

Gravitational Swarm for Graph Coloring

Thesis dissertation

Israel Carlos Rebollo Ruiz
Advisor: Prof. Dr. Manuel Graña

Computational Intelligence Group
University of the Basque Country(UPV/EHU)
Informática 68 Investigación y Desarrollo S.L.

2012-07-27



Overview

- 1 Introducción
- 2 Gravitational Swarm for Graph Coloring
- 3 Estudio sensibilidad de GSGC a sus parámetros
- 4 Algoritmos implementados
- 5 Resultados comparativos
- 6 Conclusiones

Outline

- 1 Introducción
- 2 Gravitational Swarm for Graph Coloring
- 3 Estudio sensibilidad de GSGC a sus parámetros
- 4 Algoritmos implementados
- 5 Resultados comparativos
- 6 Conclusiones

Coloreado de Grafos

- El GCP consiste en asignar colores a los nodos de un grafo de forma que ningún par de nodos unidos por una arista tienen el mismo color.
- El coloreado óptimo es aquel que resuelve el GCP con el mínimo número de colores, llamado número cromático.

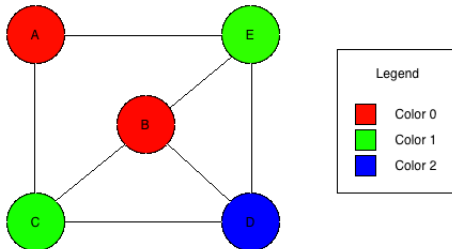


Figura : Ejemplo de coloreado de un Grafo

Motivación

- Hemos elegido el Problema del Coloreado de Grafos(GCP) por:
 - Su historia: es un problema clásico, ampliamente estudiado y fácil de entender e implementar.
 - Actualmente siguen apareciendo nuevos algoritmos para tipos de grafos nuevos o heurísticos para solucionar el problema.
 - Hay aplicaciones prácticas en las que un problema se mapea como un grafo y se resuelve con GCP, tales como:
 - Detección de comunidades en redes sociales.
 - En Robótica para el guiado automático de vehículos.

Motivación II

- Preguntas que queremos responder:
 - ¿Es posible mapear el GCP en un sistema tipo enjambre de Reynolds?
 - ¿Es posible definir una dinámica que lleve al sistema a soluciones válidas del problema?
 - ¿Es posible obtener soluciones dado un número k de colores?
 - ¿Es posible obtener el número cromático de un grafo?

Swarm Intelligence (SI)

SI: es la aparición de configuraciones significativas en el comportamiento colectivo de sistemas descentralizados y auto-organizados, cuya dinámica se inspira en la naturaleza.

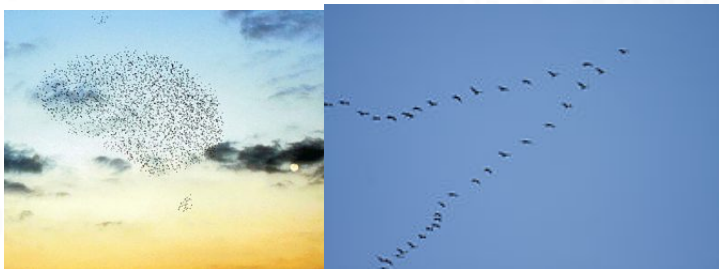


Figura : Bandadas de pájaros

Ejemplo SI con hormigas

Un ejemplo clásico de Swarm Intelligence son las colonias de hormigas que solucionan el problema del camino mínimo.

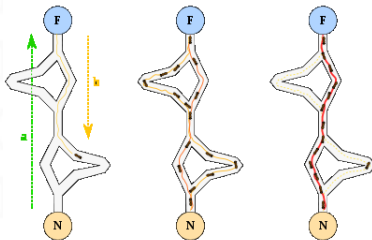


Figura : Solución clásica de camino óptimo con ACO

Contribuciones

Hemos desarrollado un algoritmo novedoso inspirado en Swarm Intelligence y en la ley de la gravedad.

- Presentamos la demostración formal de ciertas propiedades de este algoritmo.
- Mostramos un amplio conjunto de resultados experimentales.

Hemos desarrollado un nuevo método de generación de grafos que hemos llamado KRG.

- El número cromático es conocido.

Outline

- 1 Introducción
- 2 Gravitational Swarm for Graph Coloring**
- 3 Estudio sensibilidad de GSGC a sus parámetros
- 4 Algoritmos implementados
- 5 Resultados comparativos
- 6 Conclusiones

GSGC

- El algoritmo GSGC inspirado en SI de Reynolds y la ley de la gravedad.
- Las partículas representan nodos del grafo que se mueven por el espacio atraídos por puntos fijos llamados metas de color.
- Las partículas desconocen la función objetivo global pero conocen la matriz de adyacencia con el resto de las partículas.
- Existe una fuerza de repulsión que ejercen las partículas entre si cuando:
 - Están dentro de una misma meta de color.
 - La matriz de adyacencia indica que ambas partículas están unidas por una arista.

Flowchart resumido de una partícula

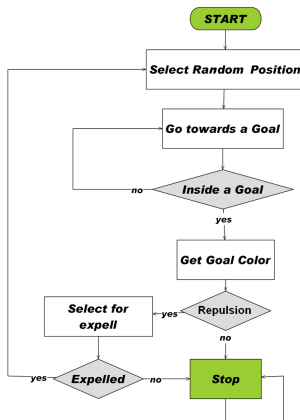


Figura : Flowchart resumido de una partícula

Confort

Definimos el confort como un parámetro que, para cada una de las partículas, indica cuando están dentro de una meta, si dicha partícula puede ser expulsada de la meta o no.

Flowchart completo

Flowchart completo

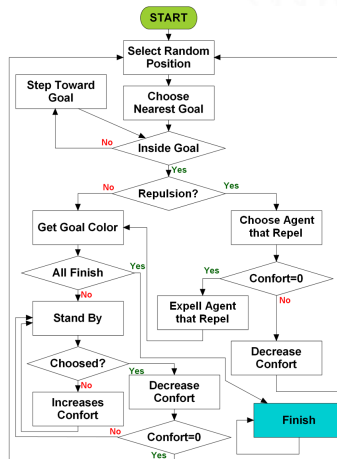


Figura : Flowchart completo

Estado estacionario del GS

Lemma

(25) Una partícula p tiene velocidad cero si esta en una meta y ninguna otra partícula de esa meta la repele.

Lemma

(26) El sistema GS tiene al menos un estado estacionario cuando la relación de repulsión entre partículas forma una relación de equivalencia.

Lemma

(27) El sistema converge si se cumple la condición anterior, puesto que todo estado no estacionario lleva a un estado estacionario.

Estado estacionario del GSGC

Lemma

(32) Una partícula p se detiene si esta a una distancia de la meta menor a un radio dado y las fuerzas de repulsión a una distancia mayor a un valor.

Corollary

(34) Por tanto las metas deben estar separadas por una distancia mayor que la zona de influencia de las fuerzas de repulsión.

Theorem

(39) El sistema está en un estado estacionario si todas las partículas están en el entorno de alguna meta y no existen fuerzas de repulsión entre ellas.

Estado estacionario del GSGC II

Theorem

(41) Si el número cromático es menor o igual que el número de metas de colores fijado, existe un conjunto no vacío de estados estacionarios.

Theorem

(43) Todo estado estacionario del algoritmo GSGC es una solución del GCP.

Theorem

(45) Si el número cromático es mayor que el número de colores, el conjunto de estados estacionarios es vacío.

Observaciones

Remark

(46) el teorema 45 nos da información sobre el número cromático.

Remark

(47) Debemos demostrar la convergencia dinámica del sistema.

Outline

- 1 Introducción
- 2 Gravitational Swarm for Graph Coloring
- 3 Estudio sensibilidad de GSGC a sus parámetros**
- 4 Algoritmos implementados
- 5 Resultados comparativos
- 6 Conclusiones

Estudio sensibilidad de GSGC a sus parámetros

- Parámetros del modelo
- Experimentos de parametrización
 - Radio de meta.
 - Confort.
- Test no Paramétricos:
 - Friedman
 - Nemenyi

Parámetros del modelo

- Tamaño del mundo
- Número de colores
- Radio de Meta
- Confort
- Máximo número de pasos

Valores probados

- Radio de Meta:

- | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|
| 1 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
|---|----|----|----|----|----|----|----|----|

- Confort:

- | | | | | | |
|---|---|---|----|----|----|
| 0 | 1 | 5 | 10 | 15 | 20 |
|---|---|---|----|----|----|

- Máximo número de pasos: 100

- Repeticiones: 1000

- Número de colores: 10 familias KRG de 20 grafos

Radio de meta: Resultados

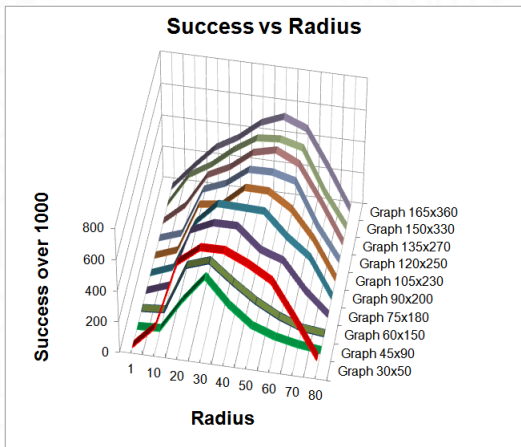


Figura : Efecto del radio

Radio de meta: Tiempo en pasos

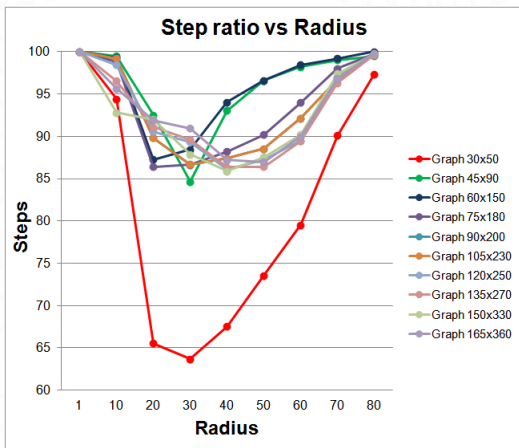


Figura : Promedio de tiempo en pasos

Confort: Resultados

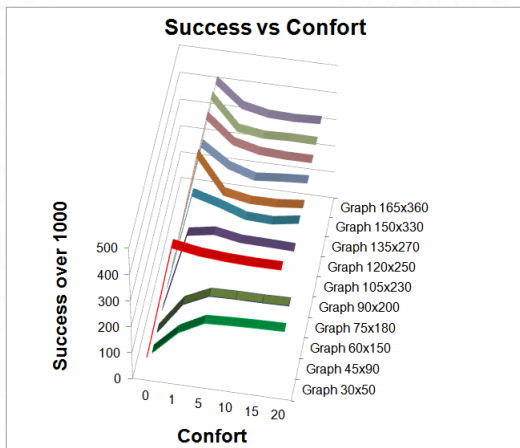


Figura : Efecto del confort

Confort: Tiempo en pasos

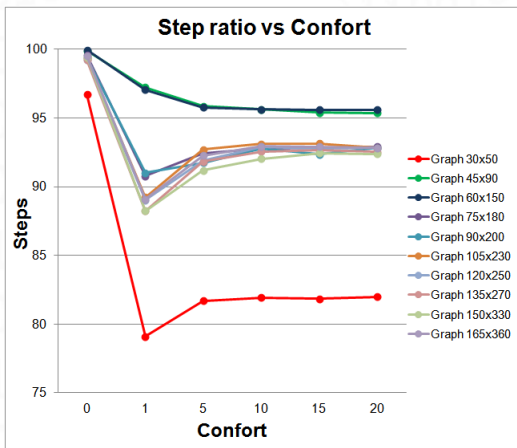


Figura : Promedio de tiempo en pasos

Test de Friedman y Nemenyi

- El test de Friedman es un test no paramétrico que indica si distintos tratamientos t sobre distintos conjuntos de prueba p son estadísticamente significativos.
 - Se utiliza una tabla de rankings para los distintos tratamientos en los conjuntos de test.
 - Si se rechaza la hipótesis nula que indica que los tratamientos no son estadísticamente significativos se pueden realizar test post-hoc.
- El test de Nemenyi es un test post-hoc que indica si los distintos tratamientos pueden formar subconjuntos de valores indiferentes.

Ranking Radios de Meta

Graph \ Radius	1	10	20	30	40	50	60	70	80
Graph 30x50	9	7	4	2	1	3	5	6	8
Graph 45x90	9	8	3	1	2	4	5	6	7
Graph 60x150	9	7	2	1	3	4	5	6	8
Graph 75x180	9	7	3	2	1	4	5	6	8
Graph 90x200	9	7	4	1	2	3	5	6	8
Graph 105x230	9	7	5	4	1	2	3	6	8
Graph 120x250	9	7	5	4	2	1	3	6	8
Graph 135x270	9	7	5	4	2	1	3	6	8
Graph 150x330	9	7	5	4	2	1	3	6	8
Graph 165x360	9	7	5	4	3	1	2	6	8
R_k	9	7.1	4.1	2.7	1.9	2.4	3.9	6	7.9

Cuadro : Friedman ranking for Goal Radius

Test no Paramétricos (Friedman / Nemenyi)

Ranking Comfort

Graph \ Comfort	0	1	5	10	15	20
Graph 30x50	6	1	2	3	5	4
Graph 45x90	6	5	4	3	2	1
Graph 60x150	6	5	4	3	2	1
Graph 75x180	6	3	1	2	4	5
Graph 90x200	6	1	2	4	5	3
Graph 105x230	6	1	2	5	4	3
Graph 120x250	6	1	2	5	4	3
Graph 135x270	6	1	2	3	5	4
Graph 150x330	6	1	2	5	3	4
Graph 165x360	6	5	4	3	1	4
R_k	6	2.4	2.5	3.6	3.5	3.2

Cuadro : Friedman ranking for Comfort

Test no Paramétricos (Friedman / Nemenyi)

Radios de Meta con todos los valores

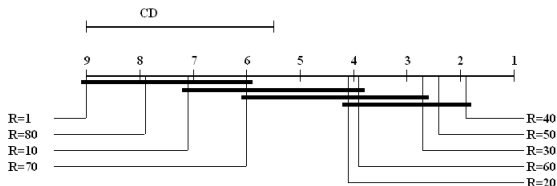


Figura : Test de Nemenyi con aceptacion del 90 %

Test no Paramétricos (Friedman / Nemenyi)

Radios de Meta con 5 valores

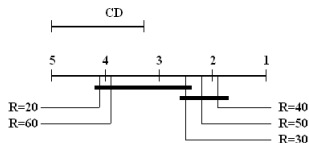


Figura : Test de Nemenyi con 5 valores y aceptación del 95 %

Test no Paramétricos (Friedman / Nemenyi)

Radios de Meta con 5 valores

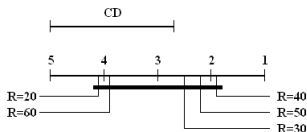


Figura : Test de Nemenyi con 5 valores y aceptación del 99%

Test no Paramétricos (Friedman / Nemenyi)

Confort con todos los valores

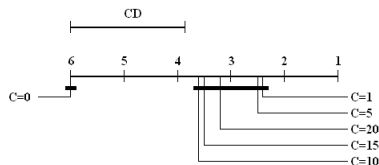


Figura : Test de Nemenyi con aceptación del 99 %

Test no Paramétricos (Friedman / Nemenyi)

Confort con 5 valores

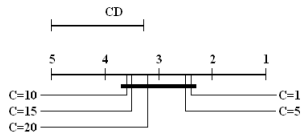


Figura : Test de Nemenyi con aceptación del 99 %

Outline

- 1 Introducción
- 2 Gravitational Swarm for Graph Coloring
- 3 Estudio sensibilidad de GSGC a sus parámetros
- 4 Algoritmos implementados**
- 5 Resultados comparativos
- 6 Conclusiones

Algoritmos implementados

- BackTracking (BT)
- Clique BackTracking (CBT)
- DSATUR (DS)
- Tabú Search (TS)
- Simulated Annealing (SA)
- Ant Colony Optimization (ACO)
- Particle Swarm Optimization (PSO)

BackTracking y CBT

- Se van coloreando secuencialmente los nodos del grafo.
- Si se llega a un estado en el que no es posible continuar el coloreado, se va al estado anterior y se modifica el color utilizado.
- Es exacto.

CBT: Hemos añadido una modificación en la que la ordenación la establece el “clique” más grande.

Clique: subgrafo completo de un grafo. Clique más grande es Cota mínima del número cromático.

BackTracking y CBT

Algorithm 3.1 Backtracking algorithm for graph coloring

Set initial values $\forall i \in V; W_i = 0; C(i) = 0$.

initialize $i = 1, c = 1$,

while $1 \leq i \leq N$

 set $C(i) = c$

 if $\forall j \in A_i, C(j) \neq C(i)$ /test coloring correctness/

 then $i \leftarrow i + 1$

 elseif $c < M$

 then $c \leftarrow c + 1$

 else $W_i = 0, C(i) = 0$,

 backtrack $i \leftarrow i - 1, c \leftarrow C(i) + 1, W_i \leftarrow W_i \cup C(i)$

 endelseif

endwhile

if $i = 0$ the algorithm has failed.

Algorithm 3.2 Clique initialization of BT graph coloring

1. Find the biggest clique BC and arrange the nodes of the clique first V_{BC}

 then the $V - V_{BC}$ nodes randomly.

2. Color the V_{BC} with the first $|V_{BC}|$ colors.

3. Apply the BT algorithm to

DSATUR (Brèlaz)

- Desarrollado por el matemático y político suizo Daniel Brèlaz.
- Consiste en ordenar los nodos del grafo por su grado de saturación e ir coloreando los nodos añadiendo colores cuando sea necesario.
- Es un algoritmo greedy, muy rápido, pero que no garantiza el número cromático.
- El grado de saturación para el algoritmo DSATUR es:
 - El número de aristas que salen de un nodo a otros nodos.
 - El número de vecinos con un color asignado.
- Los nodos se visitan una única vez.

DSATUR (Brèlaz)

Algorithm 3.3 DSATUR graph coloring algorithm

set $D_i = d_i$; $\mathcal{C}(i) = 0$; $\forall i \in V$; $V_1 = V$; $c^* = 1$

sort nodes in V_1 such that $D_i \geq D_{i+1}$

init $i = 1$

while $i \leq N$

 compute $R_i = \{\mathcal{C}(j); j \in A_i \wedge \mathcal{C}(j) > 0\}$

 if $D_i > |R_i|$ and $|R_i| > 0$

 then $D_i = |R_i|$

 sort nodes in V_i such that $D_i \geq D_{i+1}$

 elseif $|R_i| = 0$

 then $\mathcal{C}(i) = 1$ /any used color/

 elseif $|R_i| = c^*$

 then $c^* = c^* + 1$; $\mathcal{C}(i) = c^*$; $V_{i+1} = V_i - \{v_i\}$; $i = i + 1$;

 else $\mathcal{C}(i) = \min \{c^* - k; k \in R_i\}$; $V_{i+1} = V_i - \{v_i\}$; $i = i + 1$;

 endelseif

 endif

endfor

Tabú Search

La Búsqueda Tabú en un método de búsqueda local. Consiste en ir guardando soluciones en una lista de forma que se evita volver a considerar dicha solución.

$$Q(\mathcal{C}_t(i)) = |V - \{i | \forall j \in A_i; \mathcal{C}_t(j) \neq \mathcal{C}_t(i)\}|$$

Algorithm 3.4 Tabu Search for graph coloring

Generate an initial random solution $\mathcal{C}_t(i) \in \{1, \dots, M\}; \forall i \in V, t = 0.$

set the initial tabu list $L_t = \emptyset$

for $t = 1, \dots, \text{maxiter}$

Generate \mathcal{C}' a new solution from $\mathcal{C}_{t-1}(i)$ s.t. $\mathcal{C}' \notin L_{t-1}$

if $Q(\mathcal{C}_{t-1}(i)) \geq Q(\mathcal{C}'(i))$

then $\mathcal{C}_t(i) = \mathcal{C}'(i); L_t = L_{t-1} \cup \mathcal{C}_{t-1}(i)$

if $Q(\mathcal{C}_t(i)) = 0$ the search stops finding a correct coloring

endfor

Simulated Annealing

Inspirado en el proceso de recocido del acero y cerámicas, consiste en ir buscando soluciones al problema de forma que al evaluar una función de energía esta va decreciendo hasta que llega a 0. Las soluciones son aceptadas con una probabilidad mayor que un numero aleatorio entre $[0,1]$ de forma que el algoritmo consiga escapar de mínimos locales.

Algorithm 3.5 Simulated Annealing for graph coloring

Generate an initial random solution $C_t(i) \in \{1, \dots, M\}; \forall i \in V, t = 0.$

for $t = 0, \dots, \text{maxiter}$

Generate C' a new candidate solution from $C_{t-1}(i); \beta = \text{maxiter} - t$

if $e^{-\frac{1}{\beta}(Q(C'(i)) - Q(C_{t-1}(i)))} > p$

then $Q(C_t(i)) = Q(C'(i))$

else $Q(C_t(i)) = Q(C_{t-1}(i))$

if $Q(C_t(i)) = 0$ successful stopping of the algorithm

endfor

Ant Colony optimization (ACO)

Algorithm 3.6 Ant Colony Optimization algorithm for graph coloring

Generate N random colored ants $a = (c_a, i) \in \mathcal{A}_i$ for each node $i \in V$

for $t = 0, \dots, \text{maxiter}$

compute $\mathcal{C}_t(i) = \arg \max_c \{a | (c_a = c) \wedge (v_a = i)\}$

 / migrate minority ants/

for $i = 1, \dots, N$

for each $a \in \mathcal{A}_i$ s.t. $c_a \neq \mathcal{C}_t(i)$

if $\exists j \in \mathcal{A}_i$ s.t. $c_a = \mathcal{C}_t(j)$

then $v_a = j$

else $v_a \in \mathcal{A}_i$ picked randomly

endfor

 / break conflicting colorings/

for each $(i, j) \in E$ s.t. $\mathcal{C}_t(i) = \mathcal{C}_t(j)$

if $|\mathcal{A}_i| > |\mathcal{A}_j|$

then $c_a = i; \forall a \in \mathcal{A}_j$

else $c_a = j; \forall a \in \mathcal{A}_i$

endfor

 / solve depleted nodes/

for each node i s.t. $\mathcal{A}_i = \emptyset$

generate N random colored ants $a = (c_a, i)$

endfor

if $Q(\mathcal{C}_t(i)) = 0$ stop report successful coloring

endfor

Particle Swarm Optimization (PSO)

- Es una particularización de SI, en la que las partículas disponen de memoria para recordar la mejor solución local obtenida y la mejor solución global obtenida.
- Las partículas se mueven en el entorno de estas soluciones.
- Una partícula será un nodo del grafo.
- El atributo fundamental de las partículas es una distribución de probabilidades para los colores utilizados.

Particle Swarm Optimization (PSO)

Algorithm 3.7 Particle Swarm Optimization

Generate a initial state with uniform distributions per particle

$$S(0) = \left\{ s_i = \left\{ p_k = \frac{1}{M}; k = 1, \dots, M \right\}; i \in V \right\}.$$

initialize $S^g = S(0)$, $s_i^t = s_i(0)$, $q_i^t = q^g = \infty$

for $t = 1, \dots, \text{maxiter}$

 / evolve particles/

for each $i \in V$

$$s_i(t) = s_i(t-1) + \alpha_t s_i^t + \beta_t s_i^g + r_i(t)$$

endfor

 / compute graph color assignment/

for each $i \in V$

obtain $C_t(i)$ = sample of color probability distribution $s_i(t)$

if $q_i^t \geq Q(C_t(i))|_j$

then $q_i^t = Q(C_t(i))|_j$; $s_i^t = s_i(t)$

endfor

if $q^g \geq Q(C_t(i))$

then $q^g = Q(C_t(i))$; $S^g = S(t)$

if $Q(C_t(i)) = 0$ the algorithm stops successfully

endfor

Outline

- 1 Introducción
- 2 Gravitational Swarm for Graph Coloring
- 3 Estudio sensibilidad de GSGC a sus parámetros
- 4 Algoritmos implementados
- 5 **Resultados comparativos**
- 6 Conclusiones

Resultados comparativos

- Árboles y Bipartitos
- Grafos Planares
 - Usando teorema de Kuratowski
 - 4 coloreables por teoría de grafos
- Grafos 3-Coloreables
 - Usando artículo de Mizuno
 - Grafos “difíciles” demostrado por Mizuno
- Grafos KRG:
 - Pequeños
 - Grandes
- Grafos DIMACS
- Búsqueda del número cromático

Porcentajes de éxito en Árboles y Bipartitos

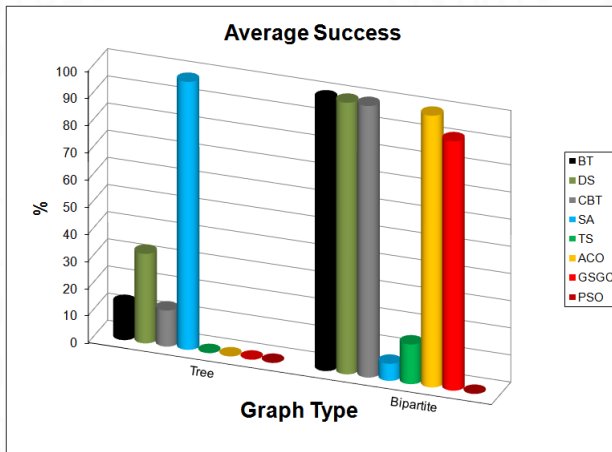


Figura : Porcentajes de éxito

Tiempo en Segundos en Árboles y Bipartitos

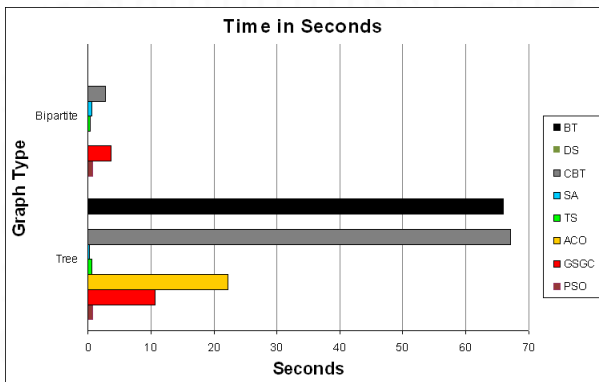


Figura : Tiempo en segundos

Porcentajes de éxito en Grafos Planares

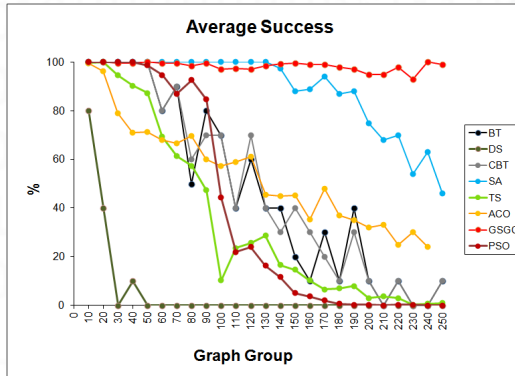


Figura : Porcentajes de éxito

Tiempo en Segundos en Grafos Planares

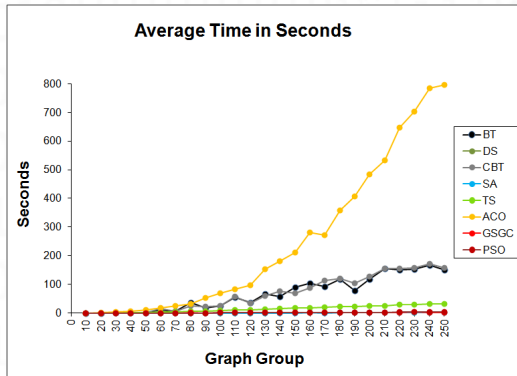


Figura : Tiempo en segundos

Porcentajes de éxito en Grafos 3-Coloreables

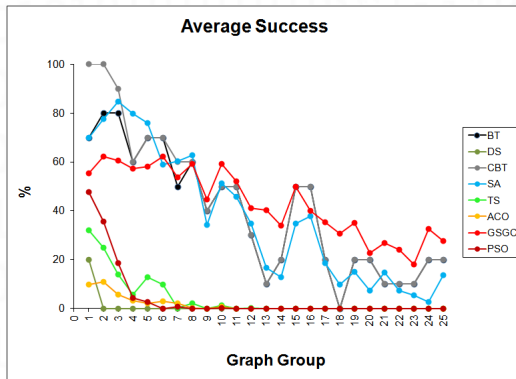


Figura : Porcentajes de éxito

Tiempo en Segundos en Grafos 3-Coloreables

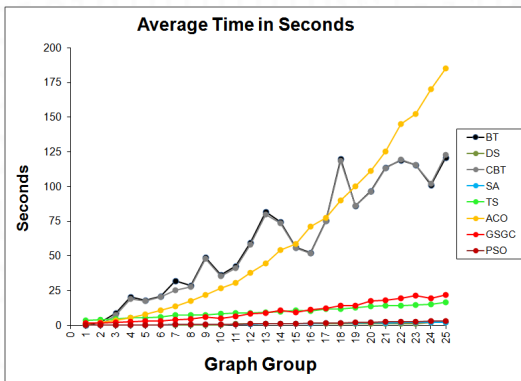


Figura : Tiempo en segundos

Grafos KRG

Porcentajes de éxito en Grafos KRG de 100 nodos y 300 aristas siendo 4,5,6,7-Coloreables

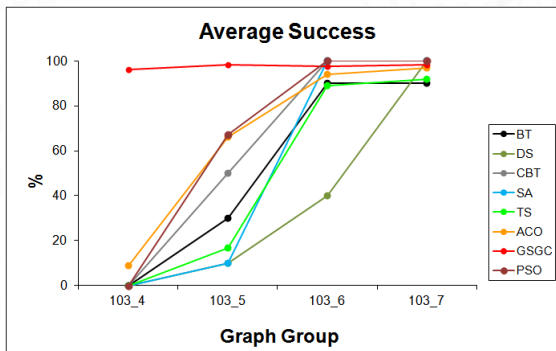


Figura : Porcentajes de éxito en KRG

Grafos KRG

Tiempo en Segundos en Grafos KRG de 100 nodos y 300 aristas siendo 4,5,6,7-Coloreables

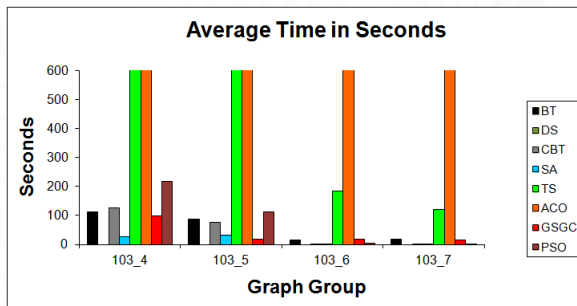


Figura : Tiempo en segundos

Grafos KRG

Porcentajes de éxito en Grafos KRG de 2000 nodos y 5400 aristas siendo 4,5,6,7- Coloreables

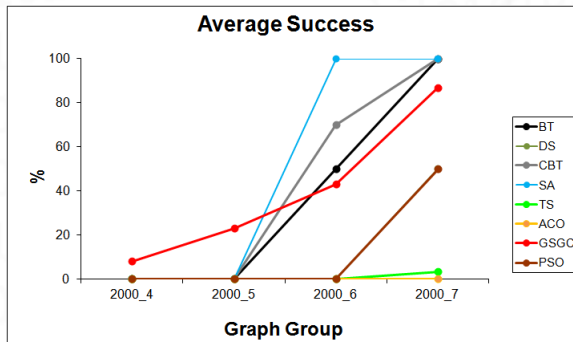


Figura : Porcentajes de éxito

Grafos KRG

Tiempo en Segundos en Grafos KRG de 2000 nodos y 5400 aristas siendo 4,5,6,7-Coloreables

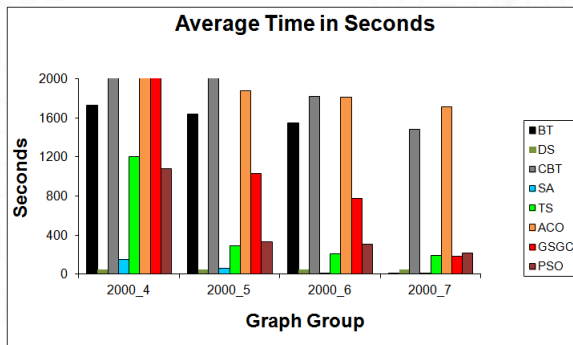


Figura : Tiempo en segundos

Grafos DIMACS

- Resultados de grafos de Stanford GraphBase (SGB)
- Resultados de grafos de Caramia (CAR)

Resultados SGB

Graph	#K	BT	DS	CBT	SA	TS	ACO	PSO	GS-GC
anna	11	0	100	0	3	17	0	0	96
david	11	0	0	100	3	0	0	0	93
homer	13	0	100	0	0	0	0	0	100
huck	11	100	100	100	100	80	0	100	90
jean	10	100	0	100	100	80	27	96	93
miles250	8	0	0	0	17	0	0	7	100
miles500	20	0	0	0	0	0	0	0	66
miles750	31	0	0	0	0	0	0	0	37
miles1000	42	0	0	0	0	0	0	0	100
miles1500	73	0	100	0	0	0	0	0	100
queen5_5	5	100	0	100	0	97	0	100	100
queen6_6	7	100	0	100	0	3	0	3	13
queen7_7	7	100	0	100	0	0	0	0	3
queen8_8	9	0	0	0	0	0	0	0	0
queen8_12	12	100	0	100	0	0	0	0	10
queen9_9	10	0	0	0	0	0	0	0	0

Figura : Resultado SGB

Resultados CAR

Graph	#K	BT	DS	CBT	SA	TS	ACO	PSO	GS-GC
myciel3	4	100	100	100	100	100	100	100	100
myciel4	5	100	100	100	100	100	100	100	100
myciel5	6	100	100	100	100	100	100	100	100
myciel6	7	100	100	100	100	100	100	100	100
myciel7	8	100	100	100	100	100	100	100	100
1-FullIns_3	3	100	100	100	100	78	93	100	100
1-FullIns_4	4	0	100	0	0	10	0	20	93
1-FullIns_5	5	0	100	0	0	0	0	0	90
2-FullIns_3	4	0	100	100	20	43	73	100	93
2-FullIns_4	5	0	100	0	0	3	0	0	93
2-FullIns_5	6	0	100	0	0	0	0	0	90
3-FullIns_3	5	0	100	0	3	37	0	100	87
3-FullIns_4	6	0	100	0	0	3	0	0	63
3-FullIns_5	7	0	100	0	0	0	0	0	87
4-FullIns_3	6	0	100	0	3	40	0	97	66
4-FullIns_4	7	0	100	0	0	0	0	0	66
4-FullIns_5	8	0	100	0	0	0	0	0	77
5-FullIns_3	7	0	100	0	0	20	0	57	97
5-FullIns_4	8	0	100	0	0	0	0	0	73

Figura : Resultados CAR

Búsqueda del número cromático

- Grafos aleatorios:
 - 100 nodos y 1000 aristas
 - 100 nodos y 2000 aristas
 - 100 nodos y 3000 aristas
 - 100 nodos y 4000 aristas
- Grafos de Leighton
- Grafos reales (programación de exámenes en universidades de USA)

Búsqueda del número cromático

Grafos aleatorios

Graphs100x1000	BT	DS	SA	TS	PSO	ACO	GS-GC
Random1	9	10	15(14)	11	11	17.1(17)	8.6(8)
Random2	10	10	15.1(15)	11	11	17(17)	9.2(9)
Random3	10	10	15(14)	14	11	17.2(17)	9.3(9)
Random4	9	11	15.1(14)	17	11	17.1(17)	9.1(9)
Random5	10	10	15(14)	11	11	17(17)	9.1(9)
Random6	11	11	14.9(14)	11	11	17(17)	9.2(9)
Random7	10	11	15(14)	11	10	17(17)	9.2(9)
Random8	10	11	15.1(14)	13	11	17.2(17)	9.1(9)
Random9	10	11	15(14)	15	10	17.3(17)	9.3(9)
Random10	11	11	15(14)	11	10	17.1(17)	9.2(9)

Figura : Resultados Random Graphs

Búsqueda del número cromático

Grafos de Leighton

Graph	#node	#Edges	Density	#K	DS	GS-GC
le450_5a	450	5714	0.056	5	12	8
le450_5b	450	5734	0.057	5	12	8
le450_5c	450	9803	0.097	5	11	9
le450_5d	450	9757	0.096	5	12	9
le450_15a	450	8168	0.081	15	19	18
le450_15b	450	8169	0.081	15	19	18
le450_15c	450	16680	0.165	15	26	20
le450_15d	450	16750	0.166	15	27	20
le450_25a	450	8260	0.082	25	26	26
le450_25b	450	8263	0.082	25	26	27
le450_25c	450	17343	0.171	25	31	30
le450_25d	450	17425	0.172	25	32	30

Figura : Resultados Leighton

Búsqueda del número cromático

Grafos reales

Graph	#K	TabuCol	PartCol	AntCol	HEA	HC	BT	DS	GSGC
sta-f-83	13	13.35	13	13.13	13	13	13	13	13
hec-s-92	17	17.22	17	17.04	17	17	19	21	18
kfu-s-93	19	20.76	19	19	19	19	19	21	20
yor-f-83	19	19.74	19	19.87	19.06	19	20	24	21
tre-s-92	20	20.58	20	20.04	20	20	23	25	23
car_f-92	27	39.92	32.48	30.04	28.5	27.96	27	34	36
car-s-91	28	39.1	30.2	29.23	29.04	29.1	28	37	37
pur-s-93	33	50.7	45.48	33.47	33.7	33.87	33	38	44

Figura : Resultados Grafos Reales

Outline

- 1 Introducción
- 2 Gravitational Swarm for Graph Coloring
- 3 Estudio sensibilidad de GSGC a sus parámetros
- 4 Algoritmos implementados
- 5 Resultados comparativos
- 6 Conclusiones

Conclusiones

La aproximación seguida en esta Tesis está inspirada en la naturaleza de dos formas:

- 1 Usando la metáfora de Swarm Intelligence que ha producido novedosas soluciones computacionales como ACO y PSO.
- 2 Usando una metáfora de la interacción gravitatoria y electromagnética para mapear el GCP como un Swarm.

Conclusiones II

Hemos tenido éxito proponiendo un algoritmo eficiente, inspirado en la naturaleza, que soluciona el GCP. Hemos probado el valor de nuestro algoritmo mediante dos líneas de trabajo separadas:

- 1 Hemos sido capaces de establecer formalmente algunas propiedades deseadas de convergencia que dan prueba formal de ello. Hemos demostrado que, si el sistema converge a un estado estacionario, este estado es una solución del GCP.
- 2 Hemos sido capaces de llevar a cabo experimentos computacionales a través de una extensa colección de grafos de referencia, en comparación con otros algoritmos del estado del arte. Los rendimientos obtenidos son competitivos e incluso mejoran otros de la literatura.

Trabajo Futuro

- Trabajo teórico: Demostrar la convergencia dinámica del sistema.
- Verificación Computacional: es deseable continuar trabajando en nuevos experimentos sobre grafos.
- No estacionario: deseamos estudiar la convergencia en el caso no estacionario.
- Aplicaciones: deseamos identificar posibles aplicaciones de esta Tesis al ámbito de la vida real. En estos momentos estamos trabajando en dirección a la identificación y seguimiento de comunidades en redes sociales.

Gracias

Muchas Gracias por su atención.

