

ALGORITMOS Y ESTRUCTURAS DE DATOS III
Trabajo Práctico N°3

Primera entrega: 29 de Junio de 2012, hasta las 18:00 horas.

Segunda entrega: 20 de Julio de 2012, hasta las 18:00 horas.

Ver información general sobre los Trabajos Prácticos en la página de la materia en Internet.

Un conjunto dominante de un grafo $G = (V, E)$ es un conjunto de nodos $V' \subseteq V$ tal que todo nodo de $V - V'$ es adyacente a algún nodo de V' (equivalentemente, cada nodo de V pertenece a V' o es adyacente a algún nodo que pertenece a V'). Dado un grafo G , el problema Mínimo Conjunto Dominante consiste en encontrar un conjunto dominante de G que tenga la menor cantidad posible de nodos.

1. Describir situaciones de la vida real que puedan modelarse utilizando Mínimo Conjunto Dominante.
 2. Desarrollar e implementar un algoritmo exacto para Mínimo Conjunto Dominante.
 3. Desarrollar e implementar una heurística constructiva golosa para Mínimo Conjunto Dominante.
 4. Desarrollar e implementar una heurística de búsqueda local para Mínimo Conjunto Dominante.
 5. Desarrollar e implementar un algoritmo que use la metaheurística GRASP [1] para Mínimo Conjunto Dominante. En cada iteración de GRASP utilizar como primera fase alguna modificación de la heurística dada en 3, y como segunda fase la heurística dada en 4. Prever la utilización de varios tamaños diferentes para la RCL (lista restringida de candidatos) en la primera fase de GRASP, y analizar posibles criterios de parada para el ciclo principal.
 6. Para cada uno de los métodos de los ejercicios 2 a 5:
 - Calcular el orden de la complejidad en el modelo uniforme.
 - Tratar de describir instancias de Mínimo Conjunto Dominante para las cuales el método no proporciona una solución óptima. ¿Qué tan mala puede ser la solución obtenida respecto de la solución óptima?
 - Aplicar el método a varias instancias de Mínimo Conjunto Dominante, respetando los formatos de archivos que se indican más abajo (utilizar como ejemplo los archivos provistos). En el caso de la metaheurística, aplicarla más de una vez modificando los valores de los parámetros que determinan su comportamiento (distintos tamaños de lista, distintos criterios de parada, etcétera). Analizar la calidad de las soluciones obtenidas y el tiempo de ejecución.
 - Ignorar los costos de lectura y escritura de los archivos tanto al calcular complejidad como al medir.
 7. Presentar los resultados obtenidos en el ejercicio anterior mediante gráficos adecuados. Utilizarlos para comparar los distintos métodos entre sí y para elegir los mejores parámetros para la metaheurística.
- Tp3.in:** El archivo contiene un conjunto de instancias a tratar, cada una representada con n líneas, en donde n es la cantidad de nodos del grafo. En la primera línea aparece n . En cada i -ésima línea restante se enumeran, separados por un espacio en blanco, los nodos de las líneas previas (aquellos con subíndice menor que i) que sean adyacentes al i -ésimo nodo.
- Tp3X.out:** Escribir los resultados en un archivo con este nombre, donde X identifica al método utilizado. Los archivos de salida contienen una línea por cada instancia de entrada, en la cual aparecen los nodos del conjunto dominante separados por un un espacio en blanco.

Referencias

- [1] Thomas A. Feo and Mauricio G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, pages 1–27, 1995.