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

1. Introduction

2. Gravitational Swarm Intelligence
   - The model
   - Agent’s Dynamic Flowchart

3. Convergence issues

4. Experimental results
   - Competing Algorithms
   - Graph Coloring Results
   - Computation time

5. Conclusions
   - Conclusions
   - Future work
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.
- 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 physical law of the gravitational attraction between objects.
- A Swarm of agents move through a toric world.
- The agents are attracted by the goals, each goal represents 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 stops moving.
The model.

- Let be $G = (V, E)$ a graph with $V$ vertices and $E$ edges.
- Let have $F = \{B, CG, \{\overrightarrow{v_i}\}, K, \{a_{i,k}\}, R\}$ 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

$$f(B, CG) = \left| \{b_i \text{ s.t. } c_i \in C \& R(b_i, g_{c_i}) = 0\} \right|$$
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 dynamics of each GSI agent in the world is specified by the iteration:

\[
\mathbf{v}_i(t+1) = \begin{cases} 
0 & \text{if } c_i \in C \& (\lambda_i = 1) \\
\mathbf{d} \cdot \mathbf{a}_{i,k^*} & \text{if } c_i \notin C \\
\mathbf{v}_r \cdot (\mathbf{p}_r - \mathbf{p}_i) & \text{if } c_i \in C \& (\lambda_i = 0)
\end{cases}
\]

- Where \( \mathbf{d} \) is the vector difference of the agent’s position \( \mathbf{p}_i \) and the position of the nearest color goal \( g_{k^*} \).
- \( \mathbf{a}_{i,k^*} \) represents the attraction force to approach the nearest goal.
- \( \mathbf{v}_r \) is a random vector to avoid being stuck in spurious unstable equilibrium, towards a random position \( \mathbf{p}_r \). Parameter \( \lambda_i \) 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 Dynamic Flowchart.

Flowchart:

START

Select Random Position

Go towards a Goal

Inside a Goal

Get Goal Color

Select for expell

Enemies

Expelled

Stop
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 benchmarks 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 Search: 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.

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

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

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