



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Heurísticas y Metaheurísticas

CIDM

Algoritmos y Estructuras de Datos III
Primer Cuatrimestre de 2015

Integrante	LU	Correo electrónico
Baraňao, Facundo	480/11	facundo_732@hotmail.com
Confalonieri, Gisela Belén	511/11	gise_5291@yahoo.com.ar
Mignanelli, Alejandro Rubén	609/11	minga_titere@hotmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Introducción	3
2. Relación con trabajos previos y ejemplos de Uso	3
3. Plataforma de pruebas	4
4. Propiedad	5
4.1. 1) Todo conjunto independiente maximal es dominante	5
4.2. 2) Todo conjunto independiente dominante es independiente maximal	5
5. Algoritmo Exacto	5
5.1. Desarrollo de la idea y correctitud.	5
5.2. Análisis de complejidad.	5
5.3. Experimentación y gráficos.	5
5.3.1. Test 1	5
5.3.2. Test 2	5
6. Heurística Golosa Constructiva	6
6.1. Desarrollo de la idea y correctitud.	6
6.2. Análisis de complejidad.	6
6.3. Instancias no óptimas.	6
6.4. Experimentación y gráficos.	6
6.4.1. Test 1	6
6.4.2. Test 2	6
7. Heurística de Búsqueda Local	7
7.1. Desarrollo de la idea y correctitud.	7
7.2. Análisis de complejidad de una iteración.	7
7.3. Experimentación y gráficos.	7
7.3.1. Test 1	7
7.3.2. Test 2	7
8. Metaheurística de GRASP	8
8.1. Desarrollo de la idea y correctitud.	8
8.2. Experimentación y gráficos.	8
8.2.1. Test 1	8
8.2.2. Test 2	8
9. Comparación de los distintos Métodos	9
9.1. Experimentación y gráficos.	9
9.1.1. Test 1	9
9.1.2. Test 2	9
10.Apéndice 1: acerca de los tests	10
10.1.Código del Problema 1	10
10.2.Código del Problema 2	10
10.3.Código del Problema 3	10

1. Introducción

El problema de Conjunto Independiente Dominante Mínimo (CIDM) consiste en hallar un conjunto independiente dominante de G con mínima cardinalidad.

Sea $G = (V, E)$ un grafo simple. Un conjunto $D \subseteq V$ es un conjunto dominante de G si todo vértice de G está en D o bien tiene al menos un vecino que está en D . Por otro lado, un conjunto $I \subseteq V$ es un conjunto independiente de G si no existe ningún eje de E entre dos vértices de I . Definimos entonces un conjunto independiente dominante de G como un conjunto independiente que a su vez es un conjunto dominante del grafo G .

Un conjunto independiente de $I \subseteq V$ se dice maximal si no existe otro conjunto independiente $J \subseteq V$ tal que $I \subset J$, es decir que I está incluido estrictamente en J . En otra sección de este informe se demostrará que todo conjunto independiente dominante, es particularmente independiente maximal.

En el presente trabajo, se pretende analizar y comparar diferentes maneras de encarar este problema. En particular se utilizará un algoritmo exacto, una heurística golosa, una heurística de búsqueda local, y la metahurística de GRASP.

2. Relación con trabajos previos y ejemplos de Uso

En el TP1 de la materia, hemos encontrado y analizado un algoritmo que resolvía un problema que fue llamado El Señor de los Caballos. El problema era el siguiente:

Se tiene un juego de mesa cuyo tablero, dividido en casillas, posee igual cantidad de filas y columnas y hace uso de una conocida pieza del popular ajedrez: el caballo. El juego es solamente para un jugador y consiste en, teniendo caballos ubicados en distintos casilleros, insertar en casilleros vacíos la mínima cantidad de caballos extras, de manera tal que, siguiendo las reglas del movimiento de los caballos en el ajedrez, todas las casillas se encuentren ocupadas o amenazadas por un caballo. Aspectos a tener en cuenta:

- Se conoce la cantidad de filas y columnas del tablero
- Se conoce la cantidad de caballos que ocupan el tablero inicialmente.
- Para cada uno de estos caballos, se sabe su ubicación en el tablero
- Una casilla se considera amenazada si existe un caballo tal que en una movida pueda ocupar dicha casilla

Observemos que si consideramos a cada casilla como un nodo, y que las aristas hacia otros nodos se dan por las casillas a las cuales puede llegar un caballo ubicado en ese lugar, entonces el señor de los caballos puede ser visto como la búsqueda de un conjunto dominante minimal que tenga a los nodos/-casillas que tienen un caballo preubicado. Sin embargo, no podemos traducir este problema a un CIDM, puesto que no necesariamente debe cumplir el requisito de independencia. O sea, no necesariamente si la solución tiene un caballos en una casilla determinada, no haya ningún caballo ubicado en alguna de las casillas que este caballo amenaza. Veamos un ejemplo:

(desde ver... hasta 4 4 habría que poner una imagen de un tablero de 5x5 vacío(1), y de esta solución(2) y de la solución que respeta CIDM(3), que es la que tiene 5 caballos en el centro.) Ver que la solución de nuestro programa para un tablero vacío de 5x5 devuelve lo siguiente: 3 2 3 3 3 4 4 3 4 4

Si nuestro tablero fuera el de la figura 1, una solución posible(de hecho, la encontrada con el algoritmo usado en el TP1) es la figura 2. Esta solución podemos observar que es dominante, pero no independiente puesto que el caballo de la fila 3 columna 2 amenaza al de la fila 4 columna 4. Sin embargo en este caso, existe una solución CIDM como lo es la figura 3.

(Dejo la parte de trabajos previos hasta aca, puesto que hay un ejemplo en el txt que quiero probar si funciona y agregaría un par de líneas)

Como por lo visto hasta ahora CIDM no se adapta del todo al problema de los caballos(asumo eso por ahora, desp de última se cambia(ALE)) cabe preguntarse, en que situaciones de la vida real podría aplicarse este problema? Veamos algunos ejemplos realistas y alguno no tanto.

1. Una empresa de bebidas gaseosas tiene sus actividades divididas en áreas, las cuales se interrelacionan de cierta manera (no necesariamente todas con todas). Esta empresa está por sacar una nueva bebida sabor cola y quiere hacerlo antes que su rival, quien está más adelantado en la producción. Para lograr sacar el producto antes que su contrincante, nuestra empresa debe lograr una mejora en el trabajo de sus diferentes áreas. Por cómo está estructurada la empresa, si un área tiene una mejora considerable en tiempo, todas las áreas que se relacionan con ella se verán beneficiadas, pero este beneficio ya no impacta en las áreas relacionadas con estas últimas. Queremos entonces lograr impulsar mejoras considerables en las áreas que sean necesarias para que toda la empresa funcione más rápido y logre sacar su producto a tiempo. Para ello se contratarán especialistas con sueldos sumamente importantes, por lo cual se desea que la cantidad de áreas a mejorar considerablemente sea mínima (para minimizar la inversión en especialistas). Además, sabemos que estos especialistas son excelentes, pero muy soberbios/tercos, y ponerlos a cargo de áreas relacionadas puede devenir en discusiones que retrasarían a la empresa, por lo cual procuraremos colocarlos en áreas "no adyacentes".
2. Se tiene un caso de asesinato, y debemos resolverlo. Nuestras investigaciones previas, nos han dado información de todas las personas que tuvieron algo que ver con el incidente, y sabemos cuales se conocen entre si y cuales no, además de que sabemos que toda persona que conoce a otra, sabe lo que hizo dicha persona el día del incidente. Debido a que somos detectives principiantes, el alquiler por día de nuestra oficina nos sale caro, y entrevistar a un testigo, independientemente del volumen de información obtenida, nos toma un día. Si para resolver el caso debieramos escuchar información sobre todas las personas involucradas. A quienes deberíamos llamar de testigos, de manera tal que reduzcamos el alquiler de oficina al mínimo? (Estos testigos no pueden conocerse entre si, para evitar complot y falsos datos.)
3. Milo de escorpio, guardian de la casa de escorpio, ha recibido muchas quejas de parte del gran patriarca, debido a que por su casa pasa todo el mundo. Cansado de ser tantas veces derrotados, nos pide ayuda, y para esto nos explica la verdad sobre su gran técnica, la aguja escarlata. Lejos de lo que se cuenta en el animé Los caballeros del zodiaco, el verdadero poder de la aguja escarlata es el siguiente:

Milo nos explica, que el cuerpo de cada ser humano, puede ser dividido en trozos, pero no trozos como el brazo, o la pierna, sino a porciones mas pequeñas de cuerpo. Estos trozos, pueden tener una correspondencia con muchos otros trozos de cuerpo, y estas correspondencias no son necesariamente a nivel local (por ejemplo, una porcion del dedo indice de un humano, puede estar conectada a una porción de la cabeza). Estos trozos pueden ser fácilmente vistos como un grafo. Cuando la aguja escarlata toca un trozo de cuerpo, este y todos aquellos trozos que tengan una correspondencia, se inmovilizan. Pero esto sucede solo si la aguja escarlata impacta contra un trozo sano de cuerpo. Si el trozo en cuestión ya estuviese inmovilizado, entonces la aplicación de la aguja escarlata no resulta en ningun efecto.

Dado que el uso de una aguja escarlata, resulta en un fuerte uso del cosmos de milo, milo quiere que dado un oponente, y la información sobre todos sus trozos y correspondencias, nosotros le fabriquemos un algoritmo tal que le diga en que sectores del cuerpo el debe usar la aguja escarlata, de manera tal que deba utilizar la mínima cantidad de agujas escarlata posibles. Milo nos asegura que el tiempo de dicho algoritmo no tiene importancia, dado que le pidio prestada la habitación del tiempo a kamisama, que esta equipada con una notebook y un enchufe para dejar la batería cargada (lo que si, no tiene wifi dado a una muy mala señal, por lo que debemos darle un pendrive con el código).

3. Plataforma de pruebas

Para toda la experimentación se utilizará un procesador Intel Core i3, de 4 nucleos a 2.20 GHZ. El software utilizado será Ubuntu 14.04, y G++ 4.8.2.

4. Propiedades

En esta sección demostraremos ciertas propiedades que serán usadas a lo largo del trabajo.

4.1. Todo conjunto independiente maximal es dominante

Supongamos que no. O sea, que existe algun grafo (V, E) tal que su conjunto independiente maximal (CIM) no es un conjunto dominante. Si esto es así, entonces por no ser dominante, existe un nodo n que pertenece al grafo pero no al CIM, y que además dentro del grafo original, n no es vecino de ningún elemento de CIM. Pero entonces, si añadimos n a CIM se obtiene un conjunto independiente que tiene los mismos elementos de CIM más un nodo, o sea que este nuevo conjunto contiene a CIM, lo cual es absurdo, puesto que habíamos dicho que CIM era maximal. El absurdo proviene de suponer que el conjunto no es dominante, por lo tanto, el conjunto debe ser dominante.

4.2. Todo conjunto independiente dominante es independiente maximal

Supongo que no. Entonces, definamos G un grafo tal que tiene un conjunto independiente dominante pero no independiente maximal, que llamaremos I . Como no es independiente maximal, puedo suponer que existe algun conjunto J independiente, tal que $I \subset J$. Entonces, existe v un nodo que pertenece a J pero no a I tal que v no es vecino de ningún elemento de I , lo cual es absurdo, ya que suponer eso es decir que I no era dominante.

5. Algoritmo Exacto

5.1. Desarrollo de la idea y correctitud.

5.2. Análisis de complejidad.

5.3. Experimentación y gráficos.

5.3.1. Test 1

5.3.2. Test 2

6. Heurística Golosa Constructiva

6.1. Desarrollo de la idea y correctitud.

6.2. Análisis de complejidad.

6.3. Instancias no óptimas.

6.4. Experimentación y gráficos.

6.4.1. Test 1

6.4.2. Test 2

7. Heurística de Búsqueda Local

7.1. Desarrollo de la idea y correctitud.

7.2. Análisis de complejidad de una iteración.

7.3. Experimentación y gráficos.

7.3.1. Test 1

7.3.2. Test 2

8. Metaheurística de GRASP

8.1. Desarrollo de la idea y correctitud.

8.2. Experimentación y gráficos.

8.2.1. Test 1

8.2.2. Test 2

9. Comparación de los distintos Métodos

9.1. Experimentación y gráficos.

9.1.1. Test 1

9.1.2. Test 2

10. Apéndice 1: acerca de los tests

10.1. Código del Problema 1

10.2. Código del Problema 2

10.3. Código del Problema 3