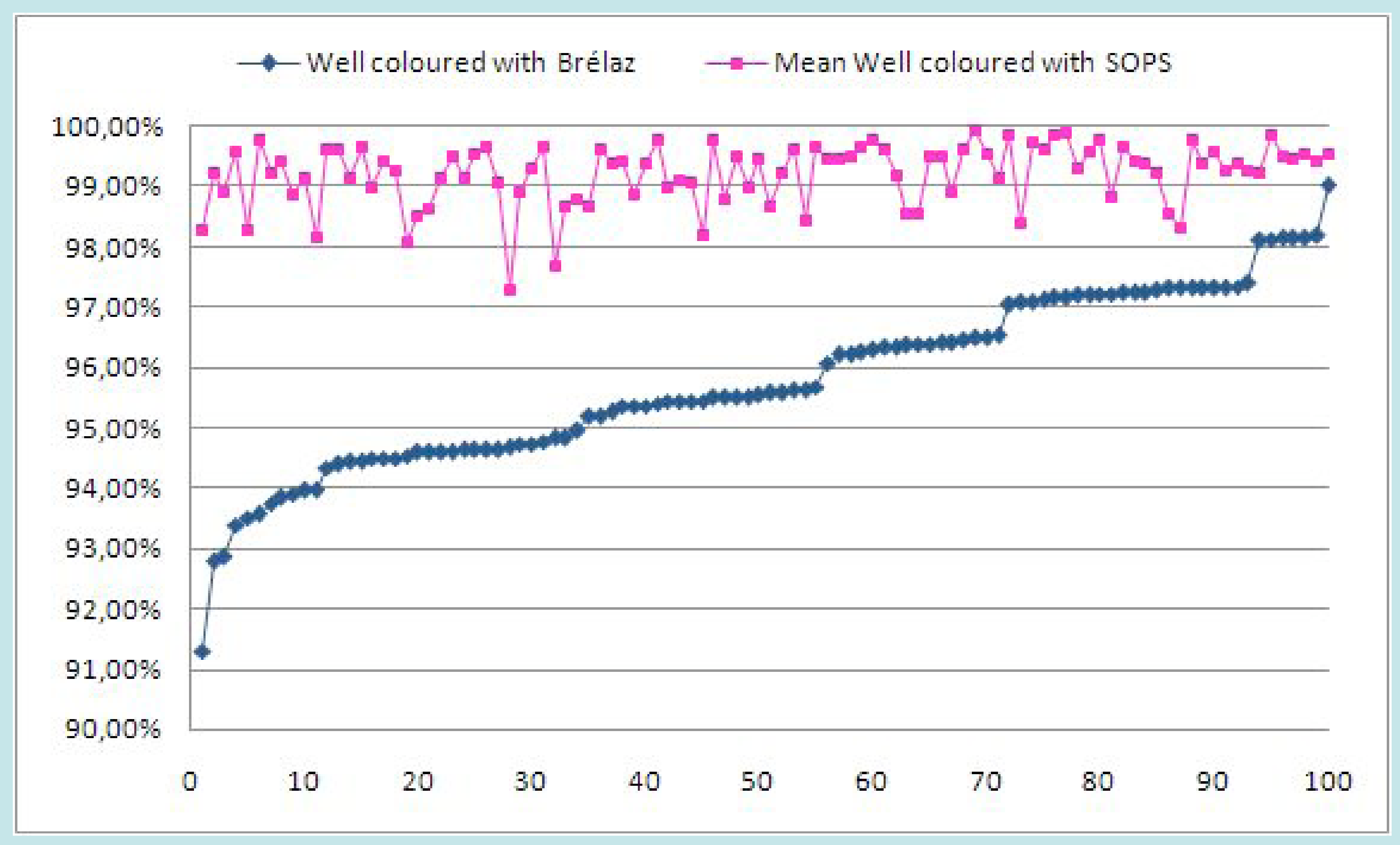
Success of SOPS 3-coloring



Counting errors

- **Brélaz algorithm gives 4 colors for the 100 samples.**
- **Errors in Brélaz coloration: percentage of nodes of color 4.**
- **Errors in SOPS coloration: Take the best configuration through an iteration.**
  - Minimum percentage of conflicting nodes.
- Average results over 100 experiments:
  - **Brélaz algorithm colors well the 95,82% of the nodes with a standard deviation of 1.45%,**
  - **SOPS reaches a mean of the 99.17% and standard deviation 0.60%.**
- Pearson coefficient of 0.30 has been found and in consequence, correlation does not exists between the results.

Brélaz versus SOPS for the benchmark hard graphs



Conclusions

- Self-Organizing Particle System solving the graph coloring problem.
- Cascading procedure of coloration 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:
  - **Is NP-complete**
  - **Steinberg's conjecture is giving an important research**
- Other biologically inspired approaches to this problem:
  - **(Dowsland and Thompson, 2008), ant colony optimization approach. identifies an individual in a population to a whole coloration of the graph, losing the biological inspiration if favor of cognitive abstraction.**
- We do not know of any other attempt to solve the problem using flocking birds.

Conclusions

- Our approach to the coloration of graphs is mainly geometrical: graph drawing.
- 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.


Questions?

Examples of the Matlab application

- fpsol2_i_1_colGRAPH.wmv: **benchmark problem,** 496 nodes and 11654 vertices, 0 fails.
- **PetersenGraph.wmv. 10 nodes, 0 edges, 0 fails.**

Results over 100x25=2500 runs

- Mean number of boids: 110=n

  $o(n) \leq$ Complexity of one step $\leq o(n^2)$
- Mean number of iterations: 3761

  $o(n) \leq 3761 \approx 34x110 \leq o(n^2)$
- Average of succeeding runs: 51%

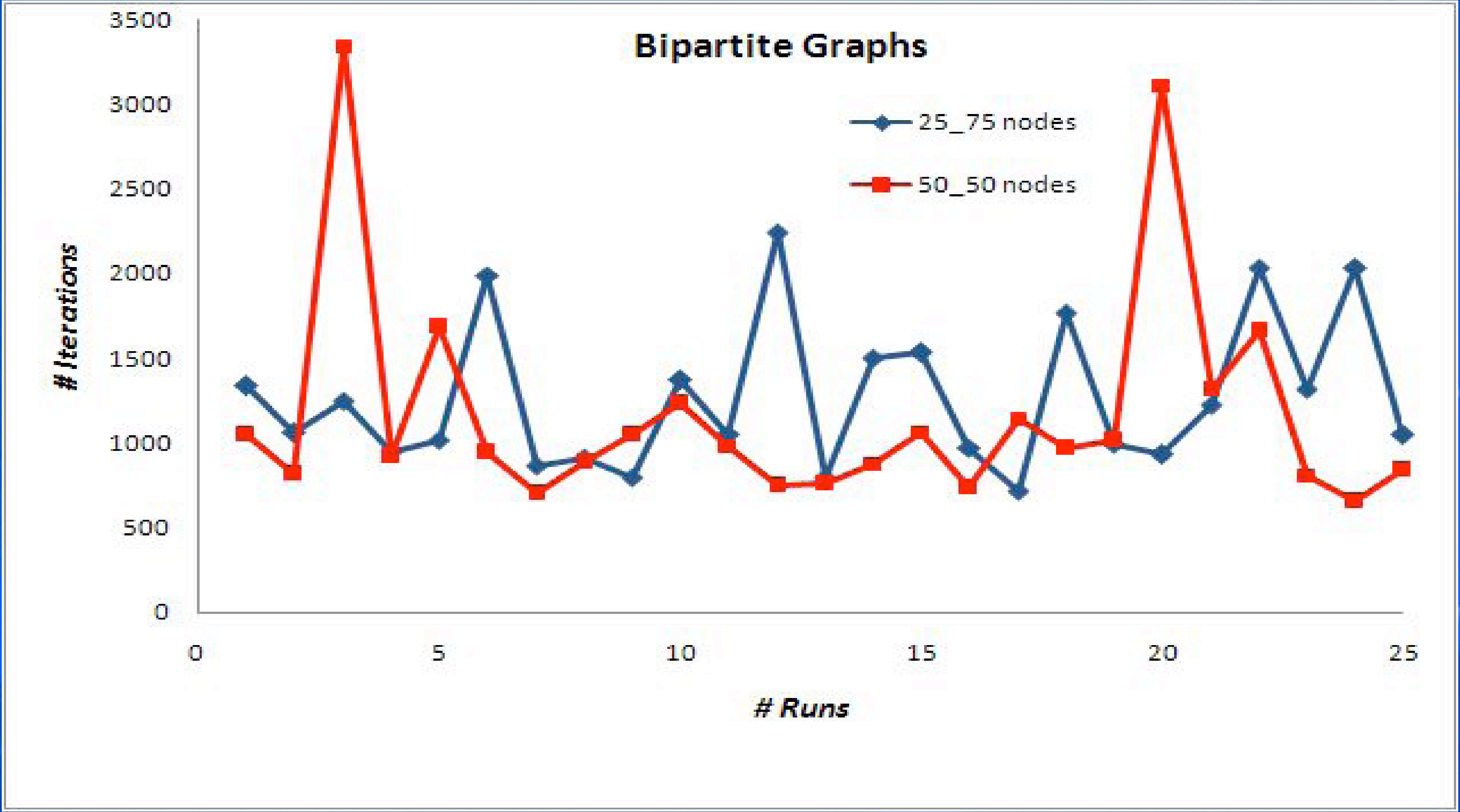
  One of each two runs is successful

  $n^3=n^2xn<$Order of complexity $< 2xn^2xn^2=n^4$

Percentage of well colored nodes

- It is well known that Brélaz algorithm needs two colors for a bipartite graph, being particularly efficient in this case.
- SOPS solves also correctly two complete bipartite graphs of 100 elements: 50-50 nodes and 25-75.
- In the 25 runs of each graph, the run was successful in both cases 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.

Iterations for 2 colorable graphs



Discussion

- Our aim was the research of the behavior 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.
- We found that amity- enmity dynamics allows to model the solving process for coloring graphs.
- Complexity of swarms, understood as the complexity of the
- behavior 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 (Csuhaj-Varju et al., 1994; Kelemen and Kelemenov,
- 1992; Kelemenov´a and Csuhaj-Varj´u, 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 furthers
- 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 Br´elaz algorithm algorithm is the
- common benchmark algorithm. Our approach improves on
- it over a sample of hard graphs.

Discussion

- We have designed and implemented a Self-Organizing Particle
- System that may interpreted as solving the graph colouring
- 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 Steinbergs conjecture is giving
- arise an important research nowadays (Borodin et al., 2005).
- A recent biologically inspired approaches to this problem
- has used the ant colony optimisation approach (Dowsland
- 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 (u1; :::; un) where ui
- is the colour of node i, losing in this way the biological inspiration
- if 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- enemity
- 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

Brélaz Algorithm

- **Greedy algorithm for graph coloring.**
- **Colors={1, 2, …, k}**
- **D(v)=degree of node v= number of neighbors.**
- **S(v)=saturation=number of different colors for neighbors.**
  - **Start in a vertex of maximum degree with color 1.**
  - **Color first the nodes v with maximum S(v)**
  - **Color node with maximum D(v) if saturation is the same.**
  - **Use the minimum available color.**
- Brélaz is a function provided by Wolfram Mathematica.

Basic rules of Reynold’s model of boids

- **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 ci 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.

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

Seek and flee toward goals

- **Seek and flee**: seek attempts to steer a vehicle so that it moves toward a static goal. Here $||p||$ denotes the norm of a position or vector p and $f_{\text{max velocity}}$ is a non-negative parameter that limits the norm (the length in the Euclidean distance) of vector v_{seek}.

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

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

Flee velocity is definided simply as the opposite of seek,

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