# Gravitational Swarm approach for Graph Coloring

Israel Rebollo[1,2]    Manuel Graña[1]

[1]Computational Intelligence Group- University of the Basque Country
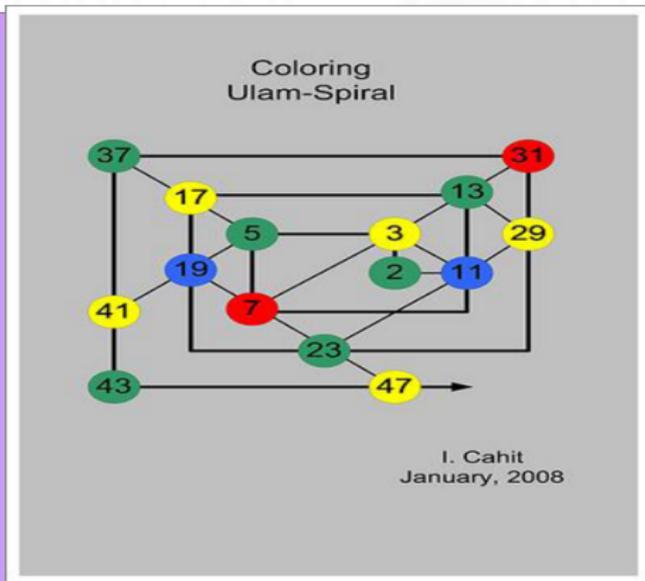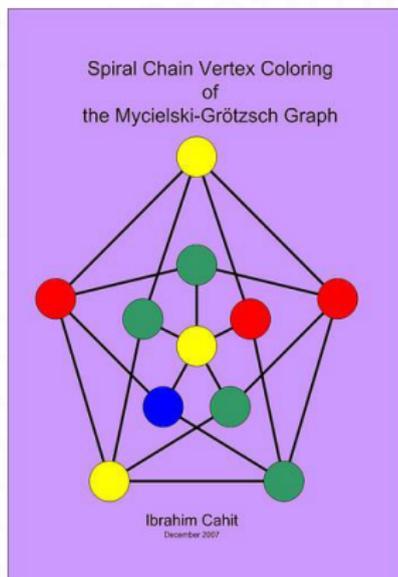
[2]Informática 68 Investigación y Desarrollo, S.L

Nature Inspired Cooperative Strategies for Optimization,2011

# Outline

# Graph Coloring Problem

- The graph coloring problem GCP: consist in assigning a color to the vertices of a graph with the limitation that a pair of vertices that are linked cannot have the same color.



Spiral Chain Vertex Coloring of the Mycielski-Grötzsch Graph — Ibrahim Cahit, December 2007

Coloring Ulam-Spiral — I. Cahit, January, 2008

# Swarm Intelligence.

- Swarm Intelligence: is a model where the emergent collective behavior is the outcome of a process of self-organization, where the agents evolve autonomously following a set of internal rules for its motion and interaction with the environment and the other agents.
    - There is no leader.
    - Has a high level of scalarity.
    - The failure of some agents would not alter too much the overall system.

# The model.

- A Swarm of agents move through a toric world.
- The agents are attracted by the goals, each goal represent a color.
- The agents have no information about the global problem, they only know the relationship friend or foe between them.
- If an agent arrives into a goal then it gets that color and stop moving.

# The model.

## Definitions

Let be $G = (V, E)$ a graph with $V$ vertices and $E$ edges.
Let have $F = \{B, \overrightarrow{v}, C, \overrightarrow{v_g}\}$ where $B = \{b_1, b_2, ..., b_n\}$ is the group of SI agents, $\overrightarrow{v}$ the speed vector in the instant $t$, $C = \{1, 2, ..., k\}$ the chromatic number of the graph and $\overrightarrow{v_g}$ the attraction to the goal.

## Fact

$f = |\forall i, Cb_i \in C$ and $\nexists j \mid enemy(b_i, b_j) \wedge Cb_i = Cb_j|$

- This energy is the count of the graph nodes which have a color assigned.
- The predicate enemy is true if a pair of SI agents have an edge between them, so they cannot have the same color.
- $Cb_i$ denotes the color of the SI agent associated with the target goal.

# The dinamic of the system.

- The agents move through the space with a speed $v$.
- The goals attract the agents with a force proportionally to the euclidean distance.
- When an agent reaches to a goal:
  - There are no enemies, so the agent take the goal color and stops.
  - There are enemies, so the agent try to expell the enemies of the goal.
- When all the agents ara inside goals without enemies, the problem is solved.

# Flowchart.
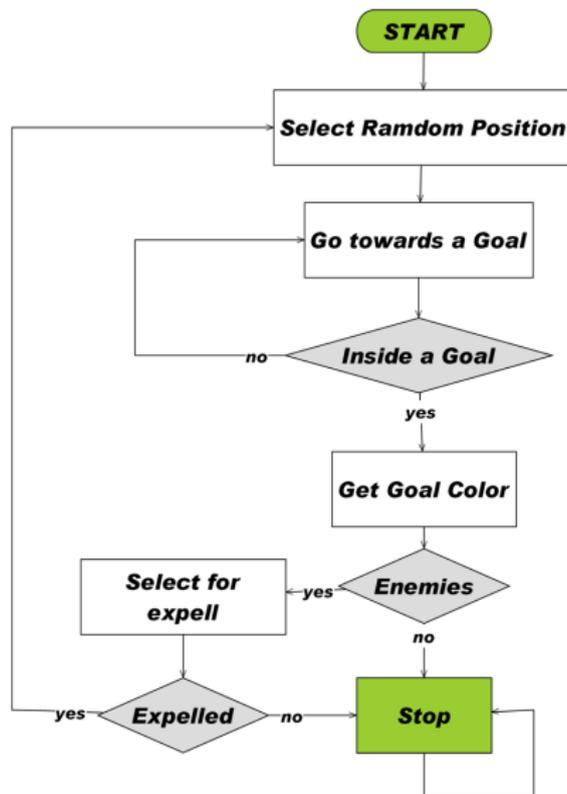
# Algorithm.

1. Supervised Coloring: the chromatic number is given to the algorithm.
2. Unsupervised Coloring: the goals are romeved and the agent move through the space creating clusters, the number of clusters is the chromatic number of the graph.
3. Coloring Waterfall

---

**Algorithm 1** Coloring waterfall

Input: $G$ the graph to color.
Upper_bound = unsupervised_coloring($G$)
while Upper_bound > 1 do
  let Upper_bound = Upper_bound - 1
  solution = supervised_coloring($G$,(Upper_bound),max_time)
  if not solution then
    chromatic_number = (Upper_bound+1)
    exit while
  end if
end while

---

## Convergence issue.

- The gravitational fields cover all the space, so all the agents moves towards a goal.
- If an agents arrive to a goal and can go inside then stoped.
- If all the agents speed is zero, then the system has converged to some fixed state.
- This state must be a solution of the problem, because:
    - An agent only stops if it is inside a goal without enemies.
    - If one agent never stops it means that the initial chromatic number is not a solution of the system.

# Experimental results.

1. We have used DIMACS well known graphs.
2. We implement our GSI algorithm, and also three more algorithm to compare with:
   1. Backtraking.
   2. Brèlaz DSATUR
   3. Tabu Search
3. We let the algorithms a maximum number of steps or cicles to find a solution.
4. We have also compare our results with test and bechmarks that appears in the bibliography.

# Previous methos.

Graph coloring problem solved with Backtraking, DSATUR and Tabu Search:

| Graph name | k | #Backtracking | #DSATUR | #Tabu Search | %Tabu success |
|---|---|---|---|---|---|
| Myciel3 | 4 | 1 | 1 | 11 | 100 |
| Myciel4 | 5 | 1 | 1 | 34 | 100 |
| Myciel5 | 6 | 1 | 1 | 107 | 100 |
| Myciel6 | 7 | 1 | 1 | 290 | 100 |
| Myciel7 | 8 | 1 | 1 | 597 | 100 |
| queen5_5 | 5 | 5 | 5 | 442 | 95 |
| anna | 11 | * | * | * | 0 |
| david | 11 | * | * | * | 0 |
| huck | 11 | * | * | * | 0 |

# Gravitational Swarm Intelligence.

| Graph name | #Vertices | #Edges | Density | K | Success % | Average #Steps |
|---|---|---|---|---|---|---|
| anna | 138 | 986 | 0.10 | 11 | 100 | 300 |
| david | 87 | 812 | 0.21 | 11 | 100 | 209 |
| huck | 74 | 662 | 0.22 | 11 | 100 | 84 |
| jean | 80 | 508 | 0.16 | 10 | 100 | 165 |
| myciel3 | 11 | 20 | 0.36 | 4 | 100 | 21 |
| myciel4 | 23 | 71 | 0.28 | 5 | 100 | 25 |
| myciel5 | 47 | 236 | 0.21 | 6 | 100 | 97 |
| myciel6 | 95 | 755 | 0.17 | 7 | 100 | 92 |
| myciel7 | 191 | 2360 | 0.13 | 8 | 100 | 417 |
| queen 5x5 | 25 | 160 | 0.53 | 5 | 100 | 302 |
| 1_fullins_3 | 30 | 100 | 0.23 | 4 | 100 | 37 |
| 1_fullins_4 | 93 | 593 | 0.14 | 5 | 100 | 76 |
| 1_fullins_5 | 282 | 3247 | 0.08 | 6 | 100 | 222 |
| 2_fullins_3 | 52 | 201 | 0.15 | 5 | 100 | 67 |
| 2_fullins_4 | 212 | 1621 | 0.07 | 6 | 100 | 176 |
| miles250 | 128 | 387 | 0.04 | 8 | 100 | 317 |

# Conclusions.

- We proposed a new algorithm for the Graph Coloring Problem using Swarm Intelligence.
- We have modeled the problem as a collection of agents trying to reach some of a set of goals.
- We have solved the GCP using a parallel evolution of the agents in the space.
- We have argued the convergence of the system, and we have demonstrated empirically that it provides effective solutions in terms of precision and computational time.

# Future work.

- We will continue to test our algorithm on an extensive collection of graphs, comparing its results with state of the art heuristic algorithms.
- For future work, we have to improve our implementation of the algorithm to make it faster.
- Even though our algorithm finds the global optimum of the problem, we will search for new nature inspired behavior rules to improve the algorithm.

Thanks for your attention