

**CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN
SUPERIOR DE ENSENADA, BAJA CALIFORNIA**



**PROGRAMA DE POSGRADO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN**

Problema de asignación de guardaespaldas multiclase

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de
Maestro en Ciencias

Presenta:

Lino Alberto Rodríguez Coyahuitl

Ensenada, Baja California, México

2014

Tesis defendida por

Lino Alberto Rodríguez Coyahuitl

y aprobada por el siguiente comité

Dr. José Alberto Fernández Zepeda
Director del Comité

Dr. Carlos Alberto Brizuela Rodríguez
Miembro del Comité

Dr. Israel Rocha Mendoza
Miembro del Comité

Dra. Ana Isabel Martínez García
*Coordinador del Programa de
Posgrado en Ciencias de la Computación*

Dr. Jesús Favela Vara
Director de Estudios de Posgrado

Septiembre, 2014

Resumen de la tesis que presenta **Lino Alberto Rodríguez Coayahuitl** como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

Problema de asignación de guardaespaldas multiclase

Resumen elaborado por:

Lino Alberto Rodríguez Coayahuitl

En este trabajo se presenta el problema de asignación de guardaespaldas multiclase (PAGM). Este problema es una versión más general del problema de asignación de guardaespaldas (PAG). El PAG lo propuso Fajardo-Delgado *et al.* (2013), y consiste en construir un árbol de esparcimiento en grafos conectados donde se presentan tres tipos de vértices: blancos, negros y la raíz. Cada tipo de vértice tiene un objetivo distinto: los vértices blancos buscan minimizar su distancia a la raíz dentro del árbol de esparcimiento, mientras que los vértices negros, buscan maximizarla. Una de las limitaciones del PAG es que sólo contempla dos clases de vértices, por lo que el objetivo de este trabajo es extender el problema del PAG para que contemple más clases de vértices, no solo aquellos que buscan estar en los extremos del árbol de esparcimiento, sino también vértices que busquen estar a cualquier distancia específica de la raíz.

Una solución del PAGM es un árbol en el que se cumple una condición de equilibrio y se maximiza el bienestar social. El PAGM se estudia desde la teoría de juegos, y se propone un enfoque cooperativo y uno no cooperativo para encontrar soluciones al PAGM. Se propone un algoritmo centralizado, CBAPM, y un algoritmo distribuido, DBAPM, que utilizan uno de estos dos enfoques para resolver casos del PAGM. Se analiza de manera rigurosa la convergencia al equilibrio de estos algoritmos bajo cualquiera de los dos enfoques, así como su tiempo de ejecución, en función del tamaño del caso de entrada.

Se propone un algoritmo denominado CBAPM mixto (M-CBAPM), que combina de manera secuencial los enfoques no cooperativo y cooperativo, con el objetivo de encontrar mejores soluciones de las que se pueden obtener si se utiliza cualquiera de los dos enfoques de manera individual. Se proponen tres variantes de este algoritmo: M-CBAPM, CBAPM mixto a partir del mejor bienestar social (M-CBAPM-BSW) y CBAPM mixto doble (DM-CBAPM).

Por último, para llevar a cabo pruebas experimentales de los algoritmos propuestos, se propuso un conjunto de casos específicos del PAGM. Se utilizó la metodología propuesta por Zatarain Aceves (2011) para el diseño de este conjunto de casos específicos. Se comparó el enfoque cooperativo contra el no cooperativo, para determinar bajo qué condiciones uno es mejor que el otro; y por último se compararon estos dos contra el enfoque del algoritmo M-CBAPM.

Palabras Clave: Análisis de algoritmos, teoría de juegos, sistemas distribuidos, problema de asignación de guardaespaldas.

Abstract of the thesis presented by **Lino Alberto Rodríguez Coayahuitl** as a partial requirement to obtain the Master of Science degree in Master in Sciences in Computer Science.

Multiclass Bodyguard Allocation Problem

Abstract by:

Lino Alberto Rodríguez Coayahuitl

In this thesis, we extended the bodyguard allocation problem (BAP) into the multiclass bodyguard allocation problem (MBAP). The BAP was originally proposed in (Fajardo-Delgado *et al.*, 2013), and its objective is to build a spanning tree in a connected graph in which there are three types of vertices: white, black and the root. Each type of vertex has a different objective: the white vertices seek to minimize their distance to the root while black vertices seek to maximize it. A limitation of the BAP is that it only considers two classes of vertex, those that try to get as close as possible to the root, and those that try to get as far as possible from it. The objective of this work is to extend the model of the BAP, so it considers vertices that try to stay at specific distances from the root. We call this extended problem the MBAP.

A solution to the MBAP is a spanning tree that satisfies a condition of equilibrium and maximizes the social welfare. We study the MBAP from a game theory perspective. We proposed a cooperative approach and a non-cooperative approach to solve the MBAP. We proposed the CBAPM, a centralized algorithm, and the DBAPM, a distributed algorithm, (both based on the algorithms originally proposed by Fajardo-Delgado *et al.* (2013) for the BAP), that use any of the two approaches to solve the MBAP. We analysed the convergence to a solution of the proposed algorithms under each approach. We also analysed the execution time of the algorithms according to the size of the input.

We also proposed a mixed approach algorithm, M-CBAPM, that combines both non-cooperative and cooperative approaches in a sequential manner. The objective of the M-CBAPM is to reach solutions with higher social welfare than those algorithms working independently. We proposed three variants of the mixed algorithm: M-CBAPM, M-CBAPM using best non-cooperative social welfare (M-BAPM-BSW), and double round M-CBAPM (DM-CBAPM).

In order to carry experimental simulations of the proposed algorithms, we proposed a collection of MBAP instances. We used the methodology proposed by Zatarain Aceves (2011) for the design of such instances. We compared the performances of the cooperative, non-cooperative and mixed approaches, to determine under what conditions each approach surpasses the others.

Keywords: Analysis of algorithms, game theory, distributed systems, Bodyguard Allocation Problem

Dedicatoria

***A la ciudad de Ensenada, que
fue mi casa durante dos años***

Agradecimientos

A mi familia, por todo el apoyo y comprensión durante esta etapa que me permitieron alcanzar uno de mis objetivos.

A mi director de tesis, el Dr. Alberto Fernández, por haberme invitado a formar parte de su equipo de trabajo, y guiarme durante el desarrollo de esta tesis.

A *Hector Zatarain*, por su invaluable apoyo durante la realización de este trabajo de tesis.

A los miembros del comité de tesis, el *Dr. Carlos Brizuela* y el *Dr. Israel Rocha*, por sus valiosas aportaciones durante las revisiones de avance de tesis.

A mis compañeros del laboratorio de algoritmos, *Ana, Alex, Ismael, Julio y Rosa* por hacer de los días más difíciles, los mejores.

Al resto de mis compañeros de generación, *Armando, Alejandro, Carlos, Ema, Franceli, Julia, Miguel, Netzahualcóyotl, Ramón, Raúl, Refu, Saúl* por todos los momentos compartidos.

A todos los investigadores, estudiantes y personal del departamento de ciencias de la computación por su enseñanza académica.

Al CONACyT y proyecto de investigación por su apoyo económico.

Tabla de contenido

	Página
Resumen en español	iii
Resumen en inglés	iv
Dedicatoria	v
Agradecimientos	vi
Lista de figuras	x
Lista de tablas	xi
1. Introducción	1
1.1. Investigación previa	3
1.1.1. Problema de asignación de guardaespaldas	3
1.1.2. Uso de la teoría de juegos para modelar el PAG	7
1.1.3. Un algoritmo evolutivo para el PAG	11
1.2. Objetivo de la investigación	13
1.3. Contribución al conocimiento	14
1.4. Organización de la tesis	14
2. Problema de asignación de guardaespaldas multiclase	16
2.1. Definición	16
2.2. Teoría de juegos y el PAGM	20
2.2.1. Contribución de un vértice al bienestar social	20
2.2.2. Función de ganancia	22
2.2.3. Enfoque cooperativo	23
2.2.4. Enfoque no cooperativo	24
3. Algoritmo CBAPM	26
3.1. Análisis del algoritmo CBAPM	26
3.1.1. Procedimiento CREAR-ARBOL()	27
3.1.2. Procedimiento ETIQUETADO-INICIAL()	27
3.1.3. Procedimiento GENERAR-FUNCIONES-GANANCIA()	28
3.1.4. Procedimiento CALENDARIZADOR()	28
3.1.5. Procedimiento MOVIMIENTO-OPTIMIZACION()	30
3.1.6. Procedimiento ACTUALIZACION()	30
3.2. Demostración de convergencia	30
3.2.1. Enfoque cooperativo	31
3.2.2. Enfoque no cooperativo	31
3.2.2.1. Grafo completo	34
3.3. Algoritmo distribuido	38
3.3.1. Descripción del algoritmo DBAP	38
3.3.1.1. Fase TREE-CONSTRUCTION	38
3.3.1.2. Fase CONVERGECAST	39
3.3.1.3. Fase TOKEN-PASSING	39
3.3.1.4. Fase OPTIMIZATION	39

Tabla de contenido (continuación)

3.3.2.	Análisis del algoritmo DBAPM	40
4.	Algoritmo mixto	42
4.1.	Algoritmo CBAPM Mixto	42
4.1.1.	Procedimiento CREAR-ARBOL()	44
4.1.2.	Procedimiento ETIQUETADO-INICIAL	44
4.1.3.	Procedimiento GENERAR-FUNCIONES-GANANCIA()	44
4.1.4.	Procedimiento CALENDARIZADOR()	44
4.1.5.	Procedimiento MOVIMIENTO-OPTIMIZACION()	44
4.1.6.	Procedimiento ACTUALIZACION()	44
4.1.7.	Procedimiento CAMBIAR-FUNCIONES-GANANCIA()	44
4.2.	Algoritmo CBAPM Mixto a partir del mejor bienestar social	45
4.2.1.	Procedimiento CREAR-ARBOL()	45
4.2.2.	Procedimiento ETIQUETADO-INICIAL	47
4.2.3.	Procedimiento GENERAR-FUNCIONES-GANANCIA()	47
4.2.4.	Procedimiento COPIAR-ARBOL()	47
4.2.5.	Procedimiento CALENDARIZADOR()	47
4.2.6.	Procedimiento MOVIMIENTO-OPTIMIZACION()	47
4.2.7.	Procedimiento ACTUALIZACION()	47
4.2.8.	Procedimiento CAMBIAR-FUNCIONES-GANANCIA()	47
4.3.	Algoritmo CBAPM Mixto doble	48
5.	Resultados experimentales	50
5.1.	Generación de casos específicos del PAGM	50
5.1.1.	Distribuciones de clases	50
5.1.2.	Generación de grafos aleatorios	51
5.1.2.1.	Modelo Barabási-Albert	52
5.1.2.2.	Modelo Erdős-Rényi	53
5.1.2.3.	Modelo evolutivo estocástico	53
5.2.	Metodología	54
5.3.	Resultados	57
5.3.1.	CBAPM	57
5.3.1.1.	PoS aproximado	57
5.3.1.2.	PoA aproximado	60
5.3.1.3.	Cantidad de movimientos	62
5.3.2.	DBAPM	64
5.3.2.1.	PoS aproximado	64
5.3.2.2.	PoA aproximado	66
5.3.2.3.	Cantidad de movimientos	68
5.3.3.	Algoritmo mixto	68
5.3.3.1.	PoS aproximado	68
5.3.3.2.	PoA aproximado	72
5.3.3.3.	Cantidad de movimientos cooperativos	74
5.3.4.	Algoritmo M-CBAPM-BSW y algoritmo DM-CBAPM	74

Tabla de contenido (continuación)

5.3.4.1. PoS aproximado	74
5.3.4.2. PoA aproximado	78
6. Conclusiones y trabajo a futuro	81
6.1. Resumen	81
6.2. Conclusiones y conjeturas	82
6.3. Trabajo a futuro	85
Lista de referencias	87

Lista de figuras

Figura		Página
1.	PoS para FCC-BAP, EA-BAP, CBAP	12
2.	PoA para FCC-BAP, EA-BAP, CBAP	13
3.	Caso específico del PAGM	18
4.	Enfoque cooperativo contra no cooperativo	22
5.	Movimiento no cooperativo	32
6.	Ejemplo ciclo no cooperativo	33
7.	Ejemplo conjunto inamovible 1	35
8.	Ejemplo conjunto inamovible 2	36
9.	Algoritmo DBAPM	40
10.	Algoritmo M-CBAPM	43
11.	Algoritmo M-CBAPM-BSW	46
12.	Algoritmo DM-CBAPM	49
13.	Grafo Barabási-Albert	53
14.	Grafo Erdős-Rényi	54
15.	Grafo evolutivo estocástico	55
16.	PoS para el CBAPM	58
17.	PoA para CBAPM	61
18.	Movimientos CBAPM	63
19.	PoS para DBAPM	65
20.	PoA para DBAPM	67
21.	Movimientos DBAPM	69
22.	PoS para M-CBAPM	71
23.	PoA para M-CBAPM	73
24.	Movimientos M-CBAPM	75
25.	PoS para M-CBAPM, M-CBAPM-BSW y DM-CBAPM	76
26.	PoA para M-CBAPM, M-CBAPM-BSW y DM-CBAPM	79

Lista de tablas

Tabla		Página
1.	Construcción de función de contribución	23
2.	Óptimos distintas distribuciones	56
3.	PoS para el CBAPM	59
4.	PoA para CBAPM	62
5.	PoS para DBAPM	64
6.	PoA para DBAPM	66
7.	PoS para M-CBAPM	70
8.	PoA para M-CBAPM	72
9.	PoS para M-CBAPM, M-CBAPM-BSW y DM-CBAPM	77
10.	PoA para M-CBAPM, M-CBAPM-BSW y DM-CBAPM	78

Capítulo 1. Introducción

La teoría de juegos es el estudio de la toma de decisiones estratégica y trata con problemas que involucran entidades racionales que pueden tener conflicto de intereses, o bien, que pueden beneficiarse si hay cooperación entre éstas. La teoría de juegos es una área de las matemáticas, por lo general considerada como una rama de la economía, que tiene aplicaciones en áreas como la biología, los negocios, el ámbito militar y la computación.

Los denominados *juegos* en la teoría de juegos consisten en modelos matemáticos bien definidos que especifican los siguientes elementos: los jugadores, la información y acciones disponibles para cada jugador, y las ganancias para cada posible escenario que pueda resultar como producto de la interacción entre los jugadores.

Lo que se busca en el estudio de la teoría de juegos es encontrar estrategias que lleven a los juegos a puntos de equilibrio, es decir, divisar estrategias para cada jugador de modo que ninguno de los jugadores se pueda beneficiar unilateralmente.

En las ciencias de la computación muchas veces se puede ver un proceso distribuido como una colección de sistemas autónomos que trabajan en conjunto para alcanzar una meta en común. Un proceso distribuido es un ejemplo de un juego del tipo cooperativo. Por otro lado, en sistemas masivamente distribuidos, como lo es la Internet, pueden existir entidades con objetivos contradictorios, por lo que se puede ver como un juego del tipo no cooperativo.

La teoría de juegos permite analizar varios problemas de cómputo distribuido en los que se presenta conflicto de intereses, como por ejemplo ruteo de Internet (Griffin *et al.* (2002), Roughgarden (2005)), balanceo de carga (Koutsoupias y Papadimitriou (2009), Grosu y Chronopoulos (2004)), servicios punto a punto (Feldman *et al.* (2004), Moscibroda *et al.* (2006)) y redes inalámbricas de sensores (Machado y Tekinay (2008), Charilas y Panagopoulos (2010)). Algunos de los estudios más amplios que relacionan la teoría de juegos con los sistemas distribuidos son (Nisan *et al.*, 2007) y (Halpern, 2007).

En (Fajardo-Delgado *et al.*, 2013) se presentó el juego del *Problema de Asignación*

de *Guardaespaldas* (PAG), el cual se explica de manera informal mediante la siguiente analogía: En una casa con n habitaciones conectadas entre ellas mediante m pasillos, existe una persona denominada *cliente*, a la que $n - 1$ guardaespaldas le brindan protección. Cada guardaespalda puede estar en una habitación y cuidar un pasillo. Se clausuran los pasillos que no protejan los guardaespaldas; pero siempre debe existir una ruta entre el cliente y cualquiera de los guardaespaldas. Existen dos tipos de guardaespaldas: expertos y novatos. Los guardaespaldas expertos buscan estar lo más cerca posible del cliente, mientras que los guardaespaldas novatos, lo más lejos posible. El conflicto de interés surge cuando un guardaespalda experto, al buscar acercarse al cliente, acerca a todos los guardaespaldas novatos que están detrás de él; y de manera análoga, cuando un guardaespalda novato, al alejarse del cliente, aleja a los guardaespaldas expertos que están detrás de él.

Matemáticamente, el PAG se representa mediante un grafo, en donde cada vértice representa una habitación con un guardaespalda o el cliente, y las aristas representan a los pasillos que conectan los cuartos. De esta manera existen tres tipos de vértices: blancos (negros) que representan a los guardaespaldas expertos (novatos); y un vértice especial, denominado raíz, que representa al cliente, de cual los vértices blancos (negros) se quieren acercar (alejar).

La solución al PAG se representa por un árbol de esparcimiento enraizado en el vértice que representa al cliente. Esta solución significa que al grafo original se le han removido algunas aristas (equivalente en la analogía a clausurar pasillos) de manera que sólo quede una arista por vértice blanco o negro (un pasillo asignado a cada guardaespalda) y además exista comunicación entre todos los cuartos.

Desde el enfoque de teoría de juegos, se determina que los guardaespaldas son los jugadores, que tienen posibilidad de alejarse o acercarse al cliente (el cual no participa en el juego, es decir, es un jugador pasivo), y el resultado del juego es una función de *bienestar social*, que es la suma de la *utilidad individual* de cada uno de los guardaespaldas. La *utilidad individual* de cada guardaespalda indica su grado de satisfacción con su posición actual. Así, la función de *utilidad individual* es diferente para cada tipo de guardaespalda, tomando en cuenta que unos buscan alejarse del vértice raíz, mientras que

otros buscan lo contrario.

El equilibrio en el PAG depende del enfoque que se utilice. Fajardo-Delgado *et al.* (2013) propone los enfoques completamente cooperativo y completamente no cooperativo. Bajo el enfoque completamente cooperativo (no cooperativo), el PAG alcanza el equilibrio cuando ningún guardaespaldas puede realizar un movimiento que mejore el bienestar social (su utilidad individual).

Un concepto que no es obvio a simple vista, pero que es de crucial importancia, es el hecho de que el que un caso específico del PAG haya alcanzado el equilibrio, no significa que no existan otras configuraciones en equilibrio con mayor bienestar social. Desde el punto de vista de optimización, se podría entender un equilibrio en el PAG como un óptimo local.

1.1. Investigación previa

En esta sección se describen los trabajos realizados por Fajardo-Delgado *et al.* (2013), Brubeck Salcedo (2011) y Zatarain Aceves (2011), sobre el PAG. Otros modelos similares al PAG, en el aspecto que consideran dos o más clases de procesos con objetivos individuales diferentes, se pueden encontrar en (Dasgupta *et al.*, 2006) y (Cohen *et al.*, 2008)

1.1.1. Problema de asignación de guardaespaldas

En sistemas computacionales distribuidos donde pueda existir un conflicto de interés, es importante encontrar una configuración donde ningún proceso se pueda beneficiar de manera unilateral. Esto quiere decir, que ningún proceso pueda realizar acciones que sólo lo beneficien a él. Esta configuración se le denomina *equilibrio* (Nash *et al.*, 1950). Cuando no existe un equilibrio único, se define el *bienestar social* como la suma de los grados de satisfacción individual de todos los procesos del sistema, para comparar la calidad de un equilibrio con respecto a otro.

La motivación que llevó al planteamiento y estudio del PAG es la siguiente:

- En primer lugar, el estudio de este problema contribuye a un mejor entendimiento

del comportamiento de procesos con objetivos contrarios en sistemas distriuidos.

- Se estudia la manera en cómo diferentes niveles de cooperación se traducen en diferentes grados de bienestar social, con la expectativa de que un mayor grado de cooperación entre los procesos conlleve a un mayor bienestar social.
- Se estudian y comparan los tiempos de ejecución de los algoritmos cuando los procesos cooperan y cuando no. El sentido común dictaría que un sistema cooperativo requiere de un mayor grado de coordinación, lo que representaría mayor tiempo de ejecución, pero sorprendentemente los resultados encontrados demuestran lo contrario.

El PAG, modelado como un problema de grafos, se define de la siguiente manera: sea la entrada un grafo conectado, no dirigido $G = (V_G, E_G)$, se divide el conjunto de vértices en una partición de tres subconjuntos V_1, V_2, V_3 . El subconjunto V_1 (V_2) representa los nodos blancos (negros), i.e. los que buscan minimizar (maximizar) su distancia a la raíz. El subconjunto V_3 está compuesto por un único elemento r , que es la raíz.

Una solución al PAG se representa por un árbol de esparcimiento $T = (V_T, E_T)$, enraizado en r , tal que $V_T = V_G, E_T \subseteq E_G$, y donde T tiene una condición de equilibrio.

El objetivo del PAG es encontrar un árbol T que maximice la función de bienestar social $g(T)$, que está dada por la Ecuación 1.

$$g(T) = \sum_{\forall u \in V_1} f_1(u) + \sum_{\forall u \in V_2} f_2(u). \quad (1)$$

La función $g(T)$ es la suma de las utilidades individuales de todos los vértices de V_1 y V_2 . Las funciones $f_1 : V_1 \rightarrow \mathbb{Z}$ y $f_2 : V_2 \rightarrow \mathbb{Z}$ denotan la utilidad individual de cada uno de los vértices blancos y negros respectivamente. Se definen las funciones f_1 y f_2 como se muestra en las Ecuaciones 2 y 3, donde d_u^T es la distancia de u a la raíz en el árbol, medida en aristas.

$$f_1(u) = n - d_u^T, \forall u \in V_1; \quad (2)$$

$$f_2(u) = d_u^T, \forall u \in V_2; \quad (3)$$

Dentro del marco de trabajo de teoría de juegos, el PAG es un juego \mathcal{G} que se compone de la tupla $(\Lambda, \{V_1 \cup V_2\}, S_u, \pi_u)$, donde

- Λ representa a un caso específico del PAG, i.e. un grafo $G = (V_G, E_G)$ junto con su partición de vértices.
- $\{V_1 \cup V_2\}$ es el conjunto de vértices que representan a los jugadores racionales del juego. El vértice raíz r es un jugador pasivo que no toma acción alguna.
- S_u es el conjunto de posibles estrategias para cada jugador. Esto es, alejarse o acercarse a la raíz.
- π_u es una función de evaluación que calcula las *preferencias* de cualquier jugador.

Se supone que cada caso específico Λ de \mathcal{G} inicia con un árbol de esparcimiento arbitrario T sobre el grafo G y enraizado en r .

Se denota como M_u a un movimiento que hace u . Este movimiento se realiza por medio de un cambio en el apuntador de $P_u = v$ a $P_u = v'$, como consecuencia de haber escogido una estrategia de S_u . El vértice v' solo puede pertenecer al conjunto de vecinos de u en G , que no sean sus descendientes en T , o de lo contrario el árbol se desconectaría.

Al estudiarse como un juego, el PAG se puede solucionar bajo enfoques completamente cooperativos o no cooperativos. Cuando todo vértice u sigue un enfoque completamente cooperativo (no cooperativo), u busca incrementar el bienestar social (utilidad individual) en lugar de la utilidad individual (bienestar social).

Bajo el enfoque completamente cooperativo, cada vértice u utiliza un sistema de votación para decidir qué estrategia (alejarse o acercarse) es la más conveniente para la mayoría de los vértices descendientes en su subárbol.

Se define como un *movimiento de optimización* a cualquier movimiento de un vértice u que genera el máximo posible valor de π_u . Durante el análisis del PAG, se demostró que bajo el enfoque completamente cooperativo, todos los movimientos de optimización incrementan, al menos en una unidad, el valor del bienestar social $g(T)$.

Bajo un enfoque completamente no cooperativo, cada vértice toma decisiones basándose únicamente en el voto emitido por él mismo.

Un equilibrio en el PAG significa que cada guardaespaldas no puede mejorar, al menos desde su propia perspectiva, la protección que proporciona al cliente. Es importante notar que el equilibrio depende del enfoque del problema, sea cooperativo o no cooperativo.

Para llevar a cabo la comparación de la calidad de los equilibrios se utilizan el *precio de estabilidad* (PoS) (Schulz y Moses, 2003), (Anshelevich *et al.*, 2004) y el *precio de anarquía* (PoA) (Koutsoupias y Papadimitriou, 2009) como medidas de ineficiencia para indicar qué tanto se aproximan los valores de bienestar social al óptimo. Dado que el PAG es un problema de maximización, el PoS (PoA) es la proporción del valor del equilibrio con mejor (peor) bienestar social entre el valor óptimo para un caso específico del PAG. Se conjetura que para el PAG no es posible encontrar el bienestar social óptimo en tiempo polinomial a menos que $P = NP$.

En (Fajardo-Delgado *et al.*, 2013) se desarrolló el algoritmo CBAP para la resolución del PAG. Se trata de un algoritmo centralizado que puede funcionar bajo esquemas completamente cooperativos o no-cooperativos. La entrada del CBAP es un caso específico del PAG y la salida es un árbol de esparcimiento que cumple con la condición de equilibrio pero que no necesariamente maximiza la función de bienestar social. El CBAP se muestra en el Algoritmo 1.

En (Fajardo-Delgado *et al.*, 2013) se demuestra que el bienestar social del PAG es a lo más $O(n^2)$. Por lo tanto, bajo el enfoque completamente cooperativo, al CBAP le tomará a lo más $O(n^2)$ movimientos de optimización para converger al equilibrio. El tiempo total de ejecución del CBAP, bajo la estrategia completamente cooperativa, es de $O(n^3 + mn^2)$ unidades de tiempo (u.t.).

Algoritmo 1 ALGORITMO-CBAP(G)

```

1:  $T \leftarrow \text{RANDOMIZED-SPANNING-TREE}(G, r)$ 
2:  $T \leftarrow \text{INITIAL-LABELING}(T)$ 
3:  $T \leftarrow \text{VOTE-COLLECTING}(T)$ 
4: mientras  $\exists u$  tal que  $u \leftarrow \text{SCHEDULER}(G, T)$  hacer
5:    $P_u^{old} \leftarrow P_u$ 
6:    $T' \leftarrow \text{OPTIMIZATION-MOVE}(G, T, u)$ 
7:    $T \leftarrow \text{UPDATING}(T', u, P_u^{old})$ 
8: fin mientras

```

En (Fajardo-Delgado *et al.*, 2013) también se demuestra que bajo el enfoque completamente no cooperativo, el algoritmo CBAP toma $O(n^4 \ln n + mn^3 \ln n)$ u.t. con *alta probabilidad* para alcanzar el equilibrio.

En (Fajardo-Delgado *et al.*, 2013) también se introduce otro enfoque denominado *semi-cooperativo*. Se dice que un vértice tiene un nivel de cooperación k , donde k es la distancia (profundidad) en su subárbol hasta donde considerará votos de sus descendientes. Se demostró que bajo este enfoque, el CBAP no siempre converge al equilibrio.

Para comparar los enfoques completamente cooperativos y no cooperativos, en (Fajardo-Delgado *et al.*, 2013) se llevaron a cabo simulaciones experimentales. Se generaron un total de 450 casos específicos mediante tres modelos diferentes de grafos aleatorios. Los resultados experimentales muestran que, en general, la estrategia cooperativa es mejor que la no cooperativa, salvo para grafos aleatorios dispersos.

1.1.2. Uso de la teoría de juegos para modelar el PAG

En (Brubeck Salcedo, 2011) se introducen diferentes modificaciones al juego del PAG y al algoritmo del CBAP para obtener mejores valores de bienestar social. A continuación se describen brevemente algunas de ellas.

Variantes de estrategias de jugadores

Estrategia voraz. La estrategia que originalmente se propone en (Fajardo-Delgado *et al.*, 2013) consiste en que cada vértice siempre busca maximizar o minimizar su distancia a la raíz. Esto se lleva a cabo mediante la función de evaluación π_u , con la que cada vértice u busca su vecino no descendiente que le permita realizar la conexión que

más lo aleje o acerque a la raíz, según sea su preferencia.

Brubeck Salcedo (2011) propone tres nuevas estrategias:

Estrategia ε -voraz. Consiste en generar dos conjuntos de vértices, Z_u y Y_u . Z_u son los elementos que maximizan la distancia (ya sea acercarse o alejarse) y Y_u son los elementos que solamente la incrementan (o decrementan, según sea el caso). Se escoge un elemento de Z_u con probabilidad $\frac{\varepsilon}{|Z_u|}$ o un elemento de Y_u con probabilidad $\frac{1-\varepsilon}{|Y_u|}$. Se demuestra que esta estrategia converge al equilibrio. Adicionalmente cabe mencionar que cuando $\varepsilon = 1$, la estrategia es exactamente igual a la estrategia voraz.

Estrategia Anti-voraz. Selecciona al elemento que incremente al mínimo π_u , i.e. que genere la ganancia mínima en el bienestar social o la utilidad individual.

Estrategia IA-Voraz. La estrategia IA-Voraz sigue una filosofía similar a la técnica heurística del *templado simulado*. Inicialmente todos los vértices pueden realizar conexiones con cualquiera de sus vecinos mientras no sean descendientes (para no desconectar el árbol), incluso con vértices que decrementan el bienestar social y la utilidad individual, pero conforme avanzan los turnos la probabilidad de escoger vértices que no maximicen estas funciones se va decrementando, hasta llegar a cero, para que el algoritmo finalmente pueda converger al equilibrio.

La idea detrás de estas variantes propuestas, es hacer que el CBAP realice una mayor cantidad y variedad de movimientos de optimización que los realizados con la estrategia voraz, con la esperanza de que eso conduzca a equilibrios más cercanos al óptimo.

Función jugador

La función jugador es la que se encarga de dar turno a los vértices para realizar sus movimientos. En el algoritmo CBAP, el calendarizador representa a la función jugador. El calendarizador original tiene dos políticas de selección: uniforme y voraz. En (Brubeck Salcedo, 2011) se propone una política adicional denominada *anti-voraz*, diseñada para usarse en combinación con la estrategia anti-voraz de los jugadores, para maximizar aún más el número y variedad de movimientos de optimización.

Técnicas de especulación

Brubeck Salcedo (2011) también propone dos modificaciones más al CBAP, con el fin de darle la capacidad de buscar mejores equilibrios que el primero que se encuentre. Se recuerda que un equilibrio representa un óptimo local en el PAG, es decir, que pueden existir equilibrios con mayor o menor bienestar social. Estas modificaciones se denominan técnicas de especulación.

Movimiento de especulación. Es un movimiento en el que un vértice sacrifica su utilidad individual para que otro vértice pueda mejorar el bienestar social. Se trata de un movimiento perturbador del equilibrio y se sobrentiende que la ganancia en el bienestar social debe superar a la pérdida del vértice que se sacrificó. A esta ganancia se le denomina dividendo. Brubeck Salcedo (2011) establece un teorema que determina cuándo un vértice es capaz de realizar un movimiento de especulación de manera que siempre garantice un dividendo positivo.

Técnica de especulación extendida. En la técnica de especulación extendida, el vértice que se libera puede buscar conexiones con cualquiera de los descendientes del vértice que se sacrificó o inclusive con cualquier otro vértice, con el fin de maximizar el dividendo obtenido.

Técnica de especulación con inversión cero. Relaja las condición de especulación para que no necesariamente haya dividendos positivos, mientras no haya pérdidas en bienestar social. Es decir, es un movimiento que no genera ningún beneficio de manera inmediata, pero se ejecuta con la esperanza de que un cambio a un equilibrio alterno pueda conducir eventualmente a un mejor equilibrio mediante un movimiento de especulación de otro tipo.

Mediante estas técnicas se extendió el CBAP propuesto por (Fajardo-Delgado *et al.*, 2013) y se diseñaron los algoritmos $CBAP^+$ (técnica de especulación), $CBAP^\times$ (técnica de especulación extendida), $CBAP^{++}$ (técnica de especulación + inversión cero), $CBAP^{\times\times}$ (técnica de especulación extendida + inversión cero).

Resultados experimentales

Brubeck Salcedo (2011) llevó a cabo simulaciones experimentales para comparar la calidad de las soluciones arrojadas por los diferentes algoritmos propuestos. Para comparar las mejores (peores) soluciones generadas por cada variante del CBAP se utiliza el PoS (PoA). Brubeck Salcedo (2011) utilizó la misma cantidad y tipos de casos específicos de grafos que en (Fajardo-Delgado *et al.*, 2013)

Uno de los factores que también afecta a la calidad de las soluciones obtenidas es la manera en cómo se construye el árbol de esparcimiento inicial, por lo que también se probaron tres diferentes maneras de construcción del árbol de esparcimiento: PrimRST (Raidl y Julstrom, 2003), búsqueda por anchura (BFS), y búsqueda por profundidad (DFS).

Las estrategias de jugador que se compararon fueron: voraz, anti-voraz e IA-Voraz.

Las políticas de calendarizador que Brubeck Salcedo (2011) probó fueron: voraz, anti-voraz y uniforme.

Para cada estrategia, se determinó qué calendarizador es mejor, y para cada método de generar el árbol de esparcimiento, se evaluó qué combinación (estrategia-calendarizador) es la que da mejores resultados.

Resultados

Para grafos poco densos (de 128 aristas), los métodos de BFS y DFS entregaron los mejores resultados. Para el resto de los grafos (de 256 aristas o más), el método de PrimRST es el que funcionó mejor, por lo que es el método de árbol de esparcimiento inicial que se utilizó para el resto de los experimentos.

Bajo el enfoque no cooperativo, la estrategia ε -voraz es estadísticamente mejor en grafos poco densos. La estrategia anti-voraz es la que mayor cantidad de movimientos de optimización realiza y por la misma razón, es la más lenta.

En el enfoque cooperativo la estrategia ε -voraz tiene ligeramente mejores resultados para grafos poco densos (de 128 y 256 aristas). Cuando el grafo se vuelve más denso

(de 512 aristas o más), la combinación anti-voraz es la que en promedio da mejores resultados. Un resultado interesante es que la configuración anti-voraz fue la única en alcanzar una solución con el bienestar social óptimo, en grafos casi completos de 2048 aristas (un grafo completo de 65 vértices tiene 2080 aristas).

El algoritmo $CBAP^+$ logra mejorar los equilibrios para grafos dispersos, especialmente para la estrategia anti-voraz. La estrategia anti-voraz en grafos dispersos es la que mayor se ve beneficiada debido a que el algoritmo ya entrega muy buenos resultados en grafos densos.

Al revés de como sucede con el $CBAP^+$, el $CBAP^\times$ tiende a mejorar más los resultados conforme los grafos se vuelven más densos. De la misma manera, el $CBAP^\times$ mejora los resultados para las estrategias voraces, por la razón de que la estrategia anti-voraz ya entrega resultados muy cercanos al óptimo.

Las configuraciones en las que se puede realizar movimientos de especulación con inversión cero son pocas. Los algoritmos $CBAP^{++}$ y $CBAP^{\times\times}$ logran mejorar los resultados previos, pero no de manera significativa.

En conclusión, el algoritmo $CBAP^{\times\times}$ es el que en promedio entrega las soluciones con los mejores PoS y PoA, sin incurrir en un costo adicional de tiempo significativo.

1.1.3. Un algoritmo evolutivo para el PAG

Encontrar un equilibrio con el máximo bienestar social se puede ver como el problema de encontrar un árbol de esparcimiento que maximice la función objetivo. Los algoritmos evolutivos, y más específicamente los algoritmos genéticos, se han utilizado exitosamente para esta clase de problemas, cuando no se sabe de la existencia de algoritmos precisos y eficientes.

Zatarain Aceves (2011) diseñó un algoritmo evolutivo llamado EA-BAP para atacar el problema de estancamiento en equilibrios no óptimos del CBAP. Sin embargo, este algoritmo no siempre encuentra un árbol con una condición de equilibrio.

Para atender esta restricción se propone un algoritmo evolutivo en cascada llama-

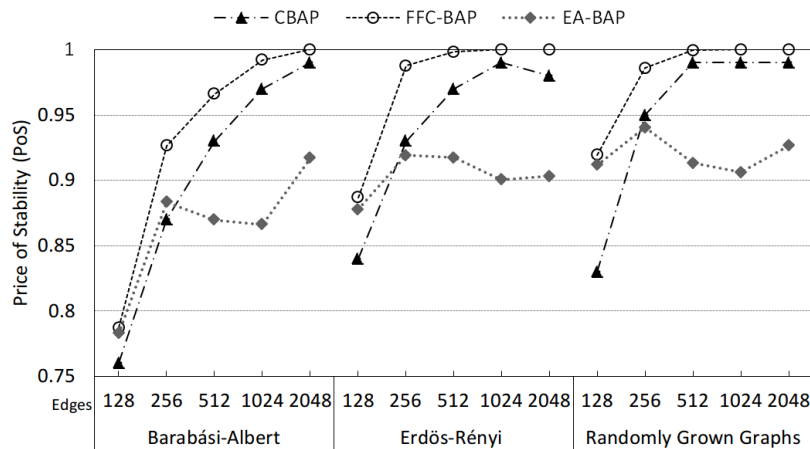


Figura 1: Valores aproximados del Precio de Estabilidad (PoS) para los algoritmos FFC-BAP, EA-BAP y CBAP para diferentes tipos de grafos con diferentes densidades con 65 vértices. (Zatarain Aceves, 2011)

do FFC-BAP, que combina los algoritmos CBAP y EA-BAP para que siempre garantice una condición de equilibrio. La etapa en cascada toma los árboles de la generación final del EA-BAP, que no estén en condición de equilibrio, y los utiliza como entrada para el CBAP. Esto hace que incrementen su bienestar social y además siempre encuentran una configuración de equilibrio.

Resultados experimentales

Zatarain Aceves (2011) llevó a cabo simulaciones experimentales para comparar la ineficiencia de los equilibrios obtenidos por los algoritmos propuestos. Zatarain Aceves (2011) utilizó el PoS y el PoA como medidas de ineficiencia de los equilibrios. Para los experimentos utilizó el mismo conjunto de datos y metodología que en (Fajardo-Delgado *et al.*, 2013) y (Brubeck Salcedo, 2011).

En las figuras 1 y 2 se muestra que el algoritmo FFC-BAP genera óptimos locales de mayor calidad que los algoritmos CBAP y EA-BAP para todos los tipos de grafos y densidades. Por otro lado, el CBAP entrega equilibrios con un mayor beneficio social que el EA-BAP para la mayoría de los casos.

Se puede apreciar que los valores aproximados del PoS del FFC-BAP son muy cercanos a 1 para varios tipos y densidades de grafos. Estos valores en realidad ya son el óptimo, o muy cercanos al óptimo, dado que se trata de cotas inferiores del valor PoS real.

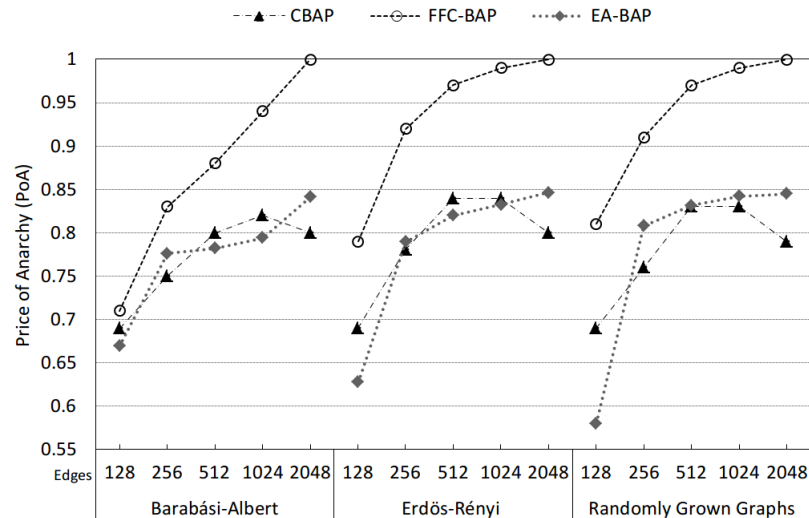


Figura 2: Valores aproximados del Precio de Anarquía (PoA) para los algoritmos FFC-BAP, EA-BAP y CBAP para diferentes tipos de grafos y densidades con 65 vértices. (Zatarain Aceves, 2011)

Sin embargo, los tiempos de ejecución del EA-BAP y FFC-BAP son mucho más altos que los del CBAP.

Como conclusión, se puede observar que el CBAP es más efectivo que el EA-BAP para encontrar árboles con una condición de equilibrio de mayor calidad. Adicionalmente, aunque el FFC-BAP es al menos tan efectivo como el CBAP en cuanto a calidad de las soluciones, el segundo es mucho más rápido.

1.2. Objetivo de la investigación

El objetivo que se definió para este trabajo de tesis fue generalizar el problema del PAG para que considere hasta n clases de vértices, tal que cada clase de vértice maximiza su ganancia individual cuando éste se coloque a una distancia k de la raíz, donde $1 < k \leq n - 1$.

Con base en este objetivo general, se lograron los siguientes resultados:

1. Se definió de manera formal el PAG *multiclase* (PAGM) para que contemplase hasta $n - 1$ clases de vértices y la raíz (no sólo los casos extremos que buscan estar lo más lejos, o lo más cerca posible, de la raíz, sino también vértices que prefieran estar a cierta distancia k de la raíz).

2. Se desarrolló y analizó un algoritmo centralizado que maximiza el bienestar social del PAGM.
3. Se modificó el algoritmo distribuido DBAP, propuesto por Fajardo-Delgado *et al.* (2013) para el CBAP, para que soportase casos del PAGM.
4. Se desarrolló y analizó un algoritmo centralizado que combina de manera secuencial los enfoques cooperativo y no cooperativo con el fin de maximizar el bienestar social del PAGM.
5. Se propuso una colección de casos específicos del PAGM y se llevaron a cabo simulaciones experimentales sobre estos casos para comparar el rendimientos de los diferentes algoritmos propuestos.
6. Se demostró de manera analítica que bajo un enfoque cooperativo, el PAGM siempre converge al equilibrio. También se demostró que bajo un enfoque no cooperativo, el PAGM no necesariamente converge al equilibrio.

1.3. Contribución al conocimiento

Una de las limitaciones del PAG es que representa un escenario restringido donde sólo se consideran casos extremos, de ahí que la motivación principal de este trabajo de investigación sea extender este problema.

Mediante el problema que se propone en este trabajo de investigación es posible analizar la ineficiencia de los equilibrios cuando se usan enfoques cooperativos y no cooperativos en escenarios heterogeneos.

Nuestro objetivo final es estudiar cómo diferentes grados de individualismo y cooperación ayudan a alcanzar mejores o peores equilibrios.

1.4. Organización de la tesis

El presente documento se organiza de la siguiente manera. En el Capítulo 2 se define, de manera matemática, el problema de asignación de guardaespaldas multiclase (PAGM), así como los enfoques cooperativos y no cooperativos que aplican para éste.

El Capítulo 3 presenta un algoritmo centralizado para la resolución del PAGM junto con su análisis de convergencia. En el Capítulo 4 se propone un algoritmo que combina los enfoques cooperativo y no cooperativo de manera secuencial, y que obtiene mejores resultados que el algoritmo propuesto en el Capítulo 3, a cambio de un incremento en el tiempo de ejecución. En el Capítulo 5 se analizan los resultados experimentales de los algoritmos propuestos en los Capítulos 3 y 4. Por último, en el Capítulo 6 se presentan las conclusiones del trabajo de investigación y una discusión del trabajo a futuro.

Capítulo 2. Problema de asignación de guardaespaldas multiclase

En este capítulo se presenta una versión extendida del PAG, denominada PAG Multiclase (PAGM). En el PAG existen únicamente tres tipos de vértices: la raíz, los vértices blancos, que buscan acercarse lo más posible a la raíz, y los vértices negros, que buscan alejarse lo más posible de la raíz. En el PAGM pueden existir hasta n clases de vértices, donde una clase sigue siendo conformada por un único vértice, la raíz, que representa al cliente que se quiere proteger, y el resto de las clases representan a los $n - 1$ guardaespaldas que tienen diferentes preferencias con respecto a la distancia que buscan estar del cliente (i.e., ya no solo prefieren estar lo más lejos o cerca posible de la raíz, sino que pueden preferir alguna distancia específica entre 1 y $n - 1$).

2.1. Definición

El Problema de Asignación de Guardaespaldas Multiclase (PAGM) se representa mediante un grafo, conectado, no dirigido, $G = (V_G, E_G)$. Existe un vértice r , denominado la raíz, que es al que el resto de los vértices buscan conectarse. Existe una función $k : V_G \setminus r \rightarrow \mathbb{Z}^+$, donde $1 \leq k(u) \leq n - 1$, $\forall u \in V_G \setminus r$, que indica la distancia a la raíz preferida por el vértice u .

La salida del PAGM es un árbol de esparcimiento $T = (V_T, E_T)$ enraizado en r , tal que $V_T = V_G$, $E_T \subseteq E_G$, que maximiza la función de bienestar social que está dada por la Ecuación 4 y donde T se encuentra en condición de equilibrio.

$$g(T) = \sum_{\forall u \in V_G} f(u) \quad (4)$$

La función $g(T)$ es la suma de las utilidades individuales de todos los vértices. La función $f : V_G \setminus r \rightarrow \mathbb{Z}^+$ denota la utilidad individual para cada uno de los vértices. Se define la función f en la Ecuación 5.

$$f(u) = (n - 1) - |k(u) - d_u^T| \quad (5)$$

donde d_u^T es la distancia entre los vértices u y r en el árbol T , para toda $u \in V_G$. En la Ecuación 5, el primer término de la derecha representa la máxima contribución individual que puede aportar un vértice al estar en su posición ideal. El segundo término resta la distancia a la cual se encuentra de esta posición.

Para el PAGM se mantienen el resto de todas las definiciones del PAG (Fajardo-Delgado *et al.*, 2013) relevantes a la estructura del árbol T :

Se denota a N_u^G como el conjunto de vecinos para cualquier vértice $u \in V_g$.

El símbolo P_u denota un apuntador que tiene cada vértice u que puede señalar a cualquier elemento del conjunto N_u^G .

Se denota a D_u^T como el conjunto de descendientes de u en T ; por simplicidad en cómo se implementa el algoritmo que se describe en el Capítulo 3, u es un descendiente de sí mismo, pero no un ancestro.

Se denota a A_u^G como el conjunto de vecinos no-descendientes de u en G , $A_u^G = \{v | v \in N_u^G \wedge v \notin D_u^T\}$.

Ejemplo 1. La Figura 3a muestra a un grafo G como un caso específico del PAGM, donde $n = 5$, $k(v) = 1$, $k(y) = 2$, $k(x) = 3$ y $k(u) = 4 = n - 1$. La Figura 3b muestra una solución para el PAGM en forma de un árbol de esparcimiento T . Se puede notar cómo cada uno de los vértices se encuentra en la posición de su preferencia con respecto a r .

Lema 1. *El PAGM es una generalización del PAG, esto es, todos los casos del PAG se pueden expresar como casos del PAGM.*

Demostración. En el PAG existen únicamente tres tipos de vértices: la raíz, blancos y negros, cuyas funciones de utilidad individual de los últimos dos están denotadas por las ecuaciones 2 y 3, respectivamente. Los vértices blancos son aquellos que buscan estar lo más cerca posible de la raíz, por lo que en el PAGM equivalen a aquellos que tienen

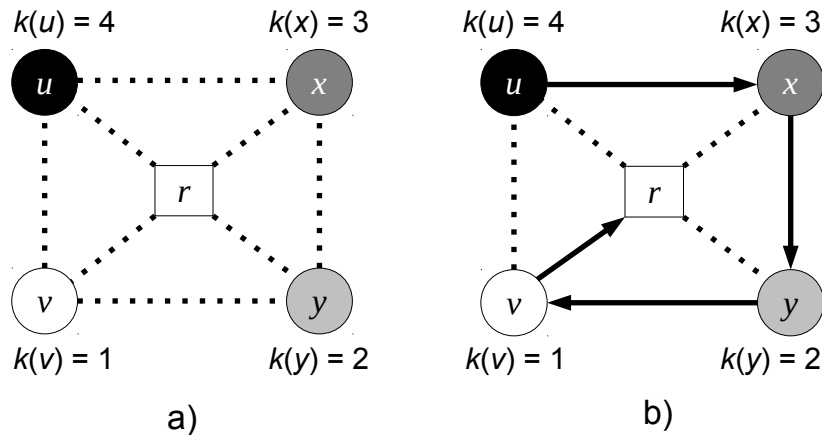


Figura 3: El problema de asignación de guardaespaldas multiclase. (a) Un caso específico del PAGM modelado como un grafo G . (b) El árbol T construido en G representa la solución para este caso específico del PAGM.

una $k(u) = 1$. Si se sustituye este valor en la Ecuación 5 se tiene la Ecuación 6.

$$f(u) = (n - 1) - |1 - d_u^T| \quad (6)$$

Dado que los términos dentro del valor absoluto representan una distancia entre dos puntos, éstos se pueden invertir como se muestra en la Ecuación 7.

$$f(u) = (n - 1) - |d_u^T - 1| \quad (7)$$

La Ecuación 8 denota la mínima y máxima distancia que puede tener un vértice hacia la raíz.

$$1 \leq d_u^T \leq n - 1 \quad (8)$$

Así, en la Ecuación 7, se puede eliminar el valor absoluto, dado que la expresión

contenida en éste siempre es positiva, quedando como se muestra en la Ecuación 9.

$$f(u) = (n - 1) - (d_u^T - 1) = n - d_u^T = f_1(u) \quad (9)$$

De esta manera, se demuestra que los vértices con una $k(u) = 1$ en el PAGM equivalen a los vértices blancos en el PAG. Los vértices negros en el PAG son aquellos que desean estar lo más alejados posibles de la raíz; dado que la máxima distancia posible a la raíz en cualquier caso del PAG es $n - 1$, entonces los vértices negros del PAG equivalen a los vértices con una $k(u) = n - 1$ en el PAGM como se demuestra a continuación.

Se sustituye el valor $n - 1$ por $k(u)$ en la Ecuación 5 y resulta la Ecuación 10.

$$f(u) = (n - 1) - |(n - 1) - d_u^T| \quad (10)$$

Dado que la expresión dentro del valor absoluto siempre es positiva, éste se puede eliminar, quedando la Ecuación 11.

$$f(u) = (n - 1) - (n - 1 - d_u^T) = d_u^T = f_2(u) \quad (11)$$

La Ecuación 11 es igual la Ecuación 3, la cual expresa la utilidad individual de los vértices negros. De esta manera queda demostrado que el PAGM incluye los tipos de vértices del PAG. \square

Lema 2. *El bienestar social $g(T)$ del PAGM es a lo más $O(n^2)$.*

Demostración. La máxima contribución que puede aportar un vértice con su utilidad individual al bienestar social es cuando se encuentra en su posición preferida; cuando esto sucede, el sustraendo del lado derecho de la Ecuación 5 se vuelve cero, y su contribución es $n - 1$. El máximo bienestar social se puede alcanzar cuando todos los $n - 1$ vértices diferentes de la raíz se encuentran en su posición preferida, así $g(T) = O(n^2)$. \square

2.2. Teoría de juegos y el PAGM

En esta sección se establecen las bases para el juego del PAGM con base en el juego del PAG propuesto por Fajardo-Delgado *et al.* (2013). El PAGM es un juego \mathcal{G} que se compone de la tupla $(\Lambda, V_G \setminus r, S_u, \pi_u)$, donde:

- El símbolo Λ representa un caso específico arbitrario de \mathcal{G} que incluye un grafo conectado $G = (V_G, E_G)$.
- $V_G \setminus r$ es el conjunto de vértices que contiene a los jugadores racionales del juego. El vértice especial r es un jugador pasivo que no toma decisiones;
- El conjunto S_u son todas las posibles estrategias para cada jugador $u \in V_G \setminus r$; y
- La función de evaluación π_u calcula las preferencias para cualquier jugador $u \in V_G \setminus r$ en el grafo.

Se supone que todos los casos específicos del PAGM comienzan con un árbol de esparcimiento arbitrario T enraizado en r . Cada vértice u sigue una estrategia de S_u , donde $S_u = \{\text{acercarse a } r, \text{ alejarse de } r\}$. Se dice que cada vértice u realiza un *movimiento* M_u cuando cambia su apuntador de $P_u = v$ a $P_u = v'$. Se restringe cada movimiento de modo que v' debe ser un elemento del conjunto A_u^G , de otro modo el árbol T se desconectaría.

El PAGM también se estudia bajo enfoques cooperativo y no cooperativo. Al igual que en el PAG, cuando un vértice u sigue un enfoque cooperativo (no cooperativo), u busca incrementar el bienestar social (utilidad individual) en vez de la utilidad individual (bienestar social).

2.2.1. Contribución de un vértice al bienestar social

Cada vértice u tiene una función $\pi'_u : v \rightarrow \mathbb{Z}, \forall v \in A_u^G$, mediante la cual puede evaluar su contribución al bienestar social al estar conectado al vértice v . Para ésto, el vértice u puede o no (dependiendo de su enfoque, cooperativo o no cooperativo) tomar en cuenta las preferencias de posición de todos los vértices del subárbol enraizado en él.

Sea $T_u = (V_{T_u}, E_{T_u})$ el subárbol de T enraizado en el vértice u . Se particiona V_{T_u} en la familia indizada de conjuntos $\{\alpha_i\}_{i \in \{1, \dots, n-1\}}$. El conjunto α_i contiene a todos aquellos vértices descendientes de u que, dada su posición en T_u , su función de preferencia k , y la distancia de v a la raíz d_v^T , contribuirán con una utilidad individual igual a i al bienestar social, cuando $P_u = v$.

La Ecuación 12 representa la función de contribución al bienestar social π'_u , del subárbol T_u cuando $P_u = v$.

$$\pi'_u(v) = (1)|\alpha_1| + (2)|\alpha_2| + \dots + (n-2)|\alpha_{n-2}| + (n-1)|\alpha_{n-1}| \quad (12)$$

La contribución al bienestar social del subárbol T_u es la suma de las utilidades individuales de todos los vértices que lo componen, esto es, existen un número $|\alpha_1|$ de vértices en el subárbol T_u con una utilidad individual igual a 1, así como un número $|\alpha_2|$ de vértices con una utilidad individual igual a 2, y así hasta un número $|\alpha_{n-1}|$ de vértices con una utilidad individual igual a $n-1$ (la máxima posible). La Ecuación 12 se puede reescribir como la Ecuación 13.

$$\pi'_u(v) = \sum_{i=1}^{n-1} (i)|\alpha_i| \quad (13)$$

Sea d un vértice descendiente de u . Para encontrar a qué conjunto α_i pertenece el vértice d , el vértice u utiliza la Ecuación 14.

$$i = (n-1) - |k(d) - (d_v^T + 1 + (d_d^T - d_u^T))| : d \in \alpha_i \quad (14)$$

Ejemplo 2. En la Figura 4 se muestra un caso específico del PAGM, donde el vértice u tiene la opción de conectarse a la raíz o al vértice x . Para que el vértice u pueda determinar la contribución que genera al conectarse a la raíz, requiere primero determinar a qué conjunto α_i pertenecen sus descendientes, v , y , y u mismo, si realiza esta conexión.

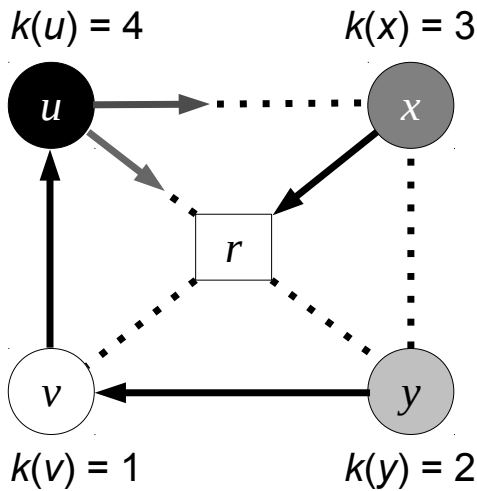


Figura 4: Un caso específico del PAGM. El vértice u tiene la opción de conectarse al vértice x o a la raíz.

Se sustituyen los valores correspondientes en la Ecuación 14 para los vértices u , v , y , en las ecuaciones 15, 16, y 17, respectivamente.

$$i = (n - 1) - |k(u) - (d_r^T + 1 + (d_u^T - d_u^T))| = (5 - 1) - |4 - (0 + 1 + (1 - 1))| = 1 \quad (15)$$

$$i = (n - 1) - |k(v) - (d_r^T + 1 + (d_v^T - d_u^T))| = (5 - 1) - |1 - (0 + 1 + (2 - 1))| = 3 \quad (16)$$

$$i = (n - 1) - |k(y) - (d_r^T + 1 + (d_y^T - d_u^T))| = (5 - 1) - |2 - (0 + 1 + (3 - 1))| = 3 \quad (17)$$

Se sustituyen los resultados de las ecuaciones 15, 16, y 17 en la Ecuación 13, para determinar cuál es la contribución del vértice u si se conecta a la raíz. El resultado queda como se muestra en la Ecuación 18.

$$\pi'_u(r) = \sum_{i=1}^{n-1} (i) |\alpha_i| = (1)(1) + (2)(0) + (3)(2) + (4)(0) = 7 \quad (18)$$

Por lo tanto, la contribución que genera el vértice u al conectarse a la raíz es igual a 7.

2.2.2. Función de ganancia

Cuando un vértice u busca cambiar de posición en el árbol T para mejorar el bienestar social en el PAGM, requiere evaluar su función de contribución π'_u para todos sus vecinos

Tabla 1: Se muestra a qué conjunto pertenecen los vértices del subárbol T_u en el ejemplo de la Figura 4.

P_u	u	v	y
r	α_1	α_3	α_3
x	α_2	α_2	α_2

en A_u^G , y después comparar con el valor de $\pi'_u(P_u)$ (su contribución actual), para poder determinar qué movimientos le pueden generar un incremento o decremento en la contribución al bienestar social. A esta diferencia se le denomina *ganancia*. La Ecuación 19 expresa la función de ganancia $\pi_u : v \rightarrow \mathbb{Z}, \forall v \in A_u^G$ del vértice u cuando se conecta al vértice v .

$$\pi_u(v) = \pi'_u(v) - \pi'_u(P_u) \quad \forall v \in A_u^G \quad (19)$$

2.2.3. Enfoque cooperativo

En el enfoque cooperativo, un vértice u toma en cuenta las preferencias de todos los vértices en el subárbol enraizado en él, para tomar la decisión sobre qué vértice le conviene para realizar una conexión. El vértice u requiere evaluar a qué conjunto α_i pertenece cada uno de sus descendientes por cada vértice con el que puede realizar conexión.

Ejemplo 3. En el caso específico del PAGM que se muestra en la Figura 4, el vértice u tiene la opción de conectarse a la raíz o al vértice x . La Tabla 1 muestra a qué conjunto α_i pertenecen cada uno de los vértices descendientes de u , según se conecte a la raíz o al vértice x .

Si se utilizan los valores de la Tabla 1, la Ecuación 12 queda como se muestra en las ecuaciones 21 y 20.

$$\pi'_u(r) = (1)(1) + (2)(0) + (3)(2) + (4)(0) = 7 \quad (20)$$

$$\pi'_u(x) = (1)(0) + (2)(3) + (3)(0) + (4)(0) = 6 \quad (21)$$

Dado que la contribución que obtiene el vértice u al conectarse a la raíz es mayor que la que obtiene al conectarse al vértice x , bajo el enfoque cooperativo, el vértice u de la Figura 4 se conecta a la raíz.

2.2.4. Enfoque no cooperativo

En el enfoque no cooperativo, para realizar movimientos, cada vértice u utiliza únicamente la información de su propia función de preferencia $k(u)$ y distancia a la raíz d_u^T , sin tomar en cuenta la información de sus vértices descendientes. Bajo este enfoque, cuando un vértice u calcula la contribución que genera cuando se conecta a un vértice $v \in A_u^G$, solamente un conjunto α_i tendrá un único elemento en la función π'_u de la Ecuación 13, que es el mismo vértice u .

Para determinar qué conjunto α_i contiene al vértice u , se sustituyen los valores correspondientes en la Ecuación 14, quedando como se muestra en la Ecuación 22.

$$i = (n - 1) - |k(u) - (d_v^T + 1 + (d_u^T - d_u^T))| = (n - 1) - |k(u) - (d_v^T + 1)| \quad (22)$$

Dado que el vértice u se conecta directamente al vértice v , $d_v^T + 1 = d_u^T$, por lo que la Ecuación 22 se reduce a la Ecuación 23.

$$i = (n - 1) - |k(u) - (d_u^T)| \quad (23)$$

De esta manera, $|\alpha_{(n-1)-|k(u)-(d_u^T)}| = 1$ y, $|\alpha_i| = 0, \forall \alpha_i : i \neq (n - 1) - |k(u) - (d_u^T)|$. Se utilizan estos valores en la Ecuación 13, que queda como se muestra en la Ecuación 24.

$$\pi'_u(v) = \sum_{i=1}^{n-1} (i)|\alpha_i| = ((n - 1) - |k(u) - (d_u^T)|)(1) \quad (24)$$

La Ecuación 24 es igual a la función de utilidad individual $f(u)$ que se muestra en la Ecuación 5. De esta manera se demuestra que bajo el enfoque no cooperativo se puede utilizar únicamente el valor de la utilidad individual del vértice u , para determinar qué conexión conviene más realizar al vértice.

Ejemplo 4. En el caso específico del PAGM que se muestra en la Figura 4, el vértice u tiene la opción de conectarse a la raíz o al vértice x . Para determinar qué opción genera una mayor contribución bajo el enfoque no cooperativo, se compara la función de utilidad individual $f(u)$ cuando u se conecta a ambos vértices. En las ecuaciones 25 y 26 se muestran los resultados de la utilidad individual del vértice u cuando se conecta a la raíz y al vértice x , respectivamente.

$$f(u) = (n - 1) - |k(u) - (d_u^T)| = (5 - 1) - |4 - 1| = 1/P_u = r \quad (25)$$

$$f(u) = (n - 1) - |k(u) - (d_u^T)| = (5 - 1) - |4 - 2| = 2/P_u = x \quad (26)$$

Por lo tanto, bajo el enfoque no cooperativo, u se conecta a x .

Capítulo 3. Algoritmo CBAPM

En este capítulo se presenta el algoritmo CBAPM que resuelve el PAGM. El algoritmo es una versión modificada del algoritmo CBAP que se presenta en (Fajardo-Delgado *et al.*, 2013) para la resolución del PAG. El CBAPM trabaja de manera centralizada y secuencial, y puede funcionar utilizando tanto el enfoque cooperativo como el no cooperativo. La entrada del algoritmo es cualquier caso específico válido del PAGM, que incluye un grafo no dirigido, conectado $G = (V_G, E_G)$, donde $|V_G| = n$ y $|E_G| = m$, y una función de preferencia $k : u \rightarrow \mathbb{Z}^+ \forall u \in V_G \setminus r$. La salida del algoritmo es un árbol de esparcimiento $T = (V_T, E_T)$ construido en G , tal que satisface la condición de equilibrio del PAGM y que no necesariamente maximiza el bienestar social.

3.1. Análisis del algoritmo CBAPM

El algoritmo CBAPM consiste de seis procedimientos como se muestra en el Algoritmo 2.

Algoritmo 2 CBAPM(G)

- 1: $T \leftarrow \text{CREAR-ARBOL}(G, r)$
 - 2: $T \leftarrow \text{ETIQUETADO-INICIAL}(T)$
 - 3: $T \leftarrow \text{GENERAR-FUNCIONES-GANANCIA}(T)$
 - 4: **mientras** $\exists u$ tal que $u \leftarrow \text{CALENDARIZADOR}(G, T)$ **hacer**
 - 5: $P_u^{old} \leftarrow P_u$
 - 6: $T' \leftarrow \text{MOVIMIENTO-OPTIMIZACION}(G, T, u)$
 - 7: $T \leftarrow \text{ACTUALIZACION}(T', u, P_u^{old})$
 - 8: **fin mientras**
-

En el Algoritmo 2, los pasos 1-3 generan un árbol de esparcimiento aleatorio inicial, en el cual cada vértice tiene la información requerida para cambiar de conexión evitando desconectar al árbol y no generar conexiones cíclicas. El Paso 4 comprende la parte encargada de la optimización. En este paso, el árbol cambia iterativamente de acuerdo a las preferencias de los vértices. El Paso 4 termina cuando el algoritmo alcanza una condición de equilibrio. El algoritmo CBAPM varía del algoritmo CBAP principalmente en los procedimientos (Paso 3) GENERAR-FUNCIONES-GANANCIA y (Paso 7) ACTUALIZACION. A continuación se explican en detalle cada uno de los procedimientos del CBAPM.

3.1.1. Procedimiento CREAR-ARBOL()

Se utiliza el mismo procedimiento propuesto por Fajardo-Delgado *et al.* (2013). La entrada de este procedimiento es el grafo G y el nodo especial r . La salida es un árbol de esparcimiento $T = (V_T, E_T)$ construido en G y enraizado en r , denominado a partir de ese momento como la raíz. Para la construcción de este árbol se utiliza el método PrimRST de Raidl y Julstrom (2003). Este método lo utiliza el CBAP. Después de enraizar el árbol T en r , cada vértice $u \neq r$ guarda una variable P_u que apunta a su padre, tal que siempre exista una trayectoria que lleve a cualquier vértice u a r . Aunque se podría considerar a T como un árbol dirigido, la comunicación que existe entre cualquier par de vértices vecinos es bidireccional. Este procedimiento toma $\Theta(n + m)$ u.t.

3.1.2. Procedimiento ETIQUETADO-INICIAL()

Se utiliza el mismo procedimiento propuesto por Fajardo-Delgado *et al.* (2013). La entrada de este procedimiento es un árbol T y regresa como salida un identificador temporal, $label_u$, y la distancia d_u^T para cada vértice. Esta información es necesaria para evitar la formación de ciclos y que cada vértice pueda realizar movimientos de optimización válidos. El procedimiento es similar al utilizado por Zeev Collin (1994). Los identificadores siguen un orden lexicográfico de acuerdo a la estructura de T . Cada vértice u asigna un orden local arbitrario η_u a las aristas incidentes a él, tal que $\eta_u(v)$ denota el índice de la arista que conecta el vértice u con el vértice v . La raíz siempre se inicializa con la etiqueta $label_r \leftarrow \perp$. La etiqueta de cada vértice u consiste en la concatenación de la etiqueta de su vértice padre v con el índice $\eta_u(v)$, es decir, $label_u \leftarrow label_v \circ \eta_u(v)$. Nótese que cualquier descendiente del vértice u contiene, como prefijo de su propia etiqueta, la etiqueta de u . Por lo tanto, cualquier vértice u sólo necesita leer las etiquetas de sus vecinos para formar el conjunto A_u^G . Estas etiquetas ayudan a preservar la estructura del árbol durante el proceso de optimización, de manera que no se realicen movimientos que desconecten el árbol. En este procedimiento cada vértice u también lleva a cabo el cálculo de su distancia d_u^T hacia raíz en T . Sea v el padre de u , $d_u^T \leftarrow d_v^T + 1$. El procedimiento ETIQUETADO-INICIAL() calcula de manera secuencial los valores $label_u$ y d_u^T de cada vértice mediante un simple recorrido preorden de T . Este procedimiento toma $\Theta(n)$ u.t.

3.1.3. Procedimiento GENERAR-FUNCIONES-GANANCIA()

Este procedimiento es donde el algoritmo difiere del CBAP. El CBAP tiene un procedimiento que recibe como entrada el árbol T y la salida es un árbol T' donde cada vértice u registra en unas variables α_u y β_u el número de vértices blancos y negros en el subárbol T_u . Este sistema de conteo de votos se utiliza para el enfoque cooperativo de manera tal que cuando un subárbol dado tuviera mayoría de vértices blancos (negros) buscará acercarse (alejarse) a (de) la raíz. Bajo el enfoque no cooperativo, cada vértice u sólo registra en su variable α_u o β_u el voto correspondiente a sí mismo e ignora la información del resto del árbol T_u . Este procedimiento se realiza mediante un recorrido post-orden y toma $O(n)$ u.t.

El CBAPM utiliza un procedimiento diferente que trabaja con base en las funciones de ganancia que se describen en el Capítulo 2. El procedimiento GENERAR-FUNCIONES-GANANCIA recibe como entrada un árbol T construido sobre G . La salida es un árbol T' , donde cada vértice u cuenta con su función π_u que informa cuáles son las ganancias posibles al conectarse a cualquiera de sus vecinos en el grafo G . Las conexiones a vecinos que desconecten el árbol generan una ganancia nula. Este procedimiento toma una diferente cantidad de pasos dependiendo si se utiliza el enfoque cooperativo o el no cooperativo. Bajo el enfoque no cooperativo la función de ganancia de un vértice u se construye únicamente mediante el valor aportado por la utilidad individual del vértice u . Las funciones de ganancia de cada vértice se van construyendo mediante un recorrido post-orden, y como cada vértice tiene que construir una función de ganancia para cada uno de sus vecinos, el proceso toma $O(n+m)$ u.t. Bajo el enfoque cooperativo el procedimiento se lleva a cabo de la misma manera, pero como por cada vecino un vértice debe recolectar información de todos sus descendientes, se agrega un factor $O(n)$ de u.t., por lo que bajo el enfoque cooperativo, el procedimiento GENERAR-FUNCIONES-GANANCIA toma $O(n^2 + mn)$ u.t.

3.1.4. Procedimiento CALENDARIZADOR()

Se utiliza un procedimiento basado en el propuesto por Fajardo-Delgado *et al.* (2013). La entrada de este procedimiento es el grafo G y el árbol T construido en G , junto con

toda la información asociada generada por los procedimientos anteriores. La salida es un vértice u que puede mejorar el bienestar social (utilidad individual) bajo el enfoque cooperativo (no cooperativo) al cambiar su apuntador P_u en T . Este procedimiento evalúa cada vértice en V_G para formar el conjunto de vértices elegibles denominado Z , donde $Z = \{u \in V_G \mid \pi_u(x) > 0 \wedge x \in A_u^G\}$, donde π_u es la función de ganancia que se define en la Ecuación 19. Cuando ningún vértice satisface la condición $\pi_u(x) > 0$, el procedimiento CALENDARIZADOR() regresa \emptyset (condición de equilibrio). Este procedimiento funciona como una entidad central que elige exactamente un vértice de Z para llevar a cabo un movimiento de optimización.

Cuando existen más de un vértice que pueden mejorar el bienestar social o la utilidad individual, según sea el enfoque, en el CBAP el procedimiento CALENDARIZADOR cuenta con dos políticas para la elección del vértice a optimizar del conjunto Z :

- *Uniforme*. Esta política elige de manera aleatoria cualquiera de los vértices del conjunto Z con la misma probabilidad.
- *Voraz*. Esta política siempre selecciona al vértice $u \in Z$ que genere el incremento más grande en ganancia, es decir, con la mayor $\pi_u(x)$. Si dos o más vértices cumplen con esta condición, se selecciona cualquiera de estos vértices con la misma probabilidad.

Adicionalmente, en el trabajo de Brubeck Salcedo (2011), se propone una política adicional para el procedimiento CALENDARIZADOR():

- *Anti-Voraz*. Selecciona el vértice $u \in Z$ tal que un movimiento de optimización de u genere el incremento más pequeño en ganancia; es decir, $\pi_u(x) \leq \pi_{u'}(x') \forall u' \in Z, \forall x' \in A_{u'}^G, \pi_u(x) > 0$. Si dos o más vértices cumplen con esta condición, se selecciona cualquiera de éstos con la misma probabilidad.

El CBAPM también puede utilizar cualquiera de estas políticas para el calendarizador. Por otro lado, en el CBAP, estas políticas pueden ser además *elitistas*, si al aplicar la política, el calendarizador prefiere seleccionar un vértice blanco sobre uno negro, o viceversa.

Esta distinción no existe en el CBAPM, puesto que pueden existir más de dos tipos de vértices, por lo que se consideran todas las políticas como *no elitistas*.

La parte que más tiempo consume en este procedimiento es generar el conjunto de vértices Z , ya que cada vértice u tiene que evaluar $\pi_u(x)$ para cada vértice x de sus vecinos no descendientes. Por lo tanto, el procedimiento CALENDARIZADOR() toma $O(n + m)$ u.t.

3.1.5. Procedimiento MOVIMIENTO-OPTIMIZACION()

Se utiliza el mismo procedimiento que el propuesto por Fajardo-Delgado *et al.* (2013). Este procedimiento recibe como entrada el grafo G , el árbol T construido en G , y el vértice u que seleccionó por el procedimiento CALENDARIZADOR(). Regresa como salida un nuevo árbol T' como resultado de un movimiento del vértice u , el cual cambia su apuntador a su padre actual $P_u = x$ a $P_u \leftarrow x'$, donde x' es el vértice tal que $\pi_u(x') = \max_{x \in A_u^G} \{\pi_u(x)\}$. Este procedimiento toma $\Theta(n)$ u.t.

3.1.6. Procedimiento ACTUALIZACION()

Este procedimiento es análogo al propuesto por Fajardo-Delgado *et al.* (2013) en el aspecto que realiza las mismas funciones que los procedimientos ETIQUETADO-INICIAL() y GENERAR-FUNCIONES-GANANCIA(), pero únicamente para los subárboles que se han modificado a causa del movimiento generado en el procedimiento MOVIMIENTO-OPTIMIZACIÓN(). Este procedimiento recibe como entrada al árbol T' y a los vértices u y P_u^{viejo} (el padre que tenía u antes de realizar el movimiento de optimización). La salida de este procedimiento es un nuevo árbol T' con los nuevos valores de etiquetas, distancias y funciones de ganancia. El procedimiento ACTUALIZACION() se completa en $O(n + m)$ u.t. bajo el enfoque no cooperativo, y en $O(n^2 + mn)$ u.t., bajo el enfoque cooperativo.

3.2. Demostración de convergencia

En esta sección se presentan las demostraciones de convergencia y no convergencia para los enfoques cooperativo y no cooperativo. El enfoque cooperativo converge al equilibrio para cualquier caso específico del PAGM. Por otro lado, el enfoque no cooperativo sólo converge al equilibrio bajo determinadas condiciones.

3.2.1. Enfoque cooperativo

Se define un *movimiento de optimización* como todo aquel *movimiento* que realiza un vértice que genera un incremento en el bienestar social y no solamente en la utilidad individual del vértice que realizó el movimiento.

Nota. El movimiento de optimización como se define para el PAGM es diferente al *movimiento de optimización* que se define para el PAG en (Fajardo-Delgado *et al.*, 2013).

Lema 3. *Bajo el enfoque cooperativo, todos los movimientos realizados por cualquier vértice son movimientos de optimización.*

Demostración. Bajo el enfoque cooperativo, por la Ecuación 13, cada vértice suma las utilidades de todos los descendientes en su subárbol para obtener la ganancia que genera cada conexión posible. Cuando un vértice va a realizar un movimiento, compara las ganancias de todos los vecinos con la ganancia obtenida en su posición actual. Solamente los vecinos que representen una ganancia mayor a la de la posición actual serán elegibles por el calendarizador para realizar movimientos, de manera que siempre se garantiza un incremento en el bienestar social. \square

Teorema 4. *El algoritmo CBAPM converge al equilibrio en $O(n^2)$ movimientos de optimización bajo el enfoque cooperativo.*

Demostración. Se supone un árbol inicial T con un bienestar social τ . Por el Lema 3, el bienestar social se incrementa en al menos uno por cada movimiento de optimización. Dado que el valor máximo del bienestar social está acotado por $O(n^2)$ (por el Lema 2), y $\tau > 0$, entonces el máximo número de movimientos de optimización es $O(n^2)$. \square

Dado que cada iteración del Paso 4 en el Algoritmo 2 toma $O(n^2 + mn)$ u.t., el algoritmo CBAPM requiere un total de $O(n^4 + mn^3)$ u.t. para converger al equilibrio bajo el enfoque cooperativo.

3.2.2. Enfoque no cooperativo

Bajo el enfoque no cooperativo, el Lema 3 no se cumple, i.e., como no todos los movimientos realizados son movimientos de optimización, existen movimientos que de

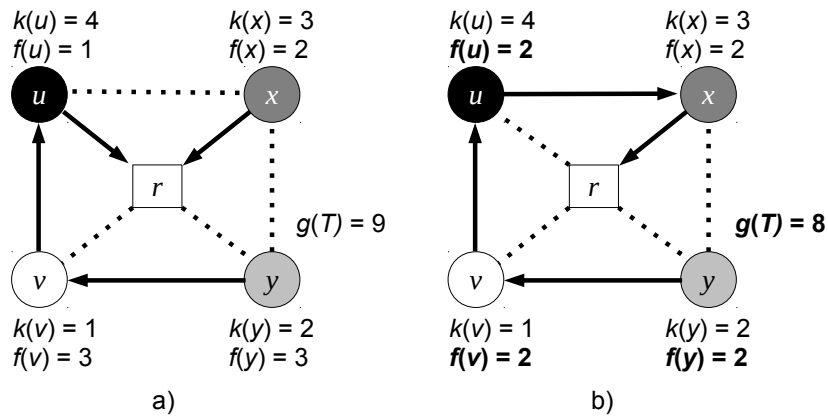


Figura 5: Se muestra un movimiento que reduce el bienestar social en vez de incrementarlo. Cuando el vértice u cambia su apuntador de r (a) a x (b), incrementa su utilidad individual, pero disminuye el bienestar social.

hecho reducen el bienestar social en vez de incrementarlo. De hecho, bajo el enfoque no cooperativo, el CBAPM no necesariamente converge al equilibrio.

Ejemplo 5. En la Figura 5 se muestra un ejemplo de un movimiento que no incrementa el bienestar social bajo el enfoque no cooperativo. En la Figura 5a el vértice u tiene una $f(u) = 1$ al estar conectado a la raíz, y el bienestar social $g(T)$, es igual a 9. En la Figura 5b el vértice u se conecta al vértice x para mejorar su utilidad individual ($f(u) = 2$), a pesar de que el bienestar social se decremента ($g(T) = 8$) como consecuencia de este movimiento.

Teorema 5. *Existe al menos un caso específico del PAG multiclase para el cual el algoritmo CBAPM no converge al equilibrio bajo el enfoque no cooperativo.*

Demostración. Se demuestra por contradicción. Se comienza suponiendo que el algoritmo CBAPM siempre converge al equilibrio bajo el enfoque no cooperativo. A continuación se supone un grafo G que representa un caso específico válido del PAGM con las siguientes características:

- La función de preferencia k de cada vértice puede tomar cualquier valor entre $1 \leq k \leq n - 1$.
- Se trata de un grafo conectado de 65 vértices que no es completo.

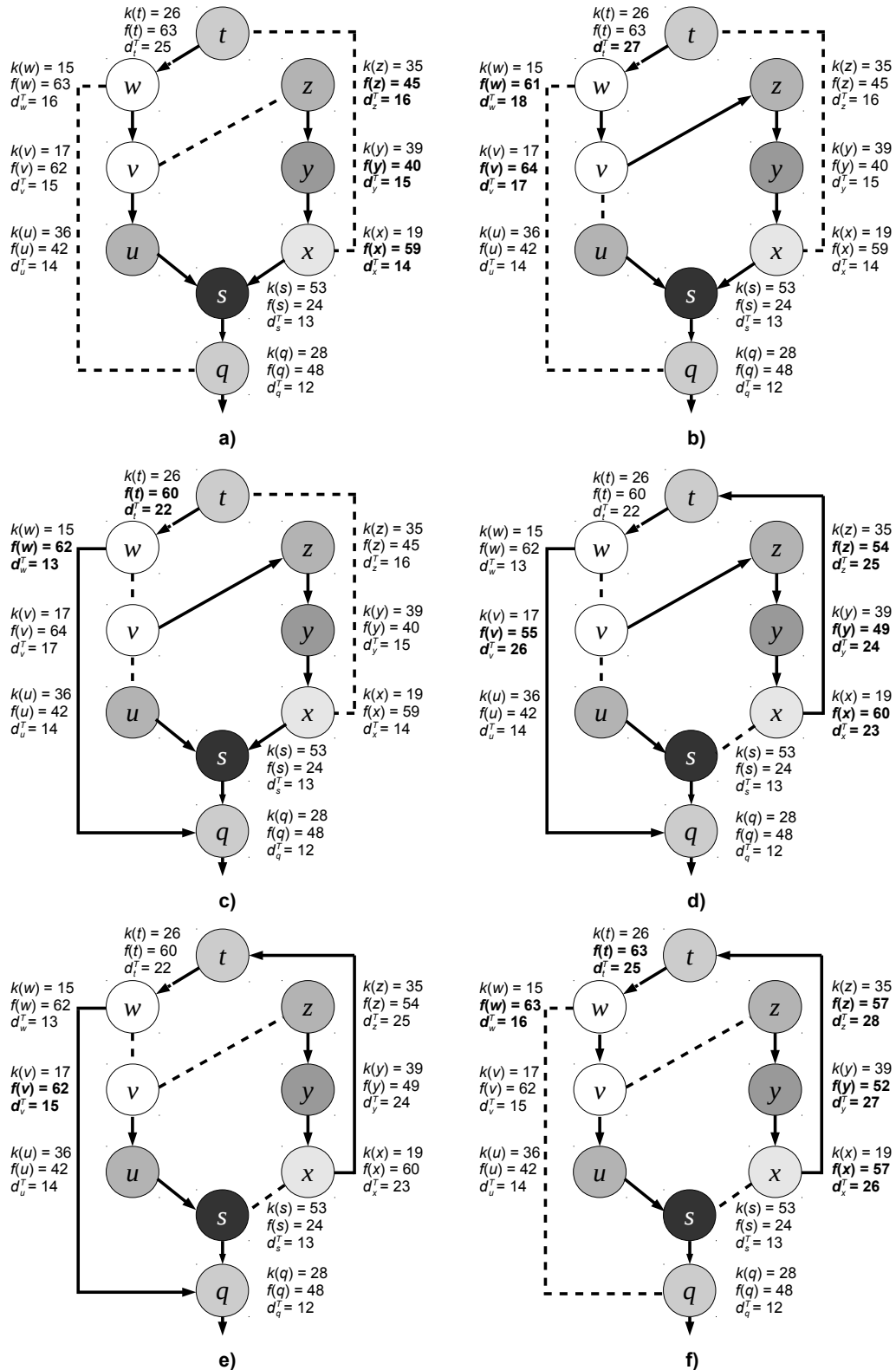


Figura 6: En este grupo de figuras se muestra la serie de movimientos que realiza el algoritmo CBAPM sobre una parte de un caso específico del PAGM, y la manera en cómo cae en una serie de movimientos cíclicos. Comienza en la configuración que se muestra en a) hasta llegar a la de f), a partir de la cual regresa a a). Se indica la posición preferida, utilidad individual y posición actual de cada vértice; se resaltan los valores que cambian de una configuración a otra. Sólo existen las aristas en G señaladas explícitamente mediante líneas o flechas.

Se supone que el algoritmo CBAPM ha realizado movimientos en el árbol de solución T sobre el grafo G hasta que todos los vértices se encuentran en equilibrio bajo el enfoque no cooperativo, con excepción de un vértice v que aún tiene la posibilidad de incrementar su utilidad individual. En la Figura 6a se muestra al vértice v junto con sus vecinos. En la Figura 6a se puede observar que el vértice v puede mejorar su utilidad individual si se desconecta de u y se conecta a z . Nótese que además éste es el único movimiento posible que puede llevar a cabo el vértice v . En la Figura 6b se muestra que cuando el vértice v se conecta a z , el vértice w , que antes estaba en equilibrio, ahora tiene la posibilidad de mejorar su utilidad individual si se conecta a q . Nótese que además éste es el único movimiento posible que puede llevar a cabo el vértice w . Cuando w se conecta a q para maximizar su utilidad individual, como se muestra en la Figura 6c, se genera un nuevo desequilibrio, por parte de x , que ahora puede beneficiarse si se conecta al subárbol de w . Nótese que además éste es el único movimiento posible que puede llevar a cabo el vértice w . Pero cuando esto sucede (Figura 6d), v , que se había conectado al subárbol de x , ya no le conviene seguir en esta posición, por lo que regresa a conectarse a u , como se muestra en la Figura 6e. De la misma manera, después de que v regresa a u , w regresa a v (Figura 6f). Cuando w se conecta a v , afecta a x , que se encuentra en el subárbol de w , por lo que cambia su conexión a s , y de esta manera regresa al punto de partida original donde el vértice v es el único en desequilibrio. Esto hace que el CBAPM nunca pueda alcanzar una configuración de equilibrio, lo cual contradice la suposición inicial. \square

Así, para ciertos casos específicos del PAGM, el árbol de solución T puede oscilar indefinidamente. Nótese que el Teorema 5 establece que el CBAPM no converge al equilibrio para todos los casos del PAGM bajo el esquema no cooperativo. Sin embargo, existen escenarios donde el CBAPM siempre convergerá, incluso bajo el esquema no cooperativo. Uno de estos casos es el grafo completo, como se muestra a continuación.

3.2.2.1. Grafo completo

En esta sección se analiza la convergencia de grafos completos bajo el enfoque no cooperativo. Se demuestra que los grafos completos siempre convergen al equilibrio, independientemente de los tipos de preferencias que pueda haber en un caso específico válido.

A grandes rasgos, la demostración parte de la suposición inicial de que existe un conjunto de vértices denominados *inamovibles*, que se encuentran en equilibrio, y mediante un proceso iterativo se anexan vértices a este conjunto hasta que todos los vértices se encuentren en equilibrio. Existen dos maneras en cómo los vértices se agregan a este conjunto; en el primer caso, que se puede denominar el caso trivial, es cuando existen vértices que al conectarse a alguno de los vértices del conjunto, encuentran su posición óptima, tal como se muestra en la Figura 7; la otra manera en que los vértices se anexan al conjunto de vértices inamovibles, es cuando todos los vértices buscan posiciones más alejadas que la inmediatamente siguiente al conjunto inamovible. En este caso, un vértice se quedará sin opciones de movimientos cuando el resto de los vértices que no están en equilibrio se vuelvan sus descendientes, y así este vértice queda en equilibrio; la Figura 8 ilustra este caso.

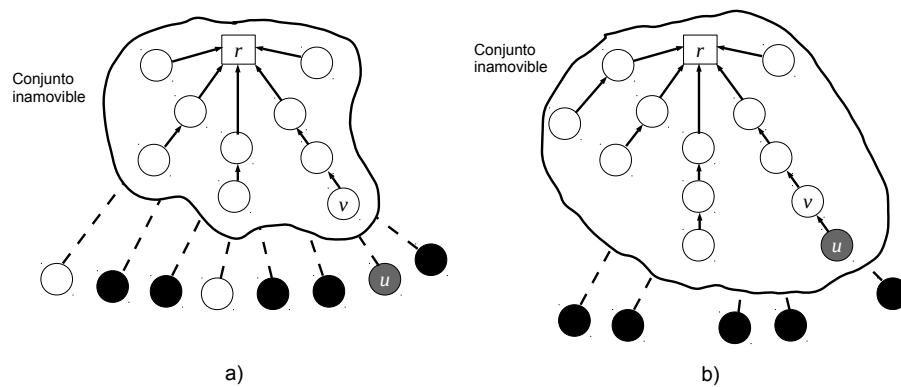


Figura 7: La figura muestra cómo se anexan vértices al conjunto inamovible. a) Existen vértices que pueden encontrar su posición óptima dentro del conjunto inamovible. b) Después de cierto número de movimientos, estos vértices pasan a formar parte del conjunto inamovible..

Se supone un caso específico válido Λ del juego PAGM \mathcal{G} que incluye un grafo arbitrario $G = (V_G, E_G)$, una función de preferencia $k : V_G \setminus r \rightarrow \mathbb{Z}^+$, que cumple la restricción $1 \leq k(u) \leq n - 1, \forall u \in V_G \setminus r$, y un árbol inicial $T_0 = (V_{T_0}, E_{T_0})$ construido sobre G , donde $V_{T_0} = V_G$ y $E_{T_0} \subseteq E_G$. Sea $\xi = u_1, u_2, \dots, u_i, \dots$ la secuencia de vértices seleccionados por el calendarizador para realizar movimientos, empezando a partir del árbol T_0 . Se dice que u_i es el i -ésimo vértice de ξ . Cuando el vértice u_i realiza un movimiento, genera una transición del árbol T_{i-1} al árbol T_i , que se denota como $T_{i-1} \xrightarrow{u_i} T_i$. Sea $\xi_{i,j} = u_i, u_{i+1}, u_{i+2}, \dots, u_j$ cualquier subsecuencia de ξ , donde $1 \leq i \leq j$; se denota la transición generada por esta subsecuencia $\xi_{i,j}$ como $T_{i-1} \xrightarrow{\xi_{i,j}} T_j$.

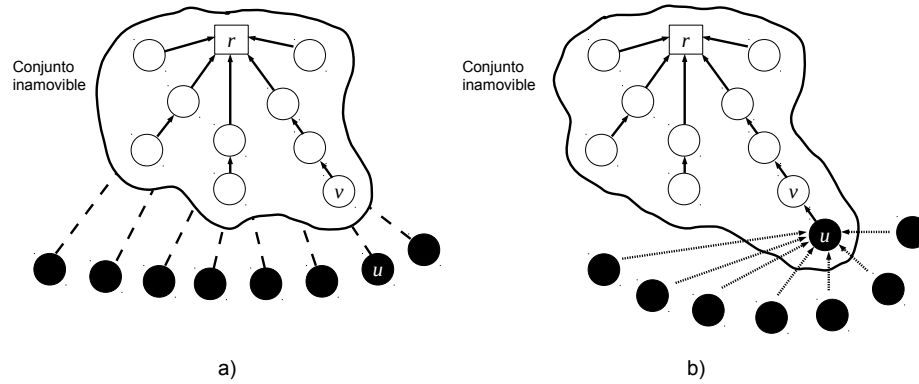


Figura 8: La figura muestra cómo se anexan vértices al conjunto inamovible. a) No existen vértices que puedan encontrar su posición óptima dentro del conjunto inamovible. b) Un vértice pasa a formar parte del conjunto inamovible cuando el resto de los vértices móviles se vuelven sus descendientes.

Se dice que cualquier vértice $u \in V_{T_j}$ es *inamovible* después del j -ésimo movimiento generado por la subsecuencia ξ , si u satisface las siguientes dos condiciones:

1. Dado Λ y T_0 , el vértice u no aparece en ninguna subsecuencia $\xi_{j+1,k}$ generada por el calendarizador, $\forall k \geq j + 1$.
2. Todos los ancestros de u en T_j son inamovibles.

Sea Φ_j el conjunto de vértices inamovibles después del j -ésimo movimiento generado por ξ ; note que $r \in \Phi_0$ y que cuando $\Phi_j = V_G$, es equivalente a que el grafo se encuentra en equilibrio. Cualquier otro vértice que no es inmovible se denomina *movible*. Se denota $\overline{\Phi_j}$ como el conjunto de vértices móviles en T_j . Dado que se trata de un grafo completo, todos los vértices de G tienen, al menos, un vecino inamovible (la raíz). Se define el concepto de *ronda*, que representa una transición $T_{i-1} \xrightarrow{\xi_{i,j}} T_j$ que incluye, al menos una vez, un movimiento de cada vértice movible.

Lema 6. *En un grafo completo, sea la transición $T_{i-1} \xrightarrow{\xi_{i,j}} T_j$ una ronda. Durante esta transición, al menos uno de los vértices móviles en el árbol T_{i-1} se convierte en inamovible en el árbol T_j .*

Demostración. Dado que $r \in \Phi_{i-1}$, $|\Phi_{i-1}| \geq 1$, i.e., existe un conjunto no vacío de vértices inamovibles. Existe un vértice $v \in \Phi_{i-1}$ tal que $d_v^{T_{i-1}} \geq d_x^{T_{i-1}}, \forall x \in \Phi_{i-1}$. Se plantean tres

escenarios diferentes en función de las clases de vértices que existen en $\overline{\Phi_{i-1}}$:

1. Si en $\overline{\Phi_{i-1}}$ existe al menos un vértice u cuya función $k(u) = d_v^{T_{i-1}} + 1$, entonces cuando este vértice sea elegido por el calendarizador, se conectará directamente al vértice v , pues este movimiento es el que maximiza su ganancia. El vértice se vuelve inamovible debido a la condición 2 y a que ha encontrado su posición óptima. El vértice puede realizar esta conexión debido a que el grafo es completo.
2. Si en $\overline{\Phi_{i-1}}$ existe al menos un vértice u cuya función $k(u) < d_v^{T_{i-1}} + 1$, entonces cuando este vértice sea elegido por el calendarizador, se conectará directamente a cualquier elemento de Φ_{i-1} que maximice su ganancia. Debido a que el grafo es completo, debe existir la arista al elemento en Φ_{i-1} al cual se conecte u , de modo que se coloque en su posición óptima. El vértice u se vuelve inamovible por las mismas razones que en el primer caso.
3. Si $\forall u \in \overline{\Phi_{i-1}}, k(u) > d_v^{T_{i-1}} + 1$, (i.e., ningún vértice movable desea estar en la posición inmediatamente siguiente a v , todos desean una posición más alejada de la raíz) entonces en este caso, el calendarizador selecciona cada uno de los vértices y éstos se van alejando de v , hasta que queda un único vértice u conectado a v . El vértice u se vuelve inamovible debido a que no tiene posibilidad de conectarse a nadie (todos los vértices movibles son sus descendientes).

Note que en cualquiera de los tres casos, al menos un vértice movable se vuelve inamovible, por lo que el Lema 6 se cumple. □

Lema 7. *En un grafo completo, el CBAPM bajo el enfoque no cooperativo requiere $O(n)$ rondas para converger al equilibrio.*

Demostración. Por el Lema 6, en cada ronda, al menos un vértice u de G se vuelve inamovible en T ; por lo que en no más de n rondas, todos los vértices se habrán vuelto inamovibles, haciendo que el árbol T se encuentre en equilibrio. □

Teorema 8. *Para cualquier caso específico del PAGM que incluya un grafo completo G , el algoritmo CBAPM requiere $O(n^2 \ln n)$ movimientos con alta probabilidad para converger al equilibrio bajo el enfoque no cooperativo y un calendarizador equiprobable.*

Demostración. Un calendarizador equiprobable elige un vértice movable con la misma probabilidad, es decir, elige cualquier vértice $u \in \overline{\Phi_j}$ con una probabilidad $\frac{1}{|\overline{\Phi_j}|} > \frac{1}{n}$. Sin pérdida de generalidad, se supone selección con reemplazo y que cualquier selección es independiente de cualquier selección previa. Mediante los resultados del problema del coleccionista de cupones (Raghavan y Motwani, 1995), cada ronda consiste de $O(n \ln n)$ movimientos con alta probabilidad. Por el Lema 7, después de $O(n)$ rondas, todos los vértices se vuelven inamovibles. Por lo tanto, $O(n)$ vértices movibles se vuelven inamovibles en $O(n^2 \ln n)$ movimientos con alta probabilidad. \square

Dado que cada iteración del Paso 4 en el Algoritmo 2 toma $O(n + m)$ u.t., el algoritmo CBAPM requiere $O(n^3 \ln n + mn^2 \ln n)$ u.t. con alta probabilidad para alcanzar el equilibrio bajo el esquema no cooperativo, para grafos completos.

3.3. Algoritmo distribuido

En esta sección se describe el algoritmo DBAPM que es una versión modificada del algoritmo DBAP realizado por Fajardo-Delgado *et al.* (2013) para el PAG. Las modificaciones en los esquemas cooperativo y no cooperativo realizadas en el CBAPM fueron lo único que se incorporó al DBAP, para poder resolver casos del PAGM.

El DBAPM es una versión distribuida del CBAPM. Se descompone la ejecución del algoritmo DBAPM en cuatro fases, cada una tiene la misma funcionalidad que los procedimientos del algoritmo CBAPM. Las fases son: TREE-CONSTRUCTION, CONVERGECAST, TOKEN-PASSING y OPTIMIZATION. A continuación se describe brevemente cada fase.

3.3.1. Descripción del algoritmo DBAP

3.3.1.1. Fase TREE-CONSTRUCTION

La fase TREE-CONSTRUCTION construye un árbol de esparcimiento arbitrario $T = (V_T, E_T)$ en el sistema. También calcula, para cada vértice u , un identificador lexicográfico $label_u$ con base en la estructura de T y la distancia d_u^T . Se utiliza una versión del

algoritmo distribuido descrito en (Zeev Collin, 1994) para realizar estas tareas. La fase TREE-CONSTRUCTION es una versión determinística de los pasos 1 y 2 del algoritmo CBAPM.

3.3.1.2. Fase CONVERGECAST

La fase CONVERGECAST consiste de una suma distribuida que recolecta la información de los descendientes de cada vértice u en el árbol. Esta información se propaga desde las hojas hasta la raíz. Cuando el algoritmo DBAPM utiliza el enfoque cooperativo, cada vértice recolecta la información proporcionada por sus descendientes directos para la composición de la función de ganancia π_u , de la misma manera en como se realiza en el CBAPM (ver Observación 3 y Ejemplo 4). La fase CONVERGECAST es equivalente al Paso 3 del algoritmo CBAPM, representa una versión distribuida del procedimiento GENERAR-FUNCIONES-GANANCIA(), descrito en la Sección 3.1.3.

3.3.1.3. Fase TOKEN-PASSING

La fase TOKEN-PASING genera un *token*, llamado ω_o , que circula de un vértice a otro en el árbol T . Se supone un sistema distribuido basado en turnos, en el que cada vértice recibe el *token* al menos una vez. También se supone que el TOKEN sigue un recorrido determinístico en profundidad. El algoritmo que se presenta en (Datta *et al.*, 2000) satisface estas condiciones. El sistema de turnos basado en el *token* provee de un mecanismo de exclusión, de modo que sólo el vértice que tenga el *token* pueda realizar movimientos. El propósito de este mecanismo es evitar que dos o más vértices realicen movimientos de manera simultánea, lo que podría provocar la generación de ciclos y la desconexión del árbol. Cuando no existen vértices que puedan llevar a cabo movimientos, el *token* ω_o circula libremente por el árbol hasta cumplir con una condición de paro. La fase TOKEN-PASSING podría verse como una política más del procedimiento CALENDARIZADOR(), descrito en la Sección 3.1.4.

3.3.1.4. Fase OPTIMIZATION

La fase OPTIMIZATION es la parte del algoritmo DBAPM donde un vértice u cambia su apuntador $P_u = v$ a $P_u = v'$. Esta fase es equivalente al Paso 4 del Algoritmo 2, lo que

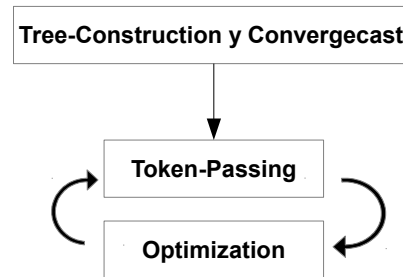


Figura 9: Diagrama a bloques de la secuencia de ejecución del algoritmo DBAPM.

incluye actualizar las etiquetas y las distancias de los vértices a la raíz en el árbol por cada movimiento que se realiza. La fase OPTIMIZATION es equivalente a los procedimientos MOVIMIENTO-OPTIMIZACION() y ACTUALIZACION() que se describen en las secciones 3.1.5 y 3.1.6, respectivamente.

Al inicio el algoritmo DBAPM ejecuta concurrentemente las fases TREE-CONSTRUCTION y CONVERGECAST. La fase TREE-CONSTRUCTION construye un árbol de esparcimiento y asigna etiquetas temporales a cada vértice. Durante la fase CONVERGECAST, cada vértice cuenta el número de descendientes en su subárbol. Después de que la raíz cuenta $n - 1$ descendientes, determina el fin del ciclo de construcción del árbol y desactiva la ejecución de ambas fases. La raíz inicia la ejecución de las fases TOKEN-PASSING y OPTIMIZATION de manera alternante. Estas dos fases no pueden ejecutarse de manera simultánea. La Figura 9 muestra la secuencia del algoritmo DBAPM. En (Fajardo-Delgado *et al.*, 2013) se propone un procedimiento similar para construir un árbol de esparcimiento mínimo auto-estabilizante.

3.3.2. Análisis del algoritmo DBAPM

Teorema 9. *El algoritmo DBAPM alcanza una condición de equilibrio en $O(n^4)$ rondas bajo el enfoque cooperativo, en cualquier caso específico del PAGM; y en $O(n^4 \ln n)$ rondas con alta probabilidad bajo el enfoque no cooperativo, en casos específicos del PAGM que incluyen grafos completos.*

Demostración. En (Fajardo-Delgado *et al.*, 2013) se define el concepto de *ronda* para el algoritmo DBAP y se demuestra que el DBAP toma $O(n^2)$ rondas para realizar un movimiento de un vértice. Utilizando *memoización* (Cormen *et al.*, 2009), el DBAPM toma $O(n^2)$ rondas para realizar un movimiento de un vértice bajo cualquiera de los dos enfoques. Utilizando los resultados de los teoremas 4 y 8, se demuestra que el DBAPM converge en $O(n^4)$ rondas y $O(n^4 \ln n)$ rondas con alta probabilidad, para los enfoques cooperativo y no cooperativo, respectivamente. \square

Capítulo 4. Algoritmo mixto

Una de las limitaciones del CBAPM está en el hecho de que una vez que encuentra una solución a un caso específico del PAGM, bajo cualquiera de los dos enfoques, cesa inmediatamente su ejecución, a pesar de que pueda existir una mejor solución a dicho caso específico. Una manera de sortear esta limitación es aprovechar el hecho de que un equilibrio bajo el enfoque no cooperativo puede ser diferente de un equilibrio bajo el enfoque cooperativo, y utilizar ambos enfoques de manera secuencial. En este capítulo se presentan tres variantes de un algoritmo llamado CBAPM mixto, que resulta de la combinación de los enfoques cooperativo y no cooperativo del algoritmo CBAPM. Las tres versiones de este algoritmo son: CBAPM Mixto (M-CBAPM), CBAPM Mixto con mejor bienestar social (M-CBAPM-BSW), y CBAPM Mixto Doble (DM-CBAPM). Las tres versiones son algoritmos centralizados y secuenciales. La entrada del algoritmo es cualquier caso específico válido del PAGM, que incluye un grafo no dirigido, conectado $G = (V_G, E_G)$ donde $|V_G| = n$ y $|E_G| = m$, y una función de preferencia $k : u \rightarrow \mathbb{Z}^+ \forall u \in V_G \setminus r$. La salida del algoritmo, si es que converge, es un árbol de esparcimiento $T = (V_T, E_T)$ construido en G , tal que satisface la condición de equilibrio del PAGM y que no necesariamente maximiza el bienestar social. Dado que una etapa de este algoritmo utiliza el enfoque no cooperativo, existen casos para los cuales el algoritmo no converge al equilibrio.

4.1. Algoritmo CBAPM Mixto

La idea central del algoritmo CBAPM Mixto (M-CBAPM) es utilizar de manera inicial el enfoque no cooperativo, para que los vértices puedan moverse con mayor libertad y realizar movimientos egoístas que maximicen su utilidad individual, aunque no necesariamente maximizan el bienestar social. Una vez que el algoritmo se ha estancado en un equilibrio no cooperativo (que se puede ver como un óptimo local y no necesariamente es un equilibrio bajo el enfoque cooperativo), el algoritmo cambia a la estrategia cooperativa, de esta manera los vértices pueden realizar movimientos donde sacrifiquen una parte de toda la utilidad individual que han obtenido en la fase anterior con el objetivo de incrementar la utilidad individual de otros vértices, y así incrementando el bienestar social total.

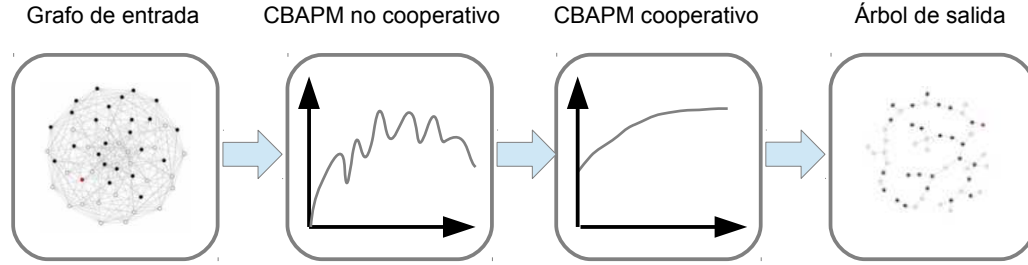


Figura 10: Diagrama de bloques del algoritmo M-CBAPM. El algoritmo recibe un caso de entrada que procesa mediante el enfoque no cooperativo. El árbol de salida en equilibrio de esta etapa se utiliza como entrada para el mismo algoritmo pero ahora bajo el enfoque cooperativo.

En la Figura 10 se muestra el diagrama de bloques del algoritmo M-CBAPM. El algoritmo recibe un caso de entrada e intenta encontrar un equilibrio mediante el enfoque no cooperativo, en el cual la función de bienestar social puede estar oscilando. Una vez que el algoritmo encuentra un equilibrio bajo este enfoque, utiliza la solución que se encontró como punto de inicio para el algoritmo bajo el enfoque cooperativo, el cual siempre incrementa la función de bienestar social. El resultado es un árbol en equilibrio bajo el enfoque cooperativo.

El algoritmo M-CBAPM se muestra en el Algoritmo 3.

Algoritmo 3 M-CBAPM(G)

- 1: $T \leftarrow \text{CREAR-ARBOL}(G, r)$
 - 2: $T \leftarrow \text{ETIQUETADO-INICIAL}(T)$
 - 3: $T \leftarrow \text{GENERAR-FUNCIONES-GANANCIA}(T)$
 - 4: **mientras** $\exists u$ tal que $u \leftarrow \text{CALENDARIZADOR}(G, T)$ **hacer**
 - 5: $P_u^{old} \leftarrow P_u$
 - 6: $T' \leftarrow \text{MOVIMIENTO-OPTIMIZACION}(G, T, u)$
 - 7: $T \leftarrow \text{ACTUALIZACION}(T', u, P_u^{old})$
 - 8: **fin mientras**
 - 9: $T \leftarrow \text{CAMBIAR-FUNCIONES-GANANCIA}(T)$
 - 10: **mientras** $\exists u$ tal que $u \leftarrow \text{CALENDARIZADOR}(G, T)$ **hacer**
 - 11: $P_u^{old} \leftarrow P_u$
 - 12: $T' \leftarrow \text{MOVIMIENTO-OPTIMIZACION}(G, T, u)$
 - 13: $T \leftarrow \text{ACTUALIZACION}(T', u, P_u^{old})$
 - 14: **fin mientras**
-

4.1.1. Procedimiento CREAR-ARBOL()

Este procedimiento se define en la Sección 3.1.1. El tiempo de ejecución de este procedimiento es $\Theta(n + m)$ u.t.

4.1.2. Procedimiento ETIQUETADO-INICIAL

Este procedimiento se define en la Sección 3.1.2. El tiempo de ejecución de este procedimiento es $\Theta(n)$ u.t.

4.1.3. Procedimiento GENERAR-FUNCIONES-GANANCIA()

Este procedimiento se define en la Sección 3.1.3. En el caso del algoritmo CBAPM Mixto, este procedimiento construye las funciones de ganancia π_u de cada vértice u utilizando en el enfoque no cooperativo. El tiempo de ejecución de este procedimiento es $O(n + m)$ u.t.

4.1.4. Procedimiento CALENDARIZADOR()

Este procedimiento se define en la Sección 3.1.4. El tiempo de ejecución de este procedimiento es $O(n + m)$ u.t.

4.1.5. Procedimiento MOVIMIENTO-OPTIMIZACION()

Este procedimiento se define en la Sección 3.1.5. El tiempo de ejecución de este procedimiento es $\Theta(n)$ u.t.

4.1.6. Procedimiento ACTUALIZACION()

Este procedimiento se define en la Sección 3.1.6. El tiempo de ejecución de este procedimiento es $O(n + m)$ u.t.

4.1.7. Procedimiento CAMBIAR-FUNCIONES-GANANCIA()

La entrada de este procedimiento es un árbol T construido sobre G . La salida es un árbol T' donde cada vértice u ha modificado su función de ganancias π_u para que ahora

utilice el enfoque cooperativo. Las funciones de ganancia de cada vértice se construyen mediante un recorrido post-orden, y como cada vértice tiene que construir una función de ganancia por cada uno de sus vecinos en G , el proceso toma $O(n + m)$ u.t.

Los pasos 10, 11, 12 y 13 del Algoritmo 3 son exactamente los mismos que los de los pasos 4 al 7 del mismo Algoritmo 3.

4.2. Algoritmo CBAPM Mixto a partir del mejor bienestar social

Una de las ventajas del enfoque cooperativo sobre el enfoque no cooperativo, es que el primero siempre garantiza movimientos que incrementan el bienestar social, mientras que en el segundo se pueden realizar movimientos que lo decrementan. De hecho, es posible que en algunos casos de ejecución del PAGM bajo el enfoque no cooperativo, el valor del bienestar social del árbol de salida no sea el valor más alto que se haya encontrado durante la optimización del mismo, i.e., que durante el procesamiento de la solución, existieron árboles con configuraciones con un bienestar social mayor al del árbol final. Lo anterior se debe a que estos árboles no se encontraban en configuración de equilibrio bajo el enfoque no cooperativo.

Una variante del M-CBAPM consiste en inicializar la etapa de enfoque cooperativo utilizando el árbol T con mayor bienestar social encontrado durante la etapa no cooperativa como entrada, en vez de utilizar la salida propiamente dicha de la primera etapa. Es posible que el árbol con mayor bienestar social de la etapa no cooperativa sea igual al árbol de salida cuando esté en equilibrio. En la Figura 11 se muestra el diagrama a bloques de este algoritmo, denominado, CBAPM Mixto con mejor bienestar social (M-CBAPM-BSW).

El algoritmo M-CBAPM-BSW se muestra en el Algoritmo 4.

4.2.1. Procedimiento CREAR-ARBOL()

Este procedimiento se define en la Sección 3.1.1. El tiempo de ejecución de este procedimiento es $\Theta(n + m)$ u.t.

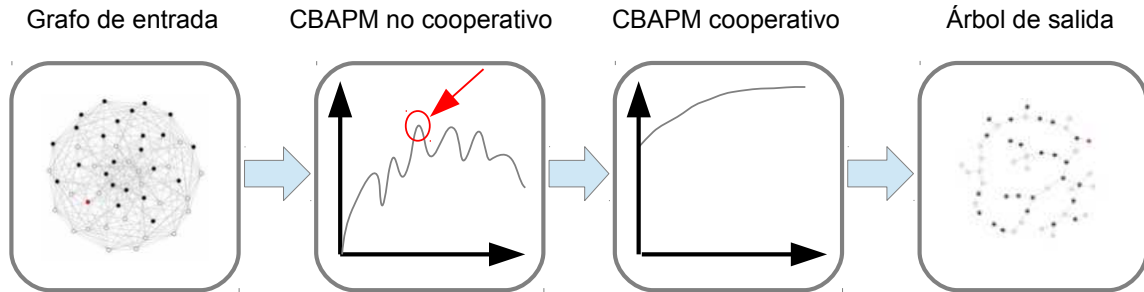


Figura 11: Diagrama de bloques del algoritmo M-CBAPM-BSW. El algoritmo recibe un caso de entrada, que procesa mediante el enfoque no cooperativo. Después utiliza el árbol con mayor bienestar social encontrado durante esta etapa como entrada para la siguiente fase.

Algoritmo 4 M-CBAPM-BSW(G)

```

1:  $T \leftarrow \text{CREAR-ARBOL}(G, r)$ 
2:  $T \leftarrow \text{ETIQUETADO-INICIAL}(T)$ 
3:  $T \leftarrow \text{GENERAR-FUNCIONES-GANANCIA}(T)$ 
4:  $A \leftarrow \text{COPIAR-ARBOL}(T)$ 
5: mientras  $\exists u$  tal que  $u \leftarrow \text{CALENDARIZADOR}(G, T)$  hacer
6:    $P_u^{old} \leftarrow P_u$ 
7:    $T' \leftarrow \text{MOVIMIENTO-OPTIMIZACION}(G, T, u)$ 
8:    $T \leftarrow \text{ACTUALIZACION}(T', u, P_u^{old})$ 
9:   si  $g(T) > g(A)$  entonces
10:     $A \leftarrow \text{COPIAR-ARBOL}(T)$ 
11:   fin si
12: fin mientras
13:  $T \leftarrow \text{COPIAR-ARBOL}(A)$ 
14:  $T \leftarrow \text{CAMBIAR-FUNCIONES-GANANCIA}(T)$ 
15: mientras  $\exists u$  tal que  $u \leftarrow \text{CALENDARIZADOR}(G, T)$  hacer
16:    $P_u^{old} \leftarrow P_u$ 
17:    $T' \leftarrow \text{MOVIMIENTO-OPTIMIZACION}(G, T, u)$ 
18:    $T \leftarrow \text{ACTUALIZACION}(T', u, P_u^{old})$ 
19: fin mientras

```

4.2.2. Procedimiento ETIQUETADO-INICIAL

Este procedimiento se define en la Sección 3.1.2. El tiempo de ejecución de este procedimiento es $\Theta(n)$ u.t.

4.2.3. Procedimiento GENERAR-FUNCIONES-GANANCIA()

Este procedimiento se define en la Sección 3.1.3. En el caso del algoritmo CBAPM Mixto, este procedimiento construye las funciones de ganancia π_u de cada vértice u utilizando el enfoque no cooperativo. El tiempo de ejecución de este procedimiento es $O(n + m)$ u.t.

4.2.4. Procedimiento COPIAR-ARBOL()

Este procedimiento recibe como entrada un árbol T y su salida es un árbol que contiene una estructura idéntica al árbol que recibió de entrada. Este procedimiento toma $O(n)$ u.t.

4.2.5. Procedimiento CALENDARIZADOR()

Este procedimiento se define en la Sección 3.1.4. El tiempo de ejecución de este procedimiento es $O(n + m)$ u.t.

4.2.6. Procedimiento MOVIMIENTO-OPTIMIZACION()

Este procedimiento se define en la Sección 3.1.5. El tiempo de ejecución de este procedimiento es $\Theta(n)$ u.t.

4.2.7. Procedimiento ACTUALIZACION()

Este procedimiento se define en la Sección 3.1.6. El tiempo de ejecución de este procedimiento es $O(n + m)$ u.t.

4.2.8. Procedimiento CAMBIAR-FUNCIONES-GANANCIA()

Este procedimiento se define en la Sección 4.1.7. El tiempo de ejecución de este procedimiento es $O(n + m)$ u.t.

4.3. Algoritmo CBAPM Mixto doble

Otra variante del algoritmo CBAPM mixto propone realizar dos veces cada enfoque, empezando por el no cooperativo, continua con el cooperativo, regresa nuevamente al no cooperativo y finaliza con el cooperativo. Se denomina a esta variante, algoritmo CBAPM Mixto doble (DM-CBAPM).

La premisa detrás de esta variante es que durante el segundo ciclo no cooperativo más vértices que siguen atrapados puedan colocarse a su distancia óptima, o cercana al óptimo, de la raíz gracias a elecciones egoistas. Durante la tercera etapa los vértices que realizan elecciones egoistas tienen más opciones para elegir vértices a los cuales conectarse. Este razonamiento cobra sentido si se piensa en los vértices que desean estar en posiciones alejadas de la raíz; es posible que el árbol inicial no tenga siquiera la profundidad que genere las distancias que estos vértices buscan.

El diagrama de bloques del algoritmo DM-CBAPM se muestra en la Figura 12.

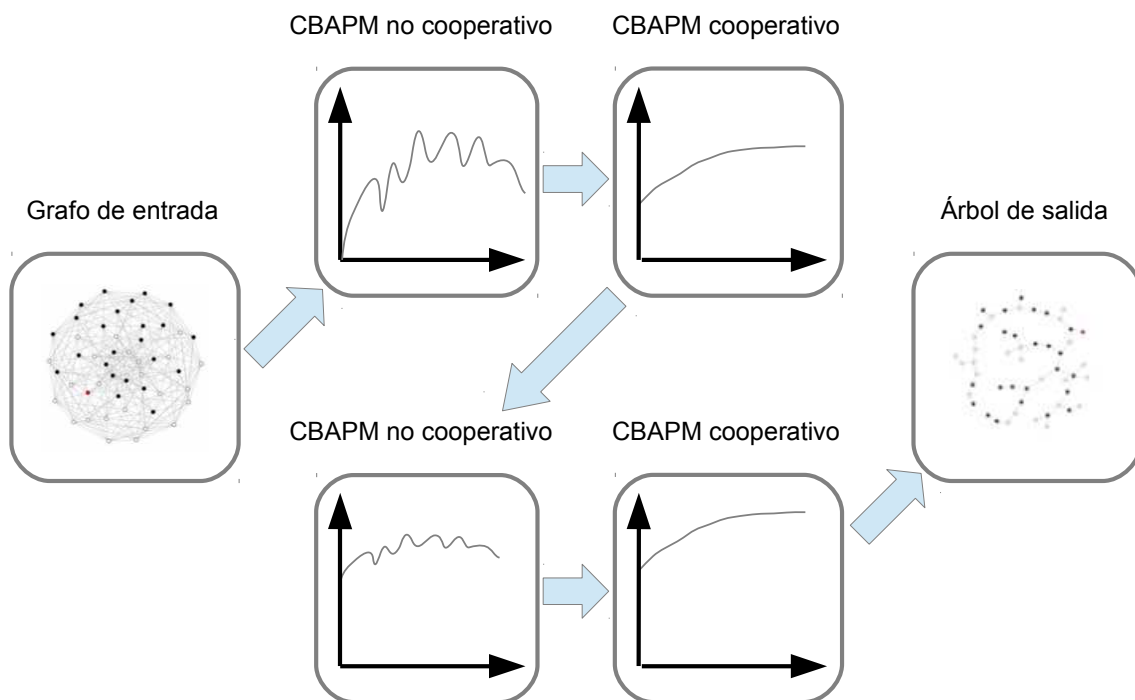


Figura 12: Diagrama de bloques del algoritmo DM-CBAPM. El algoritmo inicializa las etapas cooperativas a partir de los equilibrios encontrados en las etapas no cooperativas.

Capítulo 5. Resultados experimentales

En este capítulo se presentan los resultados de las simulaciones experimentales de los algoritmos CBAPM y DBAPM con el objetivo de comparar los resultados obtenidos de los enfoques cooperativo, no cooperativo y mixto. Se realizaron simulaciones de los algoritmos CBAPM y DBAPM en grafos generados de forma aleatoria utilizando los mismos modelos y densidades que se utilizaron en el PAG (Fajardo-Delgado *et al.*, 2013), así como también se utilizó la metodología utilizada en el análisis del PAG (Fajardo-Delgado *et al.*, 2013). Sin embargo, dado que el PAGM es un problema más amplio que el PAG, la batería de experimentos para el PAGM también es más extensa, e incluye categorías de grafos que no se utilizaron en el PAG.

5.1. Generación de casos específicos del PAGM

A continuación se describen los procedimientos y modelos mediante los cuales se generaron los casos específicos de prueba del PAGM.

5.1.1. Distribuciones de clases

En el PAG solo existen tres tipos de vértices: la raíz, vértices blancos y vértices negros. En (Fajardo-Delgado *et al.*, 2013) se llevaron a cabo simulaciones experimentales donde se generaron grafos aleatorios con 65 vértices, donde uno era la raíz, 32 eran vértices negros y 32 vértices blancos.

En el caso del PAGM es necesario definir antes, los posibles escenarios respecto a las clases de vértices que puede haber en un caso específico. Se proponen cinco diferentes escenarios que representan una cierta variedad de casos de distribución de clases en casos específicos del PAGM, para grafos con 65 vértices:

1. **Escala completa.** En un grafo de 65 vértices existe un vértice que es la raíz del juego PAGM y existe un vértice por cada nivel de preferencia de la función $k(u)$ desde 1 hasta 64. Esto es, existe un solo vértice u con una preferencia $k(u) = 1$, existe un solo vértice v con una preferencia $k(v) = 2$, y así sucesivamente hasta el vértice x con una $k(x) = 64$.

2. **Pares.** En un grafo de 65 vértices existe un vértice que es la raíz del juego PAGM y existen dos vértices por cada nivel de preferencia par y ningún vértice para alguna preferencia impar; i.e., existen dos vértices, u y v para los cuales $k(u) = k(v) = 2$, existen dos vértices x y y para los cuales $k(x) = k(y) = 4$, y así sucesivamente hasta los vértices a y b con $k(a) = k(b) = 64$; no existen vértices con una función k igual a un número impar.
3. **1, 10, 25, 50.** En un grafo de 65 vértices existe un vértice que es la raíz del juego PAGM, y existen cuatro clases diferentes de vértices de acuerdo a su función de preferencia k . Existen en el grafo 16 vértices con una función de preferencia k igual a 1; 16 vértices con una función de preferencia k igual a 10; 16 vértices con una función de preferencia k igual a 25; y 16 vértices con una función de preferencia k igual a 50.
4. **1, 2, $n - 2$, $n - 1$.** En un grafo de 65 vértices existe un vértice que es la raíz del juego PAGM y existen cuatro clases diferentes de vértices respecto a su función de preferencia k . Las preferencias de estos grupos de vértices se encuentran en los extremos del intervalo de tal manera que existe un grupo compuesto por 16 vértices con una función de preferencia k igual a 1; un grupo compuesto por 16 vértices con una función de preferencia k igual a 2; un grupo compuesto por 16 vértices con una función de preferencia k igual a 63; y un grupo de 16 vértices con una función de preferencia k igual a 64. Cabe destacar que este escenario es muy similar a los que representa el PAG.
5. **Aleatorio.** En un grafo de 65 vértices existe un vértice que es la raíz. El resto de los vértices tienen una función de preferencia k que puede tomar cualquier valor dentro del intervalo 1 a 64.

5.1.2. Generación de grafos aleatorios

Existen diferentes modelos para la generación de grafos aleatorios. En los trabajos de Fajardo-Delgado *et al.* (2013), Brubeck Salcedo (2011) y Zatarain Aceves (2011) se utilizaron tres modelos diferentes para la experimentación de sus algoritmos propuestos:

Barabási-Albert (Barabási y Albert, 1999), Erdős-Rényi (Erdos y Rényi, 1960) y grafos crecidos aleatoriamente (Callaway *et al.*, 2001). Estos tres tipos de grafos tienen propiedades que los hacen diferentes entre sí, a pesar de que todos sean generados de manera aleatoria.

5.1.2.1. Modelo Barabási-Albert

El modelo Barabási-Albert (Barabási y Albert, 1999) genera redes “libres de escala” mediante el mecanismo de conexión preferencial. Las redes libres de escala son aquellas en las que unos pocos vértices tienen un alto grado de conexiones, mientras que la mayoría de los vértices tienen sólo pocas conexiones. Las redes libres de escala se han observado tanto en la naturaleza como en los sistemas realizados por el hombre, como por ejemplo, el Internet.

La red se inicializa con un pequeño número de vértices m_0 . A cada instante de tiempo se agrega un vértice que realizará $m \leq m_0$ conexiones con los vértices ya existentes en la red. Para determinar con quién realizará conexiones cada nuevo vértice, se utiliza el concepto de conexión preferencial. Bajo este concepto, los vértices con mayor cantidad de conexiones tendrán mayor probabilidad de ser elegidos por los nuevos vértices para realizar nuevas conexiones. La probabilidad de que un vértice v sea elegido para realizar conexión con un vértice nuevo está dada por la Ecuación 27.

$$Pr = \frac{k_v}{\sum_i k_j} + z_{appeal} \quad (27)$$

Donde k_v es el número de conexiones del vértice v (que no haya generado el mismo). La variable z_{appeal} es una constante que otorga la posibilidad de generar conexiones a los vértices que aún no hayan sido elegidos para generar conexiones por ningún vértice. Los grafos generados mediante este modelo se caracterizan por tener forma de estrellas conectadas. La Figura 13 muestra un grafo generado por este método de 65 vértices con 128 aristas.

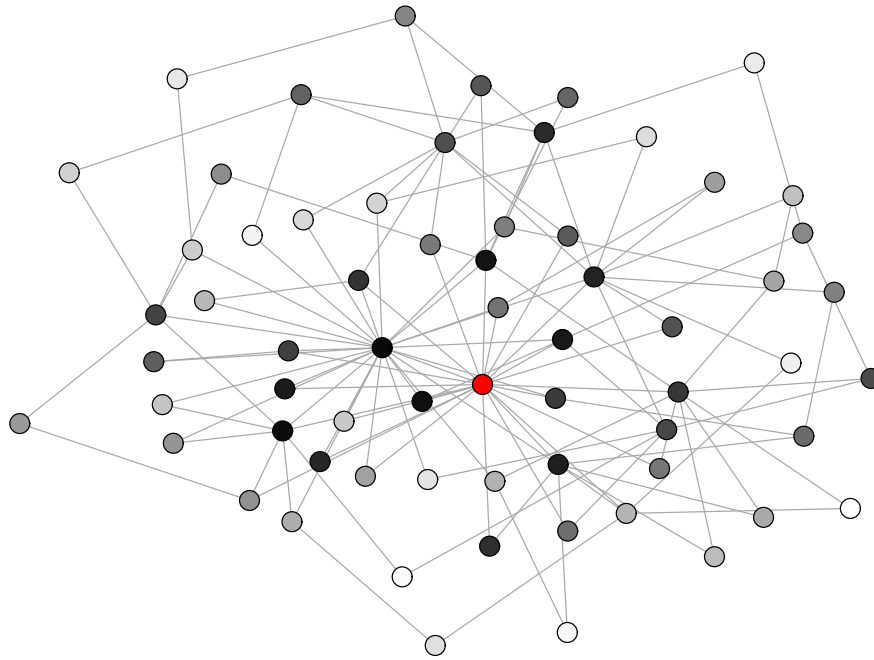


Figura 13: Grafo aleatorio de 65 vértices y 128 aristas generado por el modelo Barabási-Albert. Este grafo pertenece a la distribución de escala completa.

5.1.2.2. Modelo Erdős-Rényi

El modelo de generación de grafos Erdős-Rényi en realidad se refiere a dos modelos diferentes. El primero, llamado modelo $G(n, p)$, consiste en construir un grafo de manera aleatoria al seleccionar la cantidad de vértices, n , en el grafo y después cada posible arista se agrega con una probabilidad independiente p . En el segundo modelo, denominado $G(n, m)$, también se determina la cantidad de vértices, n , en el grafo y después se agregan m aristas de manera aleatoria uniforme. Dado que los casos específicos del PAGM se definen a partir de la densidad de los grafos, se utiliza el modelo $G(n, m)$. La Figura 14 muestra un grafo aleatorio de $n = 65$ vértices y $m = 128$ aristas generado por el modelo Erdős-Rényi $G(n, m)$.

5.1.2.3. Modelo evolutivo estocástico

El modelo evolutivo estocástico (Callaway *et al.*, 2001) crea un grafo mediante la simulación de un crecimiento aleatorio. Se utiliza un modelo de tiempo discreto, donde en cada paso se agrega un nuevo vértice y m nuevas aristas. Mediante un parámetro de control se puede establecer si las aristas generadas conectan dos vértices seleccionados de manera aleatoria o bien si al menos uno de estos vértices es el nuevo vértice generado

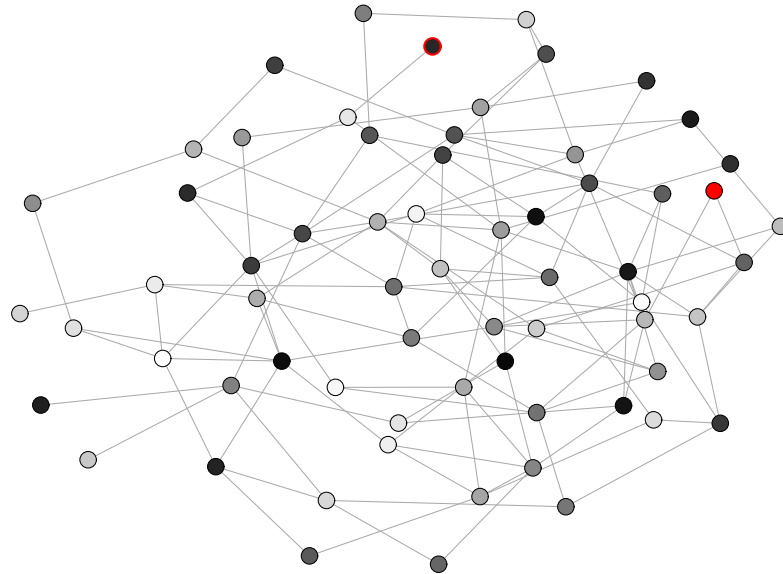


Figura 14: Grafo aleatorio de 65 vértices y 128 aristas generado por el modelo Erdős-Rényi. Este grafo pertenece a la distribución de escala completa.

en el último paso. De no utilizar esta característica, es posible generar grafos desconectados, por lo que para generar los casos de prueba del PAGM siempre se generan las conexiones a partir del último vértice agregado. Además, este modelo puede incurrir en generar aristas duplicadas, por lo cual es necesario utilizar un proceso para eliminar las aristas repetidas. La Figura 15 muestra un grafo aleatorio de 65 vértices y 128 aristas generado por el modelo evolutivo estocástico.

5.2. Metodología

Se utilizó la misma metodología propuesta por Zatarain Aceves (2011) que consistió en emplear el software estadístico R (R Core Team, 2014) y la librería *igraph* (Csardi y Nepusz, 2006) para generar grafos con 65 vértices de forma aleatoria. Por cada una de las cinco clases de distribuciones se generaron tres conjuntos de grafos, uno de cada modelo que se describe en la Sección 5.1.2. Para cada uno de estos tres modelos se generaron grafos con cinco diferentes cantidades de aristas: 128, 256, 512, 1024, 2048 (un grafo completo de 65 vértices tiene 2080 aristas). Por cada densidad se generaron 30 casos específicos. En total se tienen 75 categorías diferentes, cada una con 30 casos específicos.

La metodología de experimentación consiste en ejecutar el algoritmo CBAPM, o DBAPM,

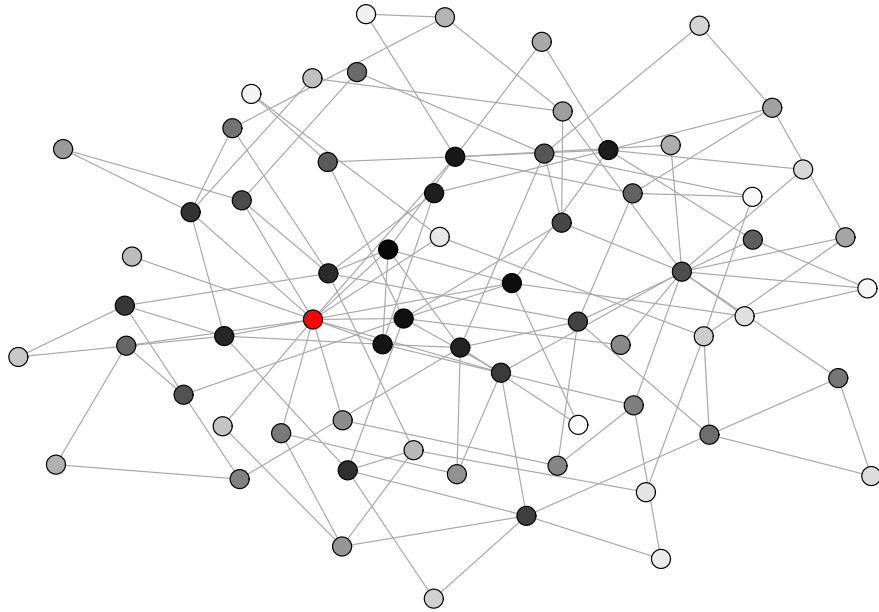


Figura 15: Grafo aleatorio de 65 vértices y 128 aristas generado por el modelo de crecimiento evolutivo estocástico. Este grafo pertenece a la distribución de escala completa.

tanto con el enfoque cooperativo como con el no cooperativo, para cada categoría, y comparar entre ambos enfoques. Dado que los algoritmos CBAPM y DBAPM toman decisiones basándose en generadores de número aleatorios (generación del árbol inicial mediante PrimRST y política de selección uniforme de la función CALENDARIZADOR), se llevan a cabo 30 simulaciones por cada caso específico propuesto. Para determinar si los resultados de un enfoque son mejores a los resultados del otro, se juntan los resultados de la categoría y se aplican las siguientes pruebas estadísticas:

1. Se aplica la prueba de Lilliefors (Lilliefors, 1967) para determinar si las muestras de cada enfoque provienen de distribuciones normales.
2. Dependiendo si la distribución es normal o no, se analiza para buscar si existe una diferencia significativa mediante alguna de las siguientes dos pruebas:
 - Si las dos muestras tienen una distribución normal, se aplica la prueba de t-student
 - Si al menos una de las muestras no tiene una distribución normal, se aplica prueba de Wilcoxon (Wilcoxon, 1945) para una y dos colas.

Para todas las pruebas se utilizó un nivel de confianza de 0.95.

Tabla 2: Los óptimos exactos y aproximados para las diferentes distribuciones de clases en las que se evaluaron los algoritmos propuestos. El símbolo * indica que se trata del óptimo exacto.

Distribución	Máximo bienestar social
Escala completa	4096*
Preferencia par	4064*
4 grupos de vértices	3817
Vértices en los extremos	3135
Completamente aleatorio	4027

Los parámetros que se comparan son el *el precio de estabilidad* (PoS), el *precio de anarquía* (PoA), y la cantidad de movimientos realizados que requiere el algoritmo en encontrar una configuración de equilibrio. El PoS (PoA) representa la calidad del mejor (peor) equilibrio encontrado por el algoritmo. Las ecuaciones 28 y 29 definen el PoS y PoA, respectivamente.

$$\text{PoS} = \frac{\text{bienestar social máximo}}{\text{bienestar social óptimo}} \quad (28)$$

$$\text{PoA} = \frac{\text{bienestar social mínimo}}{\text{bienestar social óptimo}} \quad (29)$$

El beneficio social óptimo se determinó de manera teórica, suponiendo un grafo completo, para dos de los cinco escenarios propuestos; para el resto de los escenarios se eligió el mayor valor encontrado por el algoritmo en cualquiera de ambos esquemas como *aproximadamente óptimo*. En la Tabla 2 se muestran los valores óptimos para las diferentes distribuciones de clases. Es importante notar que en un caso específico que no incluye un grafo completo, pueden no existir las aristas necesarias para llegar a estos óptimos.

El objetivo de estas comparaciones es responder a las preguntas ¿Qué enfoque (cooperativo o no cooperativo) es, en general, mejor?, ¿Bajo qué condiciones un enfoque es mejor que otro?, ¿Qué ventajas (calidad de los equilibrios, tiempo de ejecución, etc.) tiene un enfoque sobre el otro?

Por último, se llevaron a cabo simulaciones experimentales y las pruebas estadísticas siguiendo esta misma metodología, para comparar el algoritmo CBAPM bajo el enfoque

cooperativo y no cooperativo con el algoritmo CBAPM mixto.

5.3. Resultados

En esta sección se presentan los resultados de la ejecución de los algoritmos CBAPM, DBAPM y CBAPM mixto para los casos específicos propuestos. Se utiliza el PoS y el PoA como medidas de comparación entre los enfoques no cooperativo, cooperativo y mixto. También se analiza la cantidad de movimientos que realizan los algoritmos CBAPM y DBAPM bajo los enfoques no cooperativo y cooperativo.

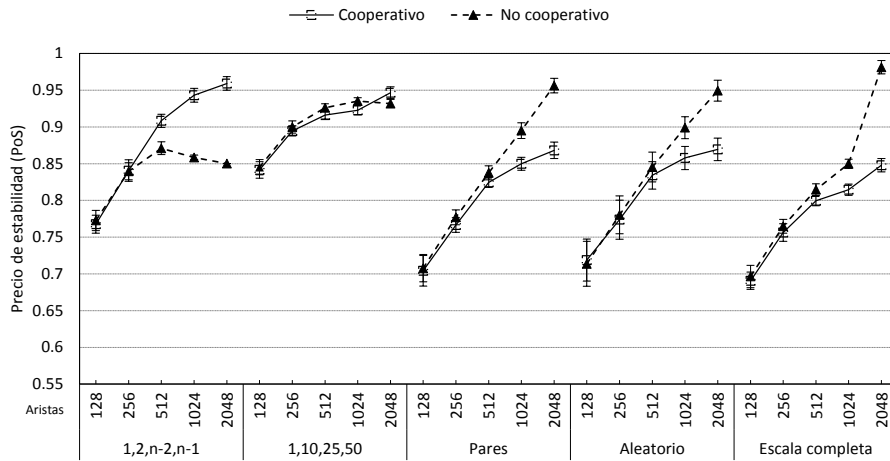
5.3.1. CBAPM

5.3.1.1. PoS aproximado

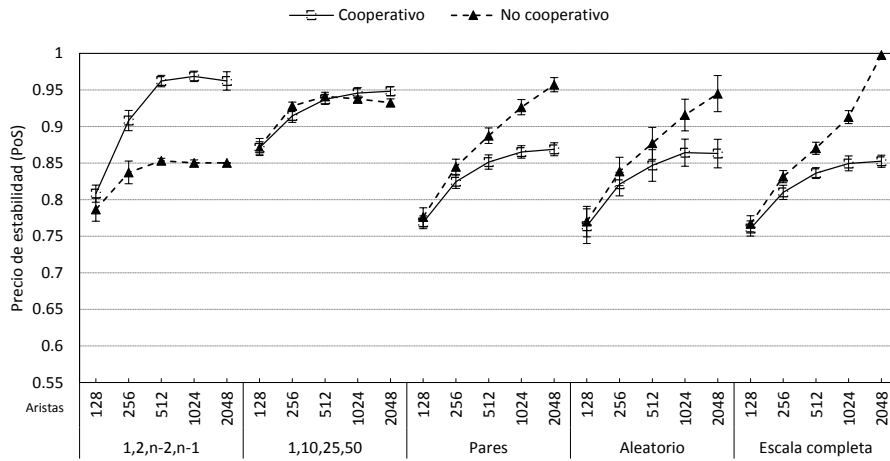
La Figura 16 muestra los PoS aproximados promedio obtenidos por el CBAPM para los diferentes casos específicos propuestos, tanto bajo el enfoque cooperativo, como bajo el enfoque no cooperativo.

Los resultados obtenidos muestran que, en general, para todas las distribuciones propuestas, así como para ambos enfoques, cuando los grafos son más densos se obtiene un mejor PoS. Esto se debe a que si existe un mayor grado de interconexión entre todos los vértices, es más probable que cada vértice pueda ubicarse en la posición de su preferencia. Una notable excepción a este comportamiento es en la distribución $(1, 2, n - 2, n - 1)$, bajo el enfoque no cooperativo, caso en el cual el PoS se mantiene relativamente constante sin importar la densidad del grafo, e incluso puede llegar a decrementar a mayor densidad del grafo, como se puede observar en la Figura 16a. Este comportamiento ya se había observado en el trabajo realizado por (Fajardo, 2012) para el PAG. La explicación para este comportamiento radica en que en grafos dispersos, es más probable que vértices con preferencias cercanas a la raíz ($k(u) = 1, k(u) = 2$) queden alejados de la raíz bajo el enfoque no cooperativo por falta de rutas cortas hacia la raíz (cosa que no sucede en grafos densos), y los vértices que desean alejarse de la raíz ($k(u) = n - 2, k(u) = n - 1$) resultan beneficiados si se conectan estos vértices que no pudieron acercarse más.

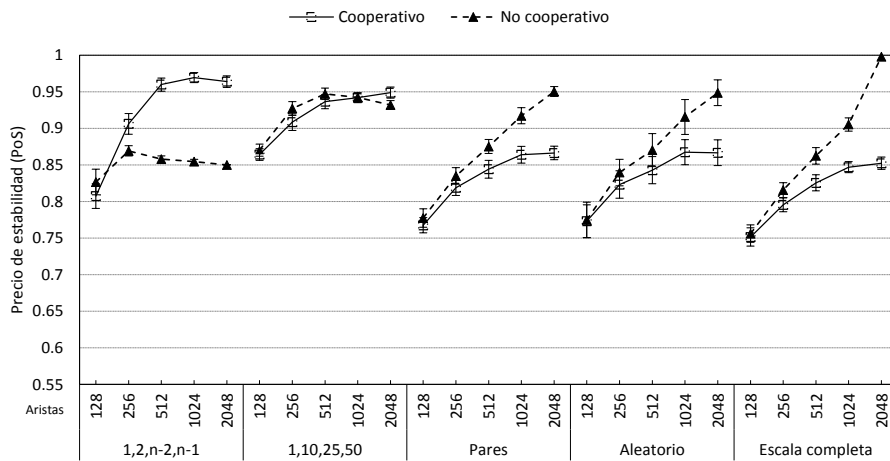
Se observa en las figuras 16a, 16b, y 16c, que el PoS obtenido para las diferentes distribuciones es similar a través de los tres diferentes modelos de grafos aleatorios con



(a) Barabási-Albert



(b) Erdős-Rényi



(c) Evolutive estocástico

Figura 16: Valores experimentales del precio de estabilidad del algoritmo CBAPM de acuerdo a diferentes distribuciones de clases. (a) El PoS de grafos Barabási-Albert. (b) El PoS de grafos Erdős-Rényi. (c) El PoS de grafos generados mediante el modelo evolutivo estocástico.

Tabla 3: Resultados PoS del CBAPM. Por cada densidad y tipo de grafo se indica cuál enfoque es mejor (Coop = cooperativo, No coop = no cooperativo y n/a = no existe diferencia significativa).

	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	Barabási-Albert	128	n/a	n/a	n/a	n/a
256		n/a	No coop	No coop	n/a	No coop
512		Coop	No coop	No coop	No coop	No coop
1024		Coop	No coop	No coop	No coop	No coop
2048		Coop	Coop	No coop	No coop	No coop
Densidad		1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
Erdős-Rényi	128	Coop	n/a	No coop	n/a	No coop
	256	Coop	No coop	No coop	No coop	No coop
	512	Coop	n/a	No coop	No coop	No coop
	1024	Coop	Coop	No coop	No coop	No coop
	2048	Coop	Coop	No coop	No coop	No coop
	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
Evolutivo estocástico	128	No coop	No coop	No coop	n/a	n/a
	256	Coop	No coop	No coop	No coop	No coop
	512	Coop	No coop	No coop	No coop	No coop
	1024	Coop	n/a	No coop	No coop	No coop
	2048	Coop	Coop	No coop	No coop	No coop
	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa

los que se experimentó, lo que significa que la topología del grafo no es un factor relevante para el rendimiento de los enfoques cooperativo o no cooperativo.

La única distribución en la cual el enfoque cooperativo es claramente superior al no cooperativo es la distribución (1, 2, $n - 2$, $n - 1$). Se recuerda que esta distribución es la más similar a la que maneja el PAG. De hecho, los resultados obtenidos por Fajardo-Delgado *et al.* (2013) concluyen que el enfoque cooperativo obtiene mejores resultados en el PAG, por lo que este nuevo resultado concuerda con los obtenidos por los autores anteriores.

La distribución (1, 10, 25, 50) es la que presenta resultados más distintos al resto de los casos. Para grafos completos, y para los tres modelos de grafos aleatorios, el enfoque cooperativo tiene resultados estadísticamente mejores que el no cooperativo. Sin embargo, para el resto de las densidades, el enfoque no cooperativo tiene mejores PoS, o bien no existe diferencia significativa entre ambos enfoques. En ambos casos, los resultados promedios de ambos enfoques son muy cercanos entre sí.

Para el resto de las distribuciones (escala completa, pares y aleatorio), el enfoque no

cooperativo obtiene un mejor PoS que el cooperativo por un amplio margen, en especial, mientras mayor sea la densidad del grafo.

En la Tabla 3 se muestran los resultados de los análisis estadísticos para el PoS, se indica cuándo un enfoque es significativamente mejor que otro (cooperativo o no cooperativo) o cuando no existe una diferencia significativa (n/a).

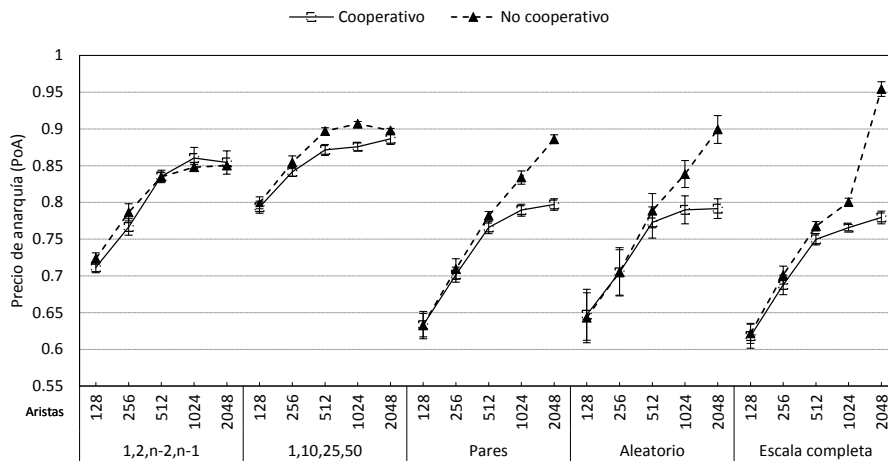
5.3.1.2. PoA aproximado

En la Figura 17 se muestra el PoA aproximado del algoritmo CBAPM para diferentes modelos de grafos aleatorios y diferentes densidades. Los resultados muestran que se siguen las mismas tendencias que el caso del PoS: a mayor sea la densidad del grafo, mayor es el PoA. Esta tendencia es más notable en el caso del enfoque no cooperativo, en las distribuciones pares, aleatorio y escala completa. Para estas tres distribuciones bajo el enfoque cooperativo, también existe un incremento en el PoA a mayor sea la densidad de los grafos; sin embargo, éste no es tan drástico como en el caso del enfoque no cooperativo.

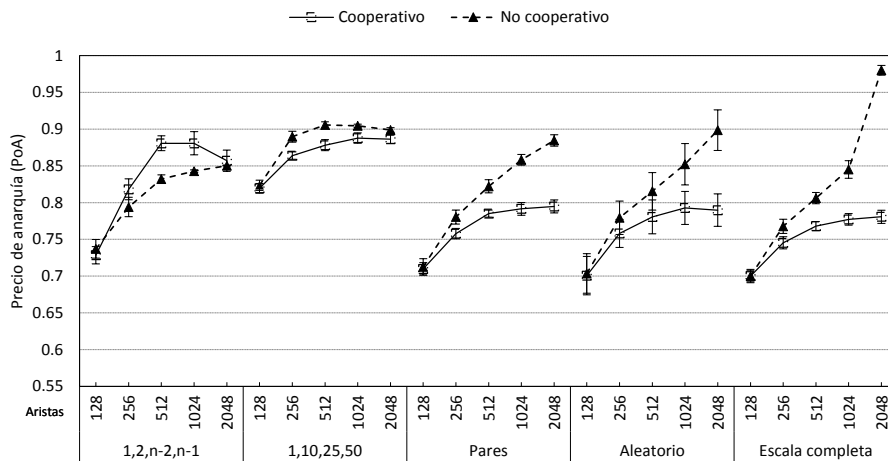
En la distribución $(1, 2, n-2, n-1)$, bajo el enfoque cooperativo, se observa el mismo comportamiento que se presentan en el PoS para esta misma distribución pero bajo el enfoque no cooperativo. En los grafos completos, el valor del PoA disminuye con respecto a los de menor densidad, en vez de incrementarse, como sucede en otras distribuciones.

En la Tabla 4 se observa que en el caso de la distribución $(1, 2, n-2, n-1)$, el enfoque cooperativo es mejor en la mayoría de los casos, aunque también existen escenarios (grafos dispersos) en donde el enfoque no cooperativo es estadísticamente mejor que el cooperativo. Este caso es diferente del PoS, en el cual el enfoque cooperativo siempre era mejor para todos los casos de la distribución $(1, 2, n-2, n-1)$, o al menos, no existía diferencia significativa entre ambos enfoques.

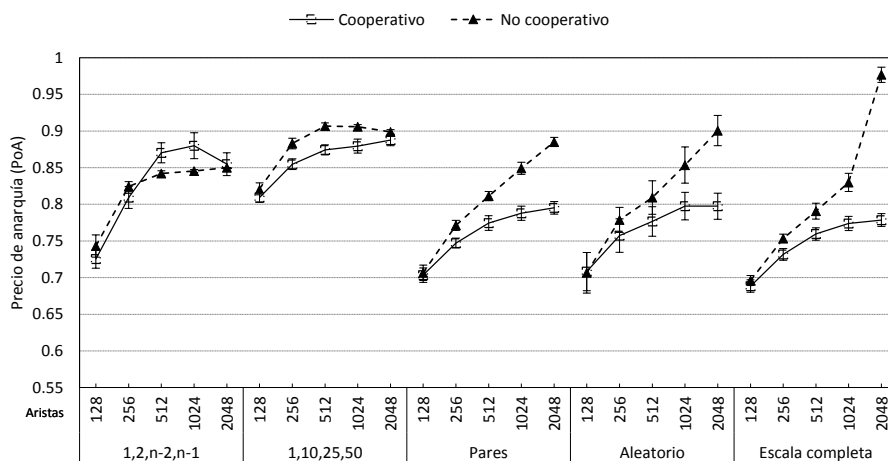
En la distribución $(1, 10, 25, 50)$, el enfoque no cooperativo obtiene PoA mayores que el enfoque cooperativo, para todos los casos. Esto contrasta con los resultados obtenidos para el PoS en esta distribución, donde sí existen casos (grafos densos) donde el enfoque cooperativo tiene PoS más altos que el no cooperativo, así como también hay casos



(a) Barabási-Albert



(b) Erdős-Rényi



(c) Evolutive estocástico

Figura 17: Valores experimentales del precio de anarquía del algoritmo CBAPM de acuerdo a diferentes distribuciones de clases. (a) El PoA de grafos Barabási-Albert. (b) El PoA de grafos Erdős-Rényi. (c) El PoA de grafos generados mediante el modelo evolutivo estocástico.

Tabla 4: Resultados PoA del CBAPM. Por cada densidad y tipo de grafo se indica cuál enfoque es mejor (Coop = cooperativo, No coop = no cooperativo y n/a = no existe diferencia significativa).

	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	Barabási-Albert	128	No coop	No coop	n/a	n/a
256		No coop	No coop	n/a	n/a	No coop
512		n/a	No coop	No coop	No coop	No coop
1024		Coop	No coop	No coop	No coop	No coop
2048		n/a	No coop	No coop	No coop	No coop
Densidad		1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
Erdős-Rényi	128	No coop	No coop	n/a	n/a	n/a
	256	Coop	No coop	No coop	No coop	No coop
	512	Coop	No coop	No coop	No coop	No coop
	1024	Coop	No coop	No coop	No coop	No coop
	2048	Coop	No coop	No coop	No coop	No coop
	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
Evolutivo estocástico	128	No coop	No coop	n/a	n/a	No coop
	256	No coop	No coop	No coop	No coop	No coop
	512	Coop	No coop	No coop	No coop	No coop
	1024	Coop	No coop	No coop	No coop	No coop
	2048	Coop	No coop	No coop	No coop	No coop
	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa

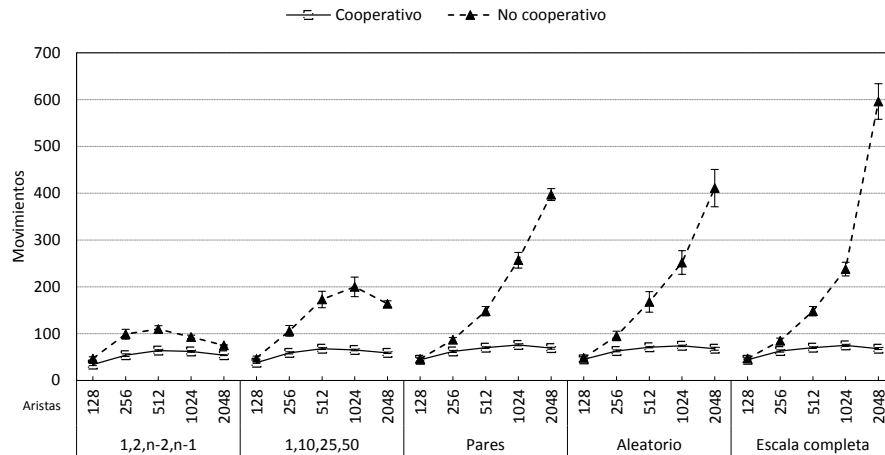
donde no existe una diferencia estadística significativa.

En las distribuciones de escala completa, pares y aleatorio, el enfoque no cooperativo obtiene PoA más altos que el enfoque cooperativo, salvo para algunos grafos de 128 y 256 aristas, en donde no existe diferencia significativa.

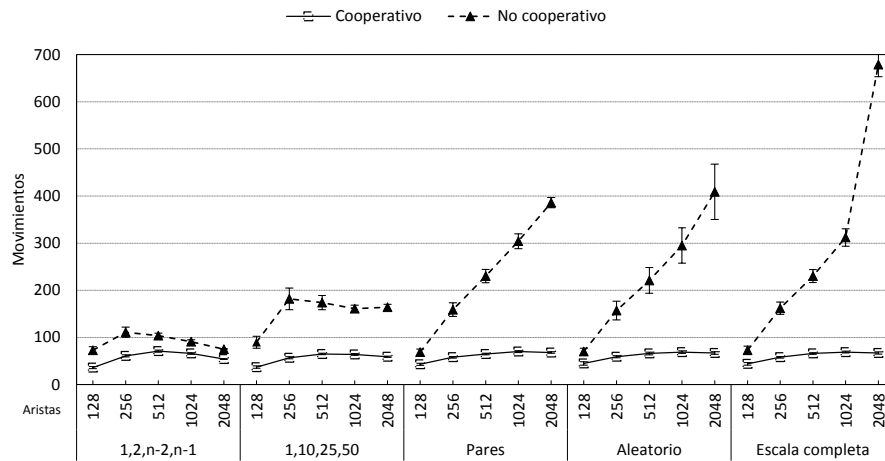
5.3.1.3. Cantidad de movimientos

En la Figura 18 se muestra la cantidad de movimientos promedio que realiza cada enfoque en cada escenario propuesto, para los diferentes modelos de grafos aleatorios. Se observa que ambos enfoques realizan una cantidad de movimientos muy diferente entre sí. Mientras que el enfoque cooperativo realiza una cantidad de movimientos que se mantiene casi constante, con respecto a la densidad o distribución analizada, el enfoque no cooperativo incrementa drásticamente en cantidad de movimientos realizados conforme se incrementa la densidad de los grafos, así como también entre mayor sea la variedad de preferencias de los vértices presentes en los grafos.

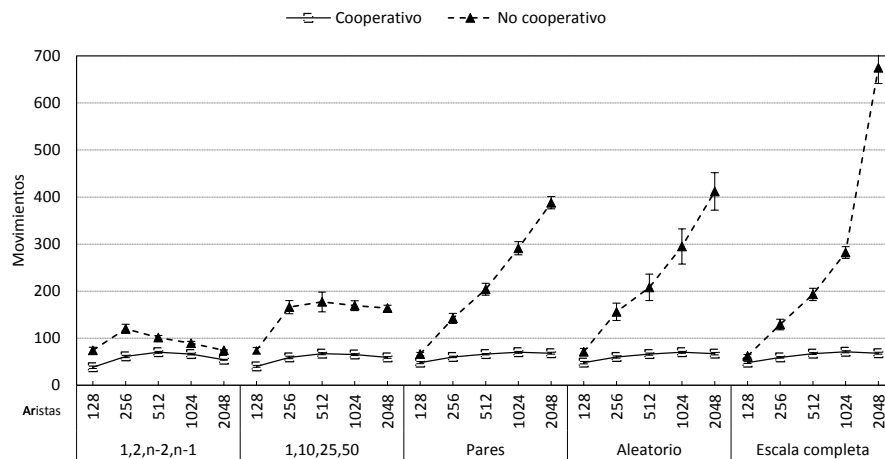
En la distribución (1, 2, $n - 2$, $n - 1$) el enfoque no cooperativo realiza una cantidad de movimientos similar a la cantidad que realiza el enfoque cooperativo. En la distribución



(a) Barabási-Albert



(b) Erdős-Rényi



(c) Evolutivo estocástico

Figura 18: Cantidad promedio de movimientos realizados por el algoritmo CBAPM de acuerdo a diferentes distribuciones de clases. (a) Cantidad de movimientos para grafos Barabási-Albert. (b) Cantidad de movimientos para grafos Erdős-Rényi. (c) Cantidad de movimientos para grafos generados mediante el modelo evolutivo estocástico.

(1, 10, 25, 50) la diferencia en la cantidad de movimientos entre ambos enfoques se acrecenta. Finalmente, en las distribuciones de vértices con preferencia par, aleatoria y escala completa, existe un orden de magnitud de diferencia entre la cantidad de movimientos que realiza cada enfoque.

5.3.2. DBAPM

5.3.2.1. PoS aproximado

Tabla 5: Resultados PoS del DBAP. Para cada densidad y tipo de grafo se indica cuál enfoque es mejor (Coop = cooperativo, No coop = no cooperativo y n/a = no existe diferencia significativa).

	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	Barabási-Albert	128	n/a	n/a	n/a	n/a
256		Coop	Coop	n/a	n/a	Coop
512		Coop	n/a	No coop	No coop	n/a
1024		No coop	Coop	No coop	No coop	n/a
2048		Coop	Coop	No coop	No coop	No coop
Densidad		1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
Erdős-Rényi	128	Coop	n/a	n/a	n/a	n/a
	256	Coop	No coop	No coop	No coop	No coop
	512	Coop	No coop	No coop	No coop	No coop
	1024	n/a	Coop	No coop	No coop	No coop
	2048	Coop	Coop	No coop	No coop	No coop
	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
Evolutivo estocástico	128	n/a	n/a	No coop	n/a	n/a
	256	Coop	No coop	No coop	No coop	n/a
	512	Coop	No coop	No coop	No coop	n/a
	1024	Coop	n/a	No coop	No coop	No coop
	2048	Coop	Coop	No coop	No coop	No coop

En la Figura 19 se muestra el PoS obtenido por el algoritmo DBAPM para los grafos de distintas densidades, modelos y clases de distribuciones. En general, mantienen las mismas tendencias y valores similares a los obtenidos mediante el CBAPM. Este comportamiento no ocurre en el caso de la distribución (1, 2, $n - 2$, $n - 1$) con grafos densos, donde existe una caída en el valor del PoS bajo el enfoque cooperativo con respecto a los valores de grafos dispersos. Este comportamiento ya se había presentado en el PAG, y se debe a la manera en como circula el *token* en el árbol, donde existen vértices que son los últimos en ser escogidos por el calendarizador, y han sido bloqueados por los primeros vértices que llevaron a cabo movimientos.

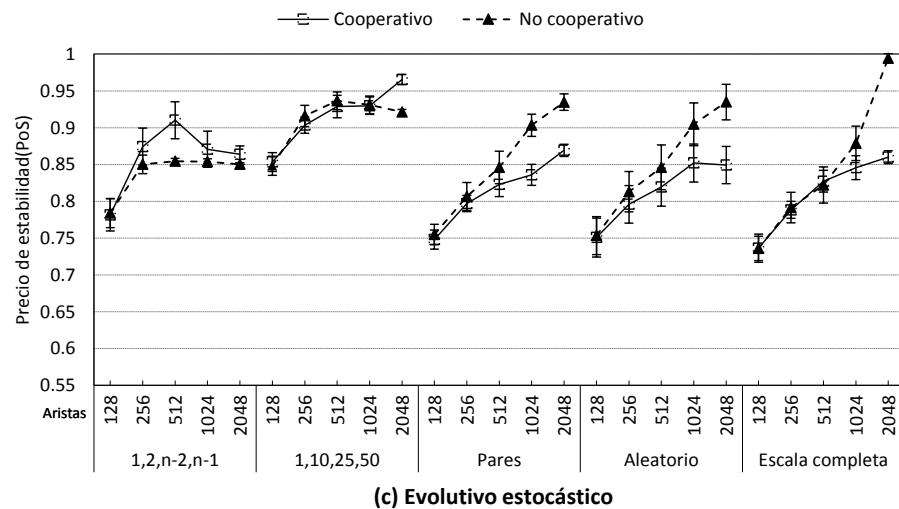
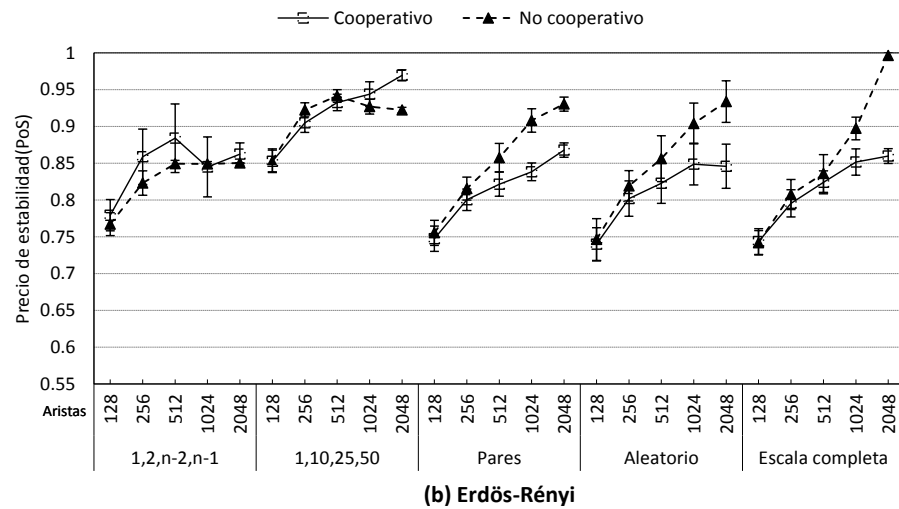
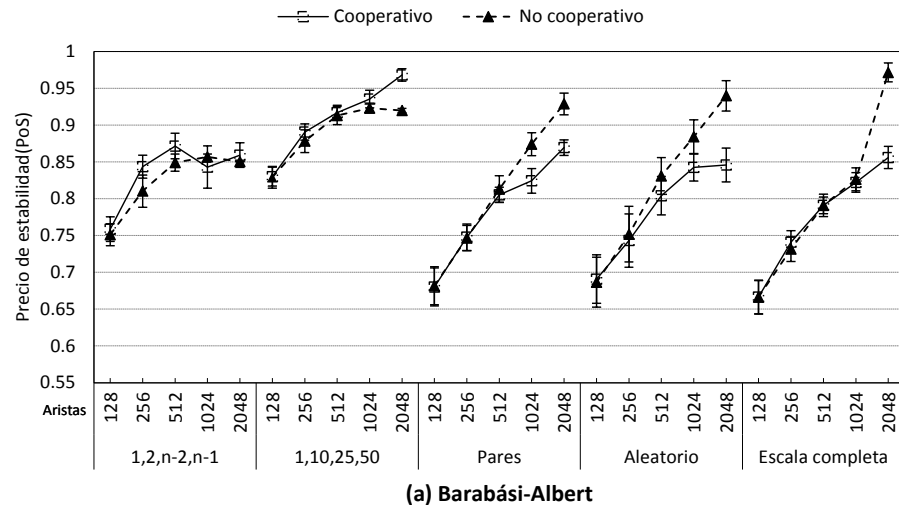


Figura 19: Valores experimentales del precio de estabilidad del algoritmo DBAPM de acuerdo a diferentes distribuciones de clases. (a) El PoS de grafos Barabási-Albert. (b) El PoS de grafos Erdős-Rényi. (c) El PoS de grafos generados mediante el modelo evolutivo estocástico.

En la Tabla 5 se muestran los resultados de los análisis estadísticos para los PoS obtenidos por el DBAPM, se indica cuándo un enfoque es significativamente mejor que otro (cooperativo o no cooperativo) o cuando no existe una diferencia significativa (n/a).

5.3.2.2. PoA aproximado

Tabla 6: Resultados PoA del DBAPM. Para cada densidad y tipo de grafo se indica cuál enfoque es mejor (Coop = cooperativo, No coop = no cooperativo y n/a = no existe diferencia significativa).

Barabási-Albert	Densidad	$1, 2, n - 2, n - 1$	$1, 10, 25, 50$	Pares	Aleatorio	Escala completa
	128	n/a	n/a	n/a	n/a	n/a
	256	No coop	No coop	n/a	n/a	n/a
	512	No coop	No coop	No coop	No coop	No coop
	1024	No coop	No coop	No coop	No coop	No coop
	2048	Coop	Coop	No coop	No coop	No coop
Erdős-Rényi	Densidad	$1, 2, n - 2, n - 1$	$1, 10, 25, 50$	Pares	Aleatorio	Escala completa
	128	Coop	n/a	n/a	n/a	n/a
	256	Coop	No coop	No coop	No coop	n/a
	512	No coop	No coop	No coop	No coop	No coop
	1024	No coop	No coop	No coop	No coop	No coop
	2048	Coop	Coop	No coop	No coop	No coop
Evolutivo estocástico	Densidad	$1, 2, n - 2, n - 1$	$1, 10, 25, 50$	Pares	Aleatorio	Escala completa
	128	n/a	No coop	n/a	n/a	n/a
	256	No coop	No coop	No coop	n/a	No coop
	512	No coop	No coop	No coop	No coop	No coop
	1024	No coop	No coop	No coop	No coop	No coop
	2048	Coop	Coop	No coop	No coop	No coop

En la Figura 20 se muestran los valores del PoA promedio obtenidos por el DBAP para los grafos de diferentes clases de distribución, densidades y modelos de grafos. Los PoA obtenidos por el DBAP siguen las mismas tendencias que los PoA obtenidos por el CBAPM. Una notable excepción es el caso de la distribución $(1, 2, n - 2, n - 1)$, en la cual existe una reducción considerable en el valor del PoA en los grafos de 1024 aristas bajo el enfoque cooperativo. Este comportamiento también se presenta en el PoS obtenido para esta misma categoría de grafos por el DBAPM.

En la Tabla 6 se muestran los resultados de los análisis estadísticos para los PoA obtenidos por el DBAPM. En esta tabla se indica cuándo un enfoque es significativamente mejor que otro (cooperativo o no cooperativo) o cuándo no existe una diferencia significativa (n/a).

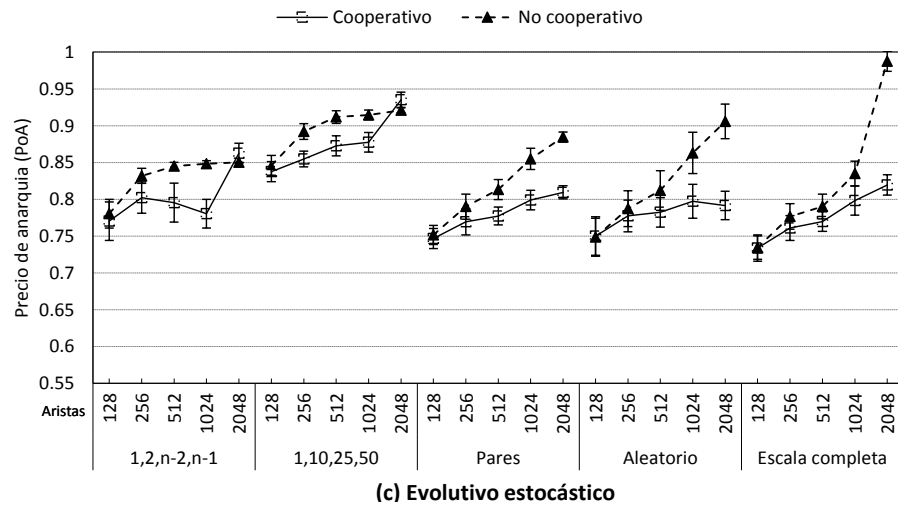
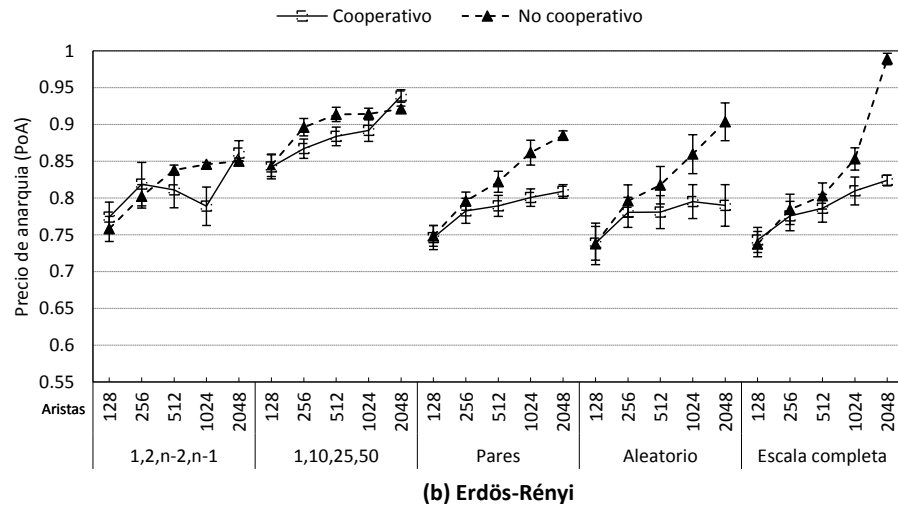
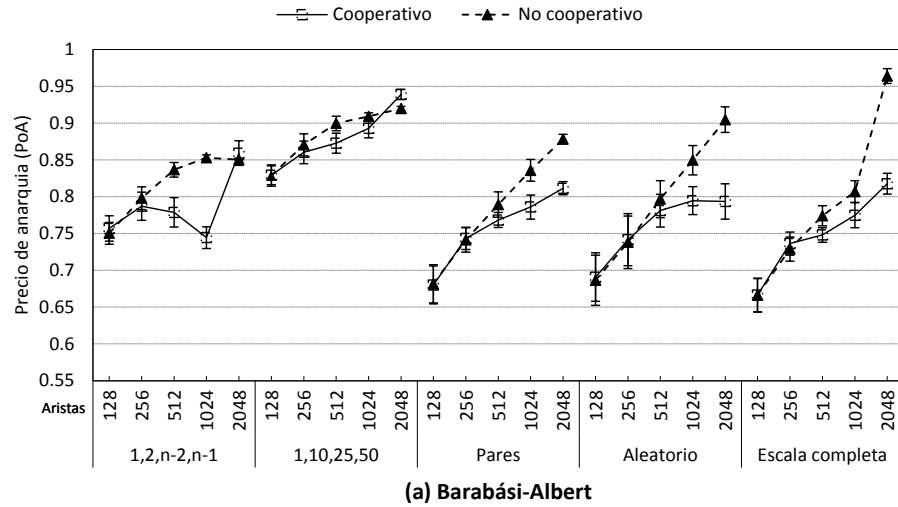


Figura 20: Valores experimentales del precio de anarquía del algoritmo DBAPM de acuerdo a diferentes distribuciones de clases. (a) El PoA para grafos Barabási-Albert. (b) El PoA para grafos Erdős-Rényi. (c) El PoA para grafos generados mediante el modelo evolutivo estocástico.

5.3.2.3. Cantidad de movimientos

La Figura 21 muestra la cantidad promedio de movimientos realizados por el DBAP para los diferentes tipos de grafos, bajo ambos enfoques. Bajo el enfoque no cooperativo, se puede observar un incremento en la cantidad de movimientos realizados conforme se incrementa el número de clases de vértices en los grafos. Bajo este mismo enfoque y en las distribuciones de vértices pares, aleatorio, y escala completa, además existe un incremento en la cantidad de movimientos a mayor sea la densidad de los grafos.

Un resultado muy interesante resulta de contrastar la cantidad de movimientos que realiza el DBAP con respecto a los que realiza el CBAPM, los dos bajo el enfoque no cooperativo, para los grafos densos de las distribuciones pares, aleatorio, y escala completa. Aunque ambos tienen un rendimiento similar en PoS y PoA, el DBAPM realiza una cantidad de movimientos en un orden de magnitud menor que el CBAPM.

5.3.3. Algoritmo mixto

5.3.3.1. PoS aproximado

En esta sección se analizan los resultados del algoritmo CBAPM mixto (M-CBAPM) que se propone en el Capítulo 4; se comparan los valores de los PoS obtenidos por este algoritmo con los obtenidos por el CBAPM bajo los enfoques tanto cooperativo como no cooperativo.

Los resultados de la Tabla 7 muestran que en ningún caso, el M-CBAPM obtiene valores de PoS menores a los obtenidos por CBAPM bajo cualquiera de los dos enfoques que puede utilizar. Sin embargo, existen casos en los que el algoritmo Mixto obtiene resultados que no son significativamente diferentes de los obtenidos por el CBAPM bajo el enfoque no cooperativo. Esto último sucede especialmente en las distribuciones de preferencia par, aleatoria y escala completa.

En la Figura 22 se muestran los valores del PoS promedio obtenidos por el algoritmo M-CBAPM, el CBAPM cooperativo y el CBAPM no cooperativo, para diferentes distribuciones de vértices, densidades de grafos, y modelos aleatorios de grafos.

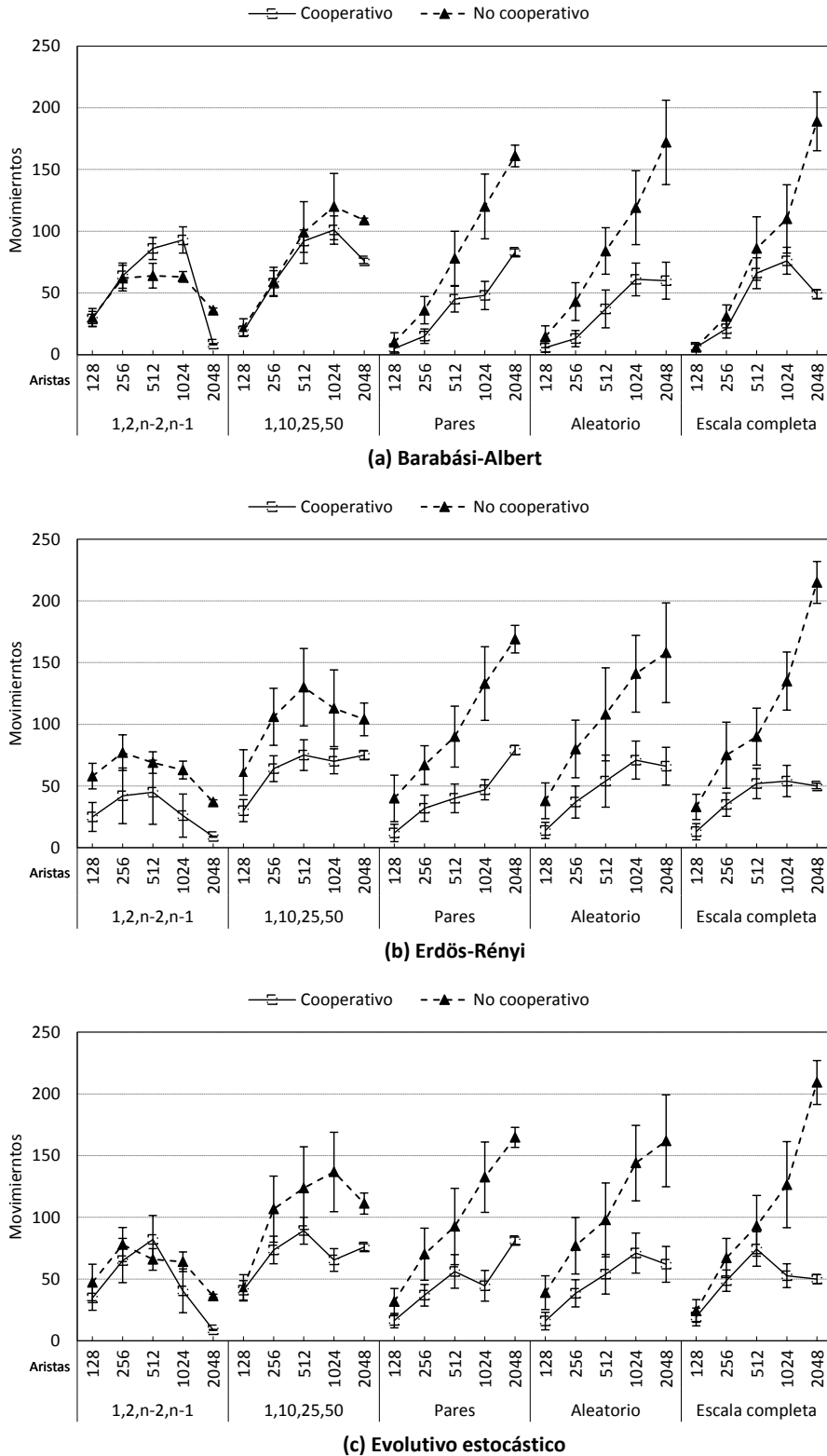


Figura 21: Cantidad promedio de movimientos realizados por el algoritmo DBAPM de acuerdo a diferentes distribuciones de clases. (a) Cantidad de movimientos para grafos Barbási-Albert. (b) Cantidad de movimientos para grafos Erdős-Rényi. (c) Cantidad de movimientos para grafos generados mediante el modelo evolutivo estocástico.

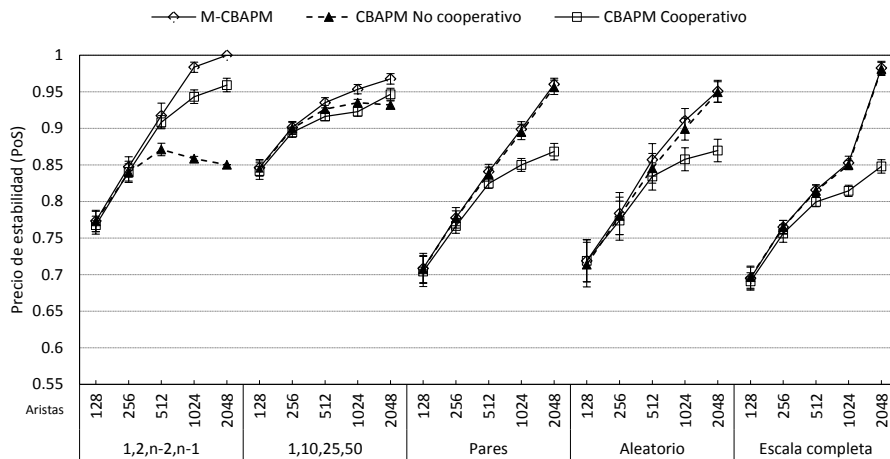
Tabla 7: Resultados PoS del M-CBAPM en comparación con el CBAPM. Para cada densidad y tipo de grafo se indica cuál enfoque es mejor (Coop = CBAPM cooperativo, No coop = CBAPM no cooperativo y Mix = M-CBAPM. El símbolo * indica que no hay diferencia significativa con el CBAPM no cooperativo).

Barabási-Albert	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	n/a	n/a	n/a	n/a	n/a
	256	n/a	Mix*	Mix*	n/a	Mix*
	512	Mix	Mix	Mix*	Mix	Mix*
	1024	Mix	Mix	Mix	Mix	Mix
	2048	Mix	Mix	Mix*	Mix*	Mix*
Erdős-Rényi	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	Mix	Mix	Mix*	Mix*	Mix*
	256	Mix	Mix	Mix	Mix*	Mix*
	512	Mix	Mix	Mix	Mix*	Mix
	1024	Mix	Mix	Mix*	Mix*	Mix*
	2048	Mix	Mix	Mix*	Mix*	Mix*
Evolutivo estocástico	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	Mix*	Mix*	Mix*	n/a	Mix*
	256	Mix	Mix	Mix	Mix*	Mix*
	512	Mix	Mix	Mix	Mix*	Mix*
	1024	Mix	Mix	Mix	Mix*	Mix*
	2048	Mix	Mix	Mix	Mix*	Mix*

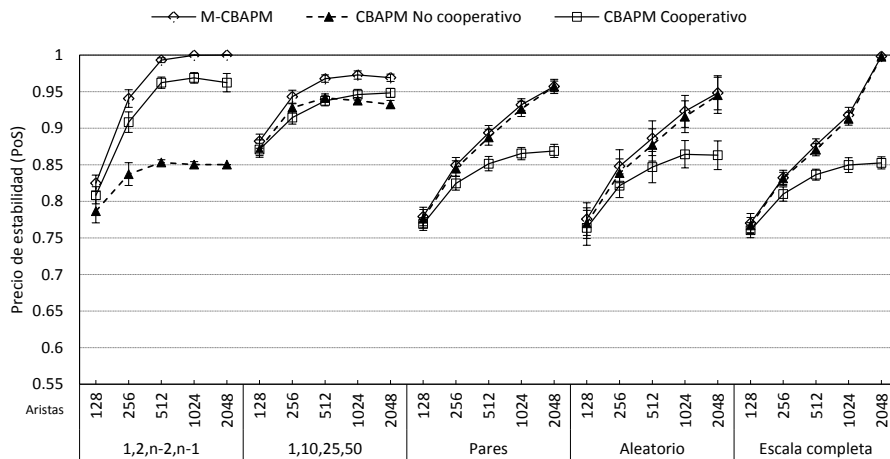
Se puede observar que la distribución $(1, 2, n - 2, n - 1)$, así como la distribución $(1, 10, 25, 50)$, son las que más se benefician de utilizar el enfoque mixto, mientras que en las distribuciones pares, aleatorio, y escala completa, el algoritmo M-CBAPM obtiene resultados muy similares a los que obtiene el CBAPM bajo el enfoque no cooperativo.

Un resultado notable ocurre en el caso de la distribución $(1, 2, n - 2, n - 1)$, donde el algoritmo M-CBAPM incrementa el valor del PoS promedio que se obtiene a medida que se incrementa la densidad de los grafos. Esto es algo que no se presenta en el CBAPM bajo el enfoque no cooperativo, donde el PoS decreta conforme se incrementa la densidad de los grafos. De hecho, el algoritmo M-CBAPM encuentra el valor óptimo, o muy cercanos al óptimo, para esta distribución, en grafos densos.

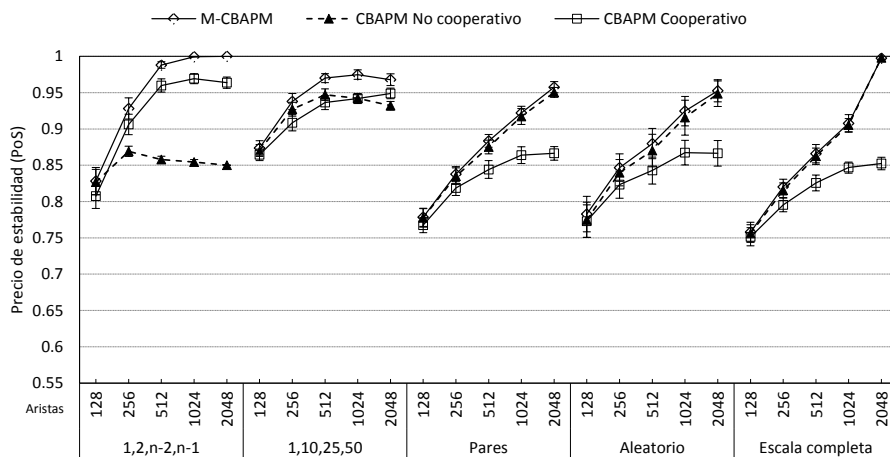
En la distribución $(1, 10, 25, 50)$ existe un ligero decremento en el valor del PoS en los grafos completos, con respecto a los de menor densidad, con excepción del PoS obtenido para esta distribución en el modelo Barabási-Albert. En este último, siempre se incrementa el PoS promedio a mayor sea la densidad de los grafos. Ésta es la única diferencia notable en los resultados obtenidos a través de los tres diferentes modelos de



(a) Barabási-Albert



(b) Erdős-Rényi



(c) Evolutivo estocástico

Figura 22: Valores experimentales del precio de estabilidad de los algoritmos CBAPM y M-CBAPM de acuerdo a diferentes distribuciones de clases. (a) El PoS para grafos Barbási-Albert. (b) El PoS para grafos Erdős-Rényi. (c) El PoS para grafos generados mediante el modelo evolutivo estocástico.

Tabla 8: Resultados PoA del M-CBAPM en comparación con el CBAPM. Para cada densidad y tipo de grafo se indica cuál enfoque es mejor (Coop = CBAPM cooperativo, No coop = CBAPM no cooperativo y Mix = M-CBAPM. El símbolo * indica que no hay diferencia significativa con el CBAPM no cooperativo).

Barabási-Albert	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	Mix*	Mix*	n/a	n/a	n/a
	256	Mix*	Mix*	Mix*	n/a	Mix*
	512	Mix	Mix	Mix*	Mix*	Mix*
	1024	Mix	Mix	Mix	Mix	Mix*
	2048	Mix	Mix	Mix*	Mix*	Mix*
Erdős-Rényi	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	Mix	Mix	Mix*	n/a	Mix
	256	Mix	Mix	Mix*	Mix*	Mix*
	512	Mix	Mix	Mix*	Mix*	Mix*
	1024	Mix	Mix	Mix	Mix	Mix
	2048	Mix	Mix	Mix	Mix*	Mix*
Evolutivo estocástico	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	Mix	Mix*	Mix	n/a	Mix*
	256	Mix	Mix	Mix*	Mix	Mix*
	512	Mix	Mix	Mix*	Mix	Mix*
	1024	Mix	Mix	Mix	Mix	Mix
	2048	Mix	Mix	Mix	Mix	Mix*

grafos aleatorios.

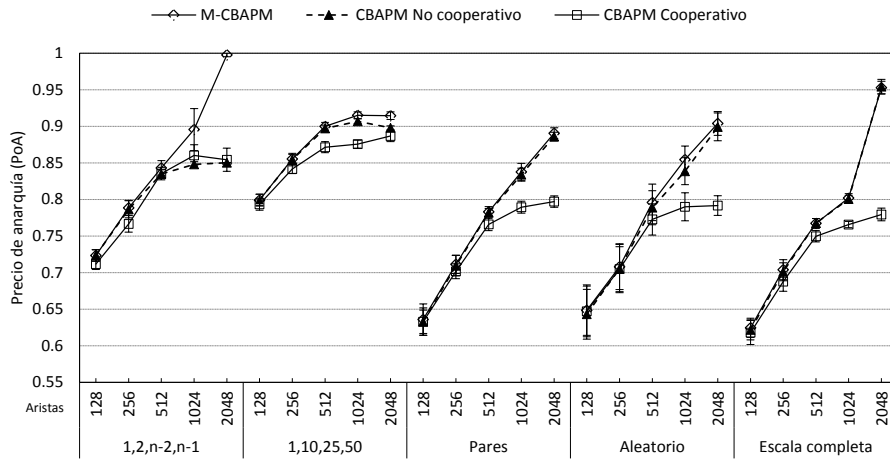
5.3.3.2. PoA aproximado

En esta sección se analizan los resultados del algoritmo M-CBAPM que se propone en el Capítulo 4. Se comparan los valores de los PoA obtenidos por este algoritmo con los obtenidos por el CBAPM bajo los enfoques tanto cooperativo como no cooperativo.

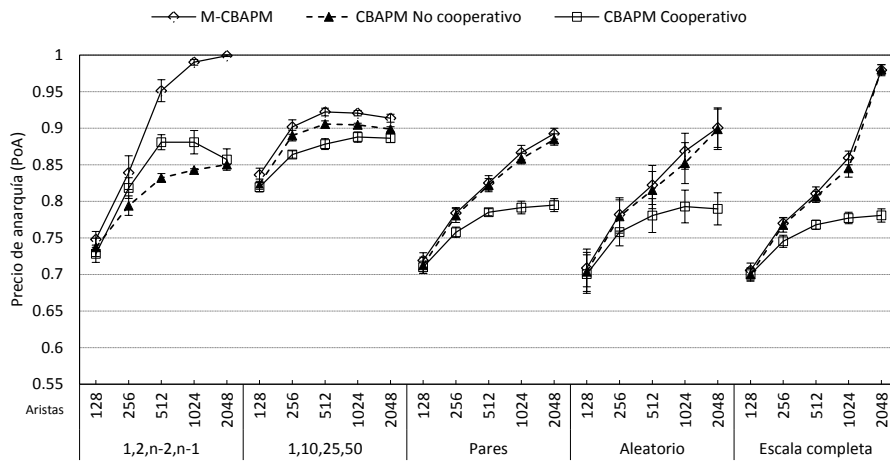
Los resultados de la Tabla 8 muestran que se siguen las mismas tendencias que se observan en el caso del PoS, esto es, en ningún caso, el M-CBAPM obtiene valores de PoA menores a los obtenidos por CBAPM bajo cualquiera de los dos enfoques que puede utilizar.

En la Figura 23 se muestran los valores del PoA promedio obtenidos por el M-CBAPM, el CBAPM cooperativo y el CBAPM no cooperativo, para diferentes distribuciones de vértices, densidades de grafos, y modelos aleatorios de grafos.

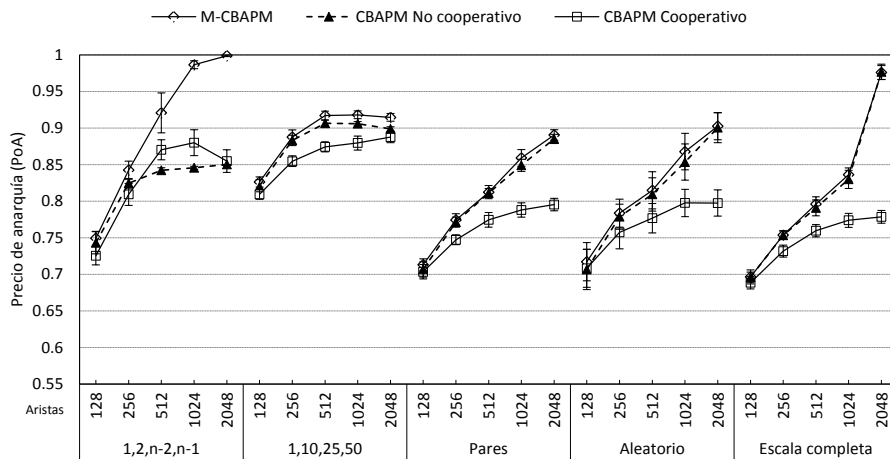
En el caso de la distribución (1, 2, $n - 2$, $n - 1$), el M-CBAPM obtiene un PoA de mayor



(a) Barabási-Albert



(b) Erdős-Rényi



(c) Evolutivo estocástico

Figura 23: Valores experimentales del precio de anarquía de los algoritmos CBAPM y M-CBAPM de acuerdo a diferentes distribuciones de clases. (a) El PoS para grafos Barabási-Albert. (b) El PoS para grafos Erdős-Rényi. (c) El PoS para grafos generados mediante el modelo evolutivo estocástico.

valor al aumentar la densidad de los grafos, a diferencia del CBAPM bajo el enfoque cooperativo, donde en los grafos completos el valor del PoA es menor a aquel que se obtiene de los grafos menos densos.

5.3.3.3. Cantidad de movimientos cooperativos

En la Figura 24 se muestran la cantidad de movimientos que realiza el algoritmo M-CBAPM durante la etapa cooperativa, es decir, los movimientos que realiza bajo el enfoque cooperativo después de haber alcanzado un equilibrio bajo el enfoque no cooperativo.

Se puede observar en la Figura 24 que la mayor cantidad de movimientos se realizan en las distribuciones $(1, 2, n - 2, n - 1)$ y $(1, 10, 25, 50)$, que son aquellas que más se ven beneficiadas en cuanto un incremento en los valores de PoS y PoA, por el algoritmo M-CBAPM con respecto al CBAPM.

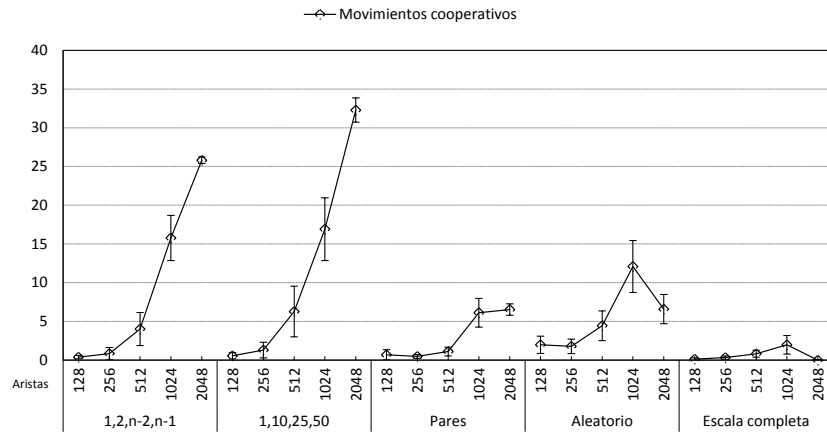
Por la Figura 24 se observa que existen algunas distribuciones y densidades donde el algoritmo M-CBAPM realiza un cantidad de movimientos en promedio cercana a cero. Esto se presenta en las distribuciones pares, aleatorio y escala completa, y en especial en aquellos grafos de baja conectividad (128 o 256 vértices), o bien, y paradójicamente, en grafos completos.

5.3.4. Algoritmo M-CBAPM-BSW y algoritmo DM-CBAPM

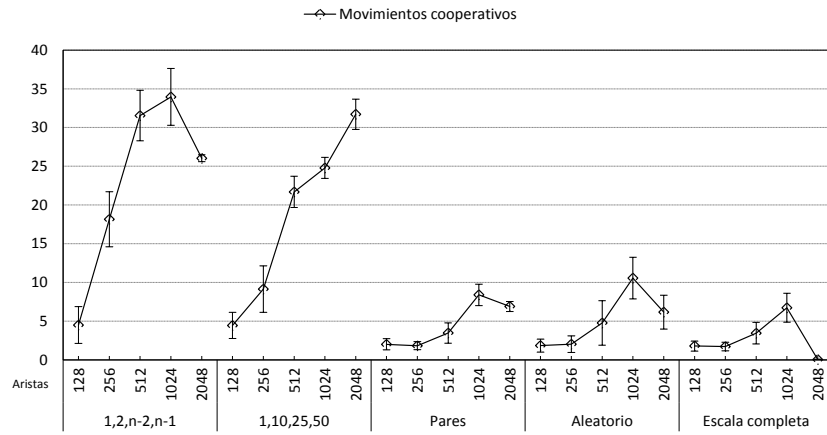
En esta sección se comparan los PoS y PoA obtenidos por los algoritmos CBAPM mixto con mejor bienestar social (M-CBAPM-BSW) y CBAPM mixto doble (DM-CBAPM), con respecto a los obtenidos por el algoritmo M-CBAPM, para las diferentes distribuciones, densidades y modelos de grafos.

5.3.4.1. PoS aproximado

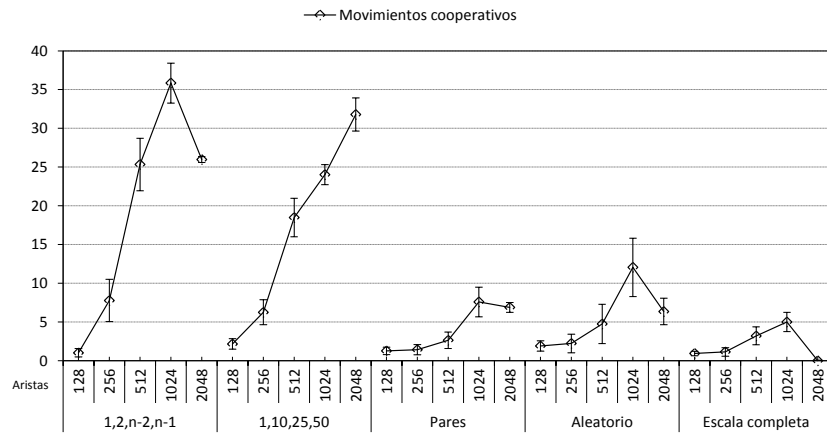
En el caso del algoritmo M-CBAPM-BSW, no existe diferencia significativa alguna con respecto a los resultados obtenidos en el valor del PoS con el M-CBAPM.



(a) Barabási-Albert



(b) Erdős-Rényi



(c) Evolutivo estocástico

Figura 24: Cantidad promedio de movimientos realizados por el algoritmo M-CBAPM en la etapa cooperativa. (a) Cantidad de movimientos para grafos Barabási-Albert. (b) Cantidad de movimientos para grafos Erdős-Rényi. (c) Cantidad de movimientos para grafos generados mediante el modelo evolutivo estocástico.

Tabla 9: Resultados PoS del algoritmo DM-CBAPM en comparación con el algoritmo M-CBAPM. Para cada densidad y tipo de grafo se indica cuál algoritmo es mejor (Mix = M-CBAPM, Doble = DM-CBAPM, n/a = no existe diferencia significativa).

Barabási-Albert	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	n/a	n/a	n/a	n/a	n/a
	256	n/a	n/a	n/a	n/a	n/a
	512	n/a	n/a	n/a	n/a	n/a
	1024	n/a	Doble	n/a	n/a	n/a
	2048	n/a	Doble	Doble	Doble	n/a
Erdős-Rényi	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	n/a	n/a	n/a	n/a	n/a
	256	n/a	n/a	n/a	n/a	n/a
	512	n/a	Doble	n/a	n/a	n/a
	1024	n/a	Doble	Doble	n/a	n/a
	2048	n/a	Doble	Doble	n/a	n/a
Evolutivo estocástico	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	n/a	n/a	n/a	n/a	n/a
	256	n/a	n/a	n/a	n/a	Doble
	512	n/a	Doble	n/a	n/a	n/a
	1024	n/a	Doble	Doble	n/a	n/a
	2048	n/a	Doble	Doble	n/a	n/a

El algoritmo M-CBAPM puede llevar a cabo movimientos durante su fase no cooperativa que decrementen el valor del bienestar social en vez de incrementarlo. El algoritmo M-CBAPM-BSW también lleva a cabo estos movimientos; sin embargo, comienza la parte cooperativa a partir de la configuración que obtuvo el mayor bienestar social durante la primera etapa, deshaciendo los posibles movimientos que hayan decrementado el bienestar social a partir de esta configuración. Sin embargo, estas pérdidas en el valor del bienestar social se pueden recuperar durante la etapa cooperativa del algoritmo M-CBAPM, de modo que ambos algoritmos son fundamentalmente iguales. En cualquier caso, el algoritmo M-CBAPM-BSW ahorra alguna cantidad de movimientos a la etapa cooperativa.

En la Tabla 9 se muestran los resultados de la comparación de los algoritmos M-CBAPM y DM-CBAPM. En ningún caso el DM-CBAPM obtiene PoS promedio menores a los del M-CBAPM. Por los resultados obtenidos se observa que el DM-CBAPM obtiene un mayor valor de PoS para las distribuciones (1, 10, 25, 50) y pares, en los casos de grafos densos.

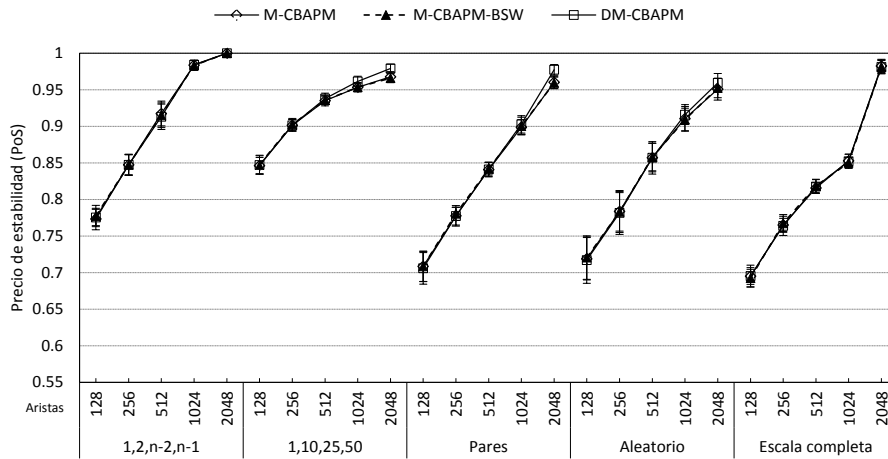
Tabla 10: Resultados PoA del algoritmo DM-CBAPM en comparación con el algoritmo M-CBAPM. Para cada densidad y tipo de grafo se indica cuál algoritmo es mejor (Mix = M-CBAPM, Doble = DM-CBAPM, n/a = no existe diferencia significativa).

Barabási-Albert	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	n/a	n/a	n/a	n/a	n/a
	256	n/a	n/a	n/a	n/a	n/a
	512	n/a	n/a	n/a	n/a	n/a
	1024	n/a	n/a	n/a	n/a	n/a
	2048	n/a	Doble	Doble	n/a	n/a
Erdős-Rényi	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	n/a	n/a	n/a	n/a	n/a
	256	n/a	n/a	n/a	n/a	n/a
	512	n/a	Doble	n/a	n/a	n/a
	1024	n/a	Doble	n/a	n/a	n/a
	2048	n/a	Doble	Doble	n/a	n/a
Evolutivo estocástico	Densidad	1, 2, $n - 2$, $n - 1$	1, 10, 25, 50	Pares	Aleatorio	Escala completa
	128	n/a	n/a	n/a	n/a	n/a
	256	n/a	n/a	n/a	n/a	n/a
	512	n/a	Doble	n/a	n/a	n/a
	1024	n/a	Doble	n/a	n/a	n/a
	2048	n/a	Doble	Doble	n/a	n/a

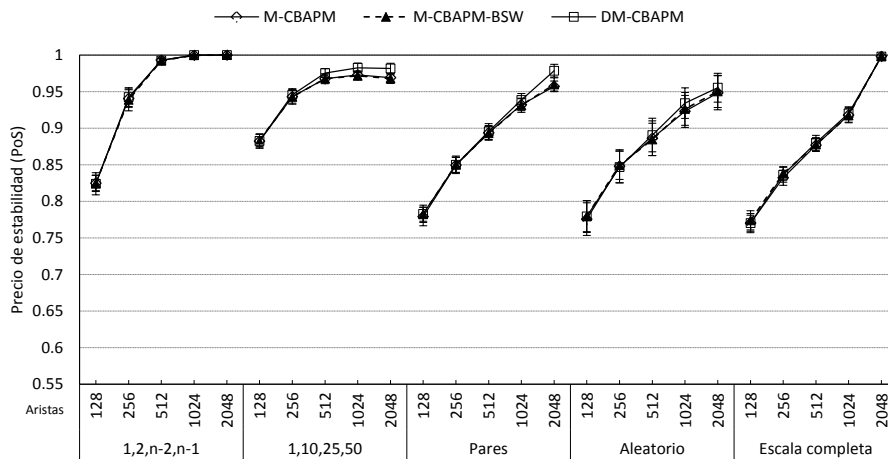
En la Figura 25 se muestran los valores de los PoS promedio obtenidos por los algoritmos M-CBAPM, M-CBAPM-BSW y DM-CBAPM. Se puede observar que los resultados de los tres algoritmos son muy cercanos, o incluso idénticos, aunque sí existen casos donde el DM-CBAPM obtiene valores ligeramente mayores a los de los otros dos algoritmos.

5.3.4.2. PoA aproximado

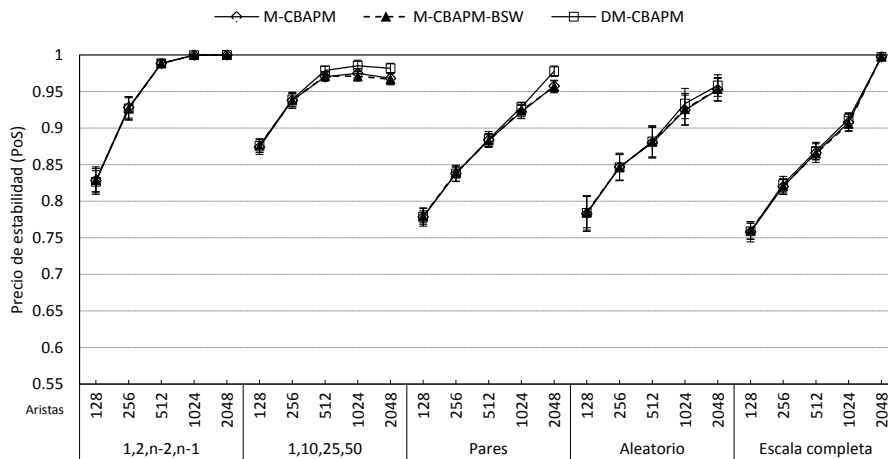
En la Figura 26 se muestran los valores de los PoA promedios obtenidos por los algoritmos M-CBAPM, M-CBAPM-BSW y DM-CBAPM. Todos los algoritmos obtienen valores de PoA muy cercanos entre sí para la mayoría de los casos, salvo en dos casos. El más notable se trata del algoritmo M-CBAPM-BSW para grafos densos de la distribución (1, 2, $n - 2$, $n - 1$), que obtiene un valor de PoA por debajo del obtenido por los otros dos algoritmos. El otro caso se presentan en la distribución (1, 10, 25, 50), donde en los grafos de 512 aristas o más, el algoritmo DM-CBAPM obtiene valores de PoA ligeramente mayores a los de los otros dos algoritmos.



(a) Barabási-Albert

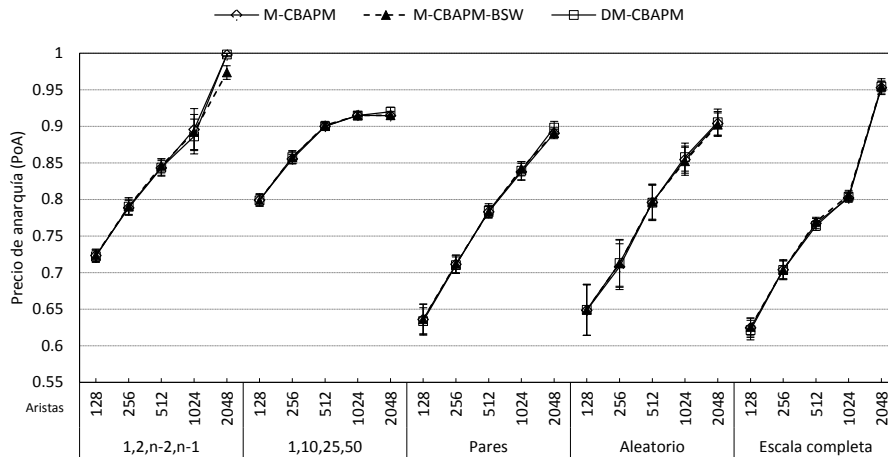


(b) Erdős-Rényi

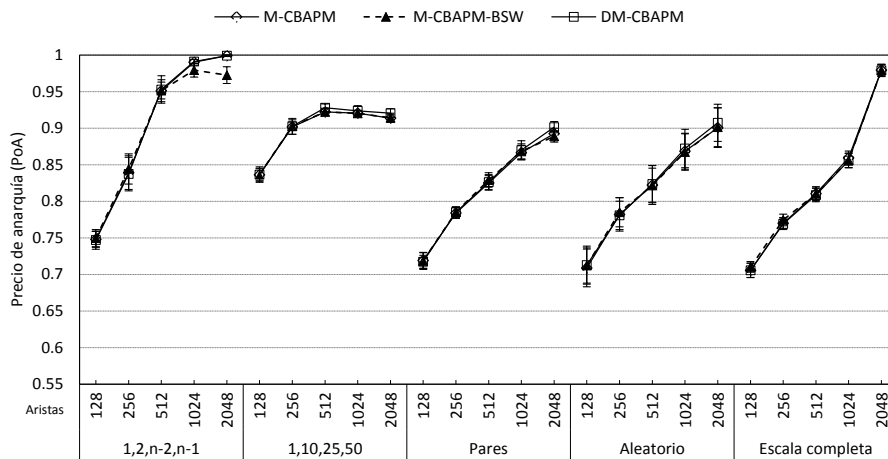


(c) Evolutivo estocástico

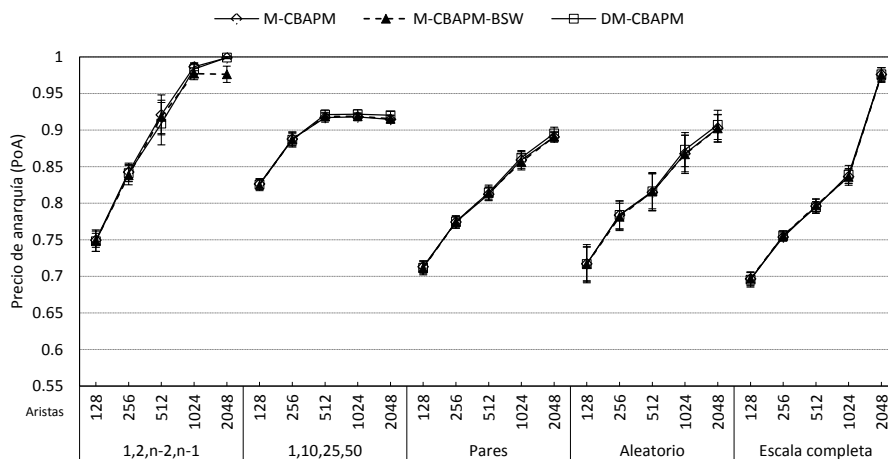
Figura 25: Valores experimentales del precio de estabilidad de los algoritmos M-CBAPM, M-CBAPM-BSW y DM-CBAPM de acuerdo a diferentes distribuciones de clases. (a) El PoS para grafos Barabási-Albert. (b) El PoS para grafos Erdős-Rényi. (c) El PoS para grafos generados mediante el modelo evolutivo estocástico.



(a) Barabási-Albert



(b) Erdős-Rényi



(c) Evolutivo estocástico

Figura 26: Valores experimentales del precio de anarquía de los algoritmos M-CBAPM, M-CBAPM-BSW y DM-CBAPM de acuerdo a diferentes distribuciones de clases. (a) El PoA para grafos Barabási-Albert. (b) El PoA para grafos Erdős-Rényi. (c) El PoA para grafos generados mediante el modelo evolutivo estocástico.

En la Tabla 10 se muestran los resultados de las comparaciones estadísticas entre el M-CBAPM y el DM-CBAPM. Los resultados corroboran lo que se puede apreciar en la Figura 26, que sí existe una diferencia significativa a favor del algoritmo DM-CBAPM para los grafos de la distribución (1, 10, 25, 50).

Capítulo 6. Conclusiones y trabajo a futuro

6.1. Resumen

El Problema de Asignación de Guardaespaldas Multiclase (PAGM) es un modelo matemático donde existen entidades racionales con objetivos, en mayor o menor medida, opuestos. El objetivo global del juego es formar una estructura, denominada árbol de esparcimiento, que maximice el bienestar social de todas las entidades, mientras que el objetivo particular de cada entidad es maximizar su utilidad individual. El bienestar social representa la suma de las utilidades individuales de todas las entidades involucradas. La solución al PAGM es un árbol de esparcimiento tal que todos los vértices se encuentran en un estado estable, es decir, ningún vértice puede incrementar su utilidad individual unilateralmente. El PAGM se basa en el Problema de Asignación de Guardaespaldas (PAG), propuesto en Fajardo-Delgado *et al.* (2013), donde modelan el PAG como un juego y lo estudian bajo dos enfoques: cooperativo y no cooperativo; también proponen el algoritmo CBAP para encontrar una solución al PAG en tiempo polinomial; sucesivos autores mejoraron el rendimiento del algoritmo CBAP (Brubeck Salcedo, 2011), así como también propusieron soluciones al PAG mediante algoritmos genéticos (Zatarain Aceves, 2011).

El presente trabajo extiende el problema propuesto por Fajardo-Delgado *et al.* (2013), y propone uno más general, que mantiene compatibilidad con el PAG.

Se proponen dos algoritmos basados en el CBAP original, el CBAPM y el M-CBAPM. El CBAPM implementa esquemas cooperativo y no cooperativo para la resolución del PAGM. Se demuestra la convergencia al equilibrio del CBAPM bajo el enfoque cooperativo; por otro lado, se encontraron casos para los cuales el CBAPM no converge al equilibrio bajo el enfoque no cooperativo. El M-CBAPM propone una combinación de ambos esquemas, con el objetivo de mejorar la calidad de las soluciones que se pueden obtener mediante el uso de ambos esquemas de manera aislada. Adicionalmente, también se implementaron las modificaciones necesarias al algoritmo DBAP, originalmente propuesto por Fajardo-Delgado *et al.* (2013) como una versión distribuida del CBAP, para que pueda soportar casos del PAGM. Este nuevo algoritmo, el DBAPM, puede funcionar bajo los esquemas cooperativo y no cooperativo.

Para llevar a cabo pruebas experimentales que ayudaran a determinar el rendimiento de los algoritmos propuestos, se diseñaron diferentes conjuntos de casos específicos del PAGM, basándose en los mismos conjuntos propuestos por Zatarain Aceves (2011), pero ampliándolos de manera que aprovecharan las capacidades que ofrece el problema extendido del PAGM. Se llevaron a cabo ejecuciones de los algoritmos propuestos sobre estos conjuntos de casos del PAGM para comparar directamente los enfoques cooperativo y no cooperativo, así como también, estos dos enfoques contra el enfoque del algoritmo M-CBAPM.

6.2. Conclusiones y conjeturas

Una de las conclusiones que más contrasta con respecto a las del trabajo de Fajardo-Delgado *et al.* (2013), concierne a la eficiencia de los enfoques cooperativo y no cooperativo. Mientras que en el estudio del PAG se concluye que el enfoque cooperativo es, en general, mejor que el no cooperativo (tanto en PoS como en PoA), los experimentos llevados a cabo en el problema extendido nos proporcionan indicios de que éste no siempre es el caso, y que en realidad, dependiendo de las clases de vértices que estén presentes en un caso específico, es que un enfoque puede desempeñarse mejor que otro.

Se puede concluir que los casos específicos del PAGM pueden catalogarse en tres grandes grupos: distribuciones heterogéneas, homogéneas e intermedias.

1. Las distribuciones heterogéneas son aquellas en las que los vértices suelen tener una preferencia de distancia a la raíz distinta de las de los demás vértices; i.e., existen pocos vértices en el grafo que comparten la misma función de preferencia de distancia a la raíz. Ejemplos de estas clases de distribuciones son las que se denominaron de escala completa, pares, y aleatoria.
2. Las distribuciones homogéneas son aquellas en las que los vértices suelen tener una preferencia de distancia a la raíz igual, o al menos similar, a las de muchos otros vértices; i.e., los vértices en el grafo se pueden agrupar por su función de preferencia en grandes grupos, y éstos además pueden ser cercanos entre sí. Los casos específicos que maneja el PAG entran en esta categoría (donde los vértices

se separan en dos grandes grupos, blancos y negros). La distribución $(1, 2, n - 2, n - 1)$ también es una distribución homogénea.

3. Las distribuciones intermedias son aquellas en las que los vértices del grafo se pueden separar en pocos grupos, pero estos grupos son muy distintos entre sí. La distribución de cuatro grupos de vértices, $(1, 10, 25, 50)$, pertenece a este tipo de distribución, ya que los grupos están casi equidistribuidos en el codominio de la función de preferencia.

La evidencia experimental sugiere, que para los casos de distribuciones heterogéneas, el enfoque no cooperativo es mejor, tanto en PoS como en PoA, que el enfoque cooperativo. De manera análoga, para casos de distribuciones homogéneas, el enfoque cooperativo obtiene mejores PoS y PoA que el enfoque no cooperativo.

El análisis de los resultados del algoritmo M-CBAPM incita a primera vista a pensar que se trata de un enfoque superior tanto al cooperativo, como al no cooperativo; sin embargo en una revisión más cautelosa existen dos puntos importantes que hay que considerar:

- Es natural que el algoritmo M-CBAPM nunca obtenga un rendimiento inferior al de los otros dos enfoques, pues es una combinación de ambos enfoques, y por la manera como está estructurado (primero la etapa no cooperativa, seguida de la etapa cooperativa), obtiene las ventajas de ambos enfoques.
- Si bien el algoritmo M-CBAPM no es inferior en PoS y PoA al enfoque no cooperativo en distribuciones heterogéneas, tampoco es significativamente mejor. Lo que hace que su uso no sea estrictamente necesario para obtener los mejores resultados en estas distribuciones.

El algoritmo M-CBAPM permite ver a los dos enfoques desde el punto de vista de heurísticas de optimización, en donde el enfoque no cooperativo tiene la función de búsqueda de *exploración*, esto es, se busca la solución óptima sin importar pérdidas

temporales en el valor a optimizar, y el enfoque cooperativo como la búsqueda de *exploración*, quiere decir que a partir de un punto se optimiza lo más posible la función objetivo, a riesgo de caer en óptimos locales.

Se puede concluir que el M-CBAPM es ideal para usar en distribuciones de tipo homogéneas e intermedias, que son aquellas donde el enfoque no cooperativo no puede incrementar más el bienestar social después de cierto punto, y el cambio alternado al enfoque cooperativo permite encontrar mejores soluciones.

Una de las preguntas más importantes que aún quedan por resonar es, ¿exactamente bajo qué condiciones el PAGM puede o no converger al equilibrio?. En el presente trabajo se demostró que mediante el algoritmo propuesto, el CBAPM, bajo el enfoque cooperativo, siempre es posible alcanzar una condición de equilibrio. Bajo el esquema no cooperativo, éste no siempre es el caso, ya que existen casos específicos para los cuales el CBAPM queda atrapado en ciclos sin salida. Sin embargo, el presente trabajo también contribuye con evidencias tanto teóricas como experimentales de que existen factores que pueden delimitar la caída del CBAPM en esta clase de ciclos. Desde el punto de vista matemático riguroso se demostró que para casos donde los grafos son completos, siempre se converge al equilibrio, bajo cualquiera de los dos esquemas. Aunado a este resultado, está la investigación original de Fajardo-Delgado *et al.* (2013), donde se demuestra que para casos que maneja el PAG, el CBAP siempre converge al equilibrio, bajo cualquiera de los dos esquemas, y sin importar el grado de interconexión entre los vértices. Por el lado experimental, se confirma que de las 67500 ejecuciones del CBAPM bajo el enfoque no cooperativo (mediante el calendarizador uniforme) de los diferentes casos específicos propuestos, en ningún caso, el algoritmo quedó atrapado en un ciclo. Por lo tanto, la evidencia tanto experimental como teórica sugiere que las características necesarias para que un caso del PAGM no converja al equilibrio radica en una delicada combinación y balance entre, escaso nivel de interconexión entre los vértices, y alto grado de heterogeneidad en las funciones de preferencias de los vértices, por lo que el grado de atracción de estos ciclos, i.e., la probabilidad de que el CBAPM caiga en uno de estos ciclos, es mínima.

6.3. Trabajo a futuro

A partir de los resultados obtenidos en este trabajo, se presentan algunas ideas como propuestas para trabajo a futuro:

- Uno de los principales problemas con el enfoque cooperativo, radica en su bajo número de movimientos realizados, por lo que se sugiere investigar nuevos enfoques cooperativos o bien, mejoras al enfoque en su forma actual, de modo que le permitan llevar a cabo una mayor cantidad de movimientos, con la esperanza de que eso incremente la calidad de sus soluciones encontradas.
- Análogamente, un comportamiento que presenta el enfoque no cooperativo con grafos densos, es una muy elevada cantidad de movimientos, de los cuales muchos se sospecha son redundantes, por lo que se sugiere investigar la manera de reducir la cantidad de movimientos que realiza el enfoque no cooperativo, para hacerlo más eficiente con respecto al enfoque cooperativo.
- Una manera que se conjetura podría llevar a lograr lo remarcado en el punto anterior, es mediante la experimentación de nuevas políticas de selección en la función CALENDARIZADOR(). El algoritmo DBAP (que es básicamente una política ordenada de selección) obtiene PoS y PoA cercanos a los del algoritmo CBAPM, y sin incurrir en una cantidad de movimientos tan alta como el CBAPM.
- Un fenómeno que no es fácil de notar es que, tanto en el PAG como en el PAGM, bajo el enfoque no cooperativo, emergen de manera natural ciertos comportamientos cooperativos, aún cuando supuestamente se trata de un enfoque completamente egoísta. Esto sucede cuando los vértices que desean estar en las posiciones más alejadas de la raíz no encuentran dichas posiciones, pero aceptan la más alejada posible, a pesar de que más adelante pueden servir de escalera para otros vértices que también desean alejarse de la raíz, y de esta manera, llevando a cabo un sacrificio. Se propone experimentar con un tipo de vértice más inteligente, con mayor cantidad de información, y aún más egoísta, que se tome la libertad de sacrificar turnos, bajo la expectativa de que sean otros vértices de su mismo tipo los que tomen

posiciones de sacrificio. Se considera interesante ver si con este tipo de vértices aún es posible converger a un equilibrio, y de que manera afecta al bienestar social.

- Análogamente al punto anterior, también se propone aprovechar estas características cooperativas emergentes en enfoques no cooperativos de modo que ayuden a acelerarlos. Se propone utilizar el algoritmo CBAPM bajo enfoque no cooperativo para resolver casos del PAG. La idea radica en que cada vértice, blanco o negro, toma temporalmente una preferencia de distancia aleatoria dentro de un intervalo correspondiente a su color, de modo que el caso se resuelve como si se tratara de la distribución aleatoria (ver Sección 5.1.1). Se especula que el resultado serían soluciones de calidad cercanas o estadísticamente iguales, a los que obtiene el enfoque cooperativo, para ciertos tipos de grafos.
- Se propone estudiar los casos específicos del PAGM, cambiando sistemáticamente la estructura de los vértices y aristas para encontrar condiciones que generan ciclos que no permiten converger al equilibrio.
- Finalmente, se propone extender aún más el problema del PAGM. En el trabajo presente se extendió el problema original propuesto por Fajardo-Delgado *et al.* (2013) al agregar nuevas clases de vértices. Se propone ahora extender aún más el problema para otorgar pesos a las aristas que conectan a los vértices. En una red de computadoras, económica, o ecológica, todos los elementos suelen estar interconectados entre ellos, pero no todas las conexiones tienen la misma fuerza o velocidad.

Lista de referencias

- Anshelevich, E., DasGupta, A., Kleinberg, J., Tardos, E., Wexler, T., y Roughgarden, T. (2004). The price of stability for network design with fair cost allocation. En: *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, Oct. pp. 295–304.
- Barabási, A.-L. y Albert, R. (1999). Emergence of scaling in random networks. *Science*, **286**(5439): 509–512.
- Brubeck Salcedo, D. (2011). *Uso de la teoría de juegos para modelar el problema de asignación de guardaespaldas*. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, Baja California, México.
- Callaway, D. S., Hopcroft, J. E., Kleinberg, J. M., Newman, M. E., y Strogatz, S. H. (2001). Are randomly grown graphs really random? *Physical Review E*, **64**(4): 041902.
- Charilas, D. E. y Panagopoulos, A. D. (2010). A survey on game theory applications in wireless networks. *Computer Networks*, **54**(18): 3421–3430.
- Cohen, J., Dasgupta, A., Ghosh, S., y Tixeuil, S. (2008). An exercise in selfish stabilization. *ACM Trans. Auton. Adapt. Syst.*, **3**(4): 15:1–15:12.
- Cormen, T., Leiserson, C., Rivest, R., y C. Stein (2009). *Introduction to Algorithms*. The MIT Press, tercera edición.
- Csardi, G. y Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*: 1695.
- Dasgupta, A., Ghosh, S., y Tixeuil, S. (2006). Selfish stabilization. En: *Stabilization, Safety, and Security of Distributed Systems*. Springer, pp. 231–243.
- Datta, A. K., Johnen, C., Petit, F., y Villain, V. (2000). Self-stabilizing depth-first token circulation in arbitrary rooted networks. *Distributed Computing*, **13**(4): 207–218.
- Erdos, P. y Rényi, A. (1960). On the evolution of random graphs. En: *Publication of the Mathematical Institute of Hungarian Academy of Sciences*. pp. 17–61.
- Fajardo-Delgado, D., Fernandez-Zepeda, J. A., y Bourgeois, A. G. (2013). The bodyguard allocation problem. *IEEE Trans. Parallel Distrib. Syst.*, **24**(7): 1465–1478.
- Feldman, M., Lai, K., Stoica, I., y Chuang, J. (2004). Robust incentive techniques for peer-to-peer networks. En: *Proceedings of the 5th ACM conference on Electronic commerce*. ACM, pp. 102–111.
- Griffin, T. G., Shepherd, F. B., y Wilfong, G. (2002). The stable paths problem and inter-domain routing. *IEEE/ACM Transactions on Networking (ToN)*, **10**(2): 232–243.
- Grosu, D. y Chronopoulos, A. T. (2004). Algorithmic mechanism design for load balancing in distributed systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **34**(1): 77–84.

- Halpern, J. Y. (2007). Computer science and game theory: A brief survey. *arXiv preprint cs/0703148*.
- Koutsoupias, E. y Papadimitriou, C. (2009). Worst-case equilibria. *Computer science review*, **3**(2): 65–69.
- Lilliefors, H. W. (1967). On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, **62**(318): 399–402.
- Machado, R. y Tekinay, S. (2008). A survey of game-theoretic approaches in wireless sensor networks. *Computer Networks*, **52**(16): 3047–3061.
- Moscibroda, T., Schmid, S., y Wattenhofer, R. (2006). On the topologies formed by selfish peers. En: *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, New York, NY, USA. ACM, PODC '06, pp. 133–142.
- Nash, J. F. *et al.* (1950). Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, **36**(1): 48–49.
- Nisan, N., Roughgarden, T., Tardos, E., y Vazirani, V. V. (2007). *Algorithmic Game Theory*. Cambridge University Press. New York, NY, USA. 776 pp.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Raghavan, P. y Motwani, R. (1995). *Randomized Algorithms*. Cambridge Univ. Press Cambridge. 476 pp.
- Raidl, G. R. y Julstrom, B. A. (2003). Edge sets: an effective evolutionary coding of spanning trees. *Evolutionary Computation, IEEE Transactions on*, **7**(3): 225–239.
- Roughgarden, T. (2005). *Selfish routing and the price of anarchy*. MIT press Cambridge. 240 pp.
- Schulz, A. S. y Moses, N. S. (2003). On the performance of user equilibria in traffic networks. En: *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, pp. 86–87.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, pp. 80–83.
- Zatarain Aceves, H. (2011). *Resolviendo el problema de asignación de guardaespaldas utilizando algoritmos genéticos*. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, Baja California, México.
- Zeev Collin, S. D. (1994). Self-stabilizing depth first search. *Information Processing Letters*, **49**: 297–301.