Further results of Gravitational Swarm Intelligence for Graph Coloring

Israel Rebollo[1,2] Manuel Graña[1]

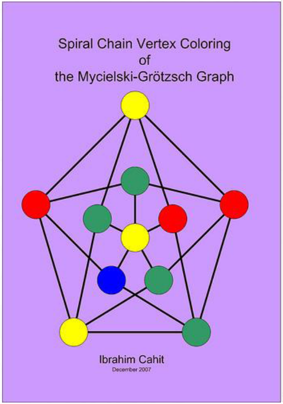
[1]Computational Intelligence Group- University of the Basque Country

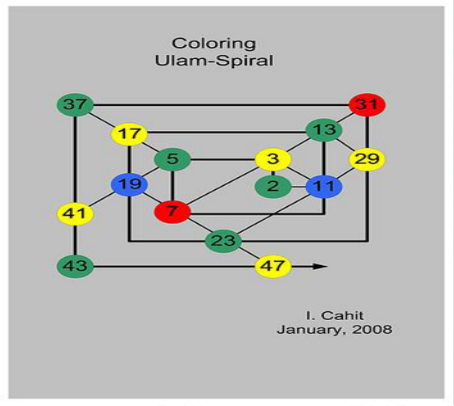
[2]Informática 68 Investigación y Desarrollo, S.L

Nature and Biologically Inspired Computing,2011

Outline

- 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.





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 scalability.
  - The failure of some agents would not alter too much the overall system.

The model.

- The natural inspiration came from the physic law of the gravitational atraction between objects.
- 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.

- Let be $G = (V, E)$ a graph with V vertices and E edges.
- Let have $F = \{B, CG, \{\overrightarrow{v_i}\}, K, \{\overrightarrow{a_{i,k}}\}, R\}$ where:
  - $B = \{b_1, b_2, ..., b_n\}$ is the group of SI agents.
  - $CG = \{g_1, g_2, ..., g_k\}$ the color goals.
  - $\{\overrightarrow{v_i}\}$ the speed vector in the instant t.
  - $C = \{1, 2, ..., k\}$ the number of colors.
  - $\{\overrightarrow{a_{i,k}}\}$ the attraction forces of the color goal.
  - R denotes the repulsion forces in the neighbourhood of color goals.

Fact

$f(B, CG) = |\{b_i \ s.t. \ c_i \in C \ \& \ R(b_i, g_{c_i}) = 0\}|$

The model.

- This cost function is the count of number of graph nodes which have a color assigned and no conflict inside the goal.
- The agents outside the neighbourhood of any color goal can't be evaluated, they are not part of the solution.
- The dimension of the world and the goal radius parameters determine the convergence speed of the algorithm:
    - With a big world, the convergence is slow but monotonically to the solution.
    - With a big goal radius, is faster but because the algorithm falls in local minima.

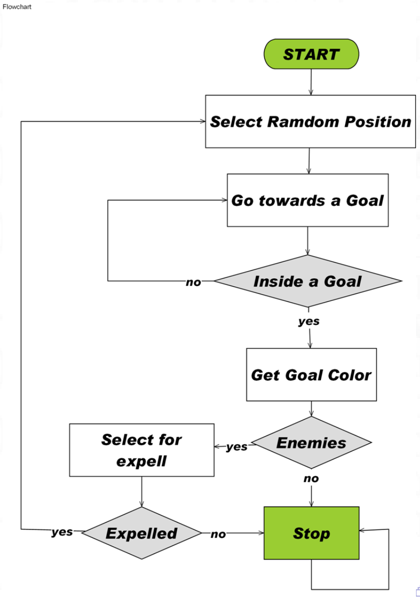
The dinamic of the system.

The dynamics of each GSI agent in the world is specified by the iteration:

$$\overrightarrow{v_i}(t+1) = \begin{cases} 0 & c_i \in C \& (\lambda_i = 1) \\ d \cdot \overrightarrow{a_{i,k^*}} & c_i \notin C \\ \overrightarrow{v_r} \cdot (p_r - p_i) & (c_i \in C) \& (\lambda_i = 0) \end{cases}$$

- Where d is the vector difference of the agent's position p_i and the position of the nearest color goal g_{k^*}.
- $\overrightarrow{a_{i,k^*}}$ represents the attraction force to approach the nearest goal
- $\overrightarrow{v_r}$ is a random vector to avoid being stuck in spurious unstable equilibrium, towards a random position p_r. Parameter λ_i represents the effect of the degree of *Comfort* of the GSI agent.
    - When a GSI agent b_i reaches to a goal in an instant t, its velocity becomes 0 and $\lambda_i = 0$
    - $\lambda_i = 1$ in other case.

Agent's Dinamic Flowchart.



Convergence issues.

- The gravitational fields cover all the space, so all the agents move towards a goal.
- If an agent arrives to a goal and can go inside then stopped.
- 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.

- We have used DIMACS well known graphs.
- We implement our GSI algorithm, and also four more algorithms to compare with.
- We let the algorithms a maximum number of steps or cicles to find a solution.
  - And also a maximum time.
- We have also compare our results with test and bechmarks that appearing in the bibliography.

Competing Algorithms

1. A greedy backtracking algorithm: this algorithm explores all the search space and always return the optimal solution if exists.
2. DSATUR (Degree of Saturation): this algorithm developed by Brèlaz is a greedy backtraking algorithm but does not explore exhaustively all the search space.
3. Tabu Seach: it is a random local search with some memory of the previous steps, so the best solution is always retained while exploring the environment.
4. Simulated Annealing: this random algorithm has a big problem in the graph coloring problem, because there are a lot of neighboring states that have the same energy value.

Graph Coloring Results.

Graph coloring results over the test graphs. The * means that no solution is found in the given time.

		BT	DSATUR	TS		SA		GSI	
Graph name	K	#back	#back	#iter	%success	#iter	%success	#iter	%success
Myciel3	4	1	1	13	100	21	100	25	100
Myciel4	5	1	1	51	100	716	100	46	100
Myciel5	6	1	1	393	96	407074	28	241	100
Myciel6	7	1	1	970	94	*	0	630	100
Myciel7	8	1	1	1575	92	*	0	1103	98
anna	11	*	1	4921	2	483859	6	718	98
david	11	*	1	*	0	478207	10	1428	92
homer	13	*	*	*	0	*	0	2583	76
huck	11	1	1	3363	54	180975	64	251	98
jean	10	1	1	2471	68	281418	44	439	98

Computation time.

Computation time in seconds. The * means that the algorithm hasn't find a solution in 3 hours time.

Graph Name	BT	DSATUR	TS	SA	GSI
Myciel3	1	1	1	1	1
Myciel4	1	1	1	1	1
Myciel5	1	1	11	1067	9
Myciel6	1	1	69	*	55
Myciel7	1	1	307	*	210
anna	*	2	959	596	137
david	*	1	*	319	177
homer	*	*	*	*	2456
huck	1	1	276	134	26
jean	1	1	206	239	48

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.

Definition

Goals represent node colorings, agents represent graph's nodes.
The color goals exert a kind of gravitational attraction over the entire virtual world space.

- With these assumptions, we have solved the GCP using a parallel evolution of the agents in the space.
- We have argued the convergence of the system.
- 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.
- We are working on a formal convergence proof of the algorithm dynamics.

Thanks for your attention.

You can contact in http:\\www.ehu.es\ccwintco