



UNIVERSIDAD SIMÓN BOLÍVAR

CI5651- Diseño de algoritmos I

Prof. Guillermo Palma

Realizado por:

Wilmer Bandres **Carnet:** 10-10055

Juan A. Escalante **Carnet:** 10-10227

GRAPH COLORING

Sartenejas, 9 de febrero de 2015

- Modelo utilizado:

Se implementó un grafo con un modelo basado en listas de adyacencia haciendo uso de vectores de c++, asignándole a cada nodo un vector de nodos adyacentes, que representan los arcos.

- Estructuras de datos y detalles de implementación:

- Se creó una estructura nodo para representar cada uno de los nodos del grafo.
- Se creó una función que realiza la lectura del archivo en formato DIMACS.
- Las estructuras de datos utilizadas en todo el programa son:
 - Vectores de c++ para la representación del grafo y como estructuras auxiliares para obtener rango, color y otras características de los nodos con un acceso rápido.
 - Sets de c++ utilizados para la representación de los $U[X_k]$ utilizados en el pseudocódigo de la corrección del algoritmo de Brelaz.
- La heurística implementada es Dsatur, implementada en un tiempo de $O(n^2)$ como lo indica el estudio previo hecho por Brelaz, donde se usó un vector que guarda información del grado de saturación actual de un nodo para su posterior elección durante el algoritmo.
- El algoritmo implementado para obtener una cota inferior de la solución es un algoritmo que busca un clique en el grafo. El algoritmo escoge los nodos de mayor grado y empieza a buscar en el grafo los nodos de mayor grado no visitados que sean adyacentes a todos los nodos del clique

formado, si esto ocurre el clique aumenta en 1 su tamaño y se sigue buscando por otros nodos que sigan aumentando el clique.

- El algoritmo que busca la coloración óptima del grafo es el algoritmo corregido de Brelaz que es al mismo tiempo una modificación del algoritmo de Brown y que posee dos rutinas:

- La función labeling: encargada de etiquetar nodos del grafo cuando se va a realizar el backtracking del algoritmo.
- La función principal: encargada de colorear el grafo, haciendo uso de los $U[X_k]$ (colores actualmente disponibles para cada nodo X_k) y del respectivo backtracking de ser necesario.

- Problemas encontrados:

- Al ser la coloración de grafos un problema de gran complejidad dentro de las ciencias de la computación, el tiempo de ejecución de este algoritmo con un espacio de búsqueda tan grande y con tan pocas oportunidades de descartar opciones nos pareció un elemento desafiante, el cual enfrentamos con una solución sencilla del estilo de maratones de programación; poco estructurada pero con uso efectivo de la memoria disponible para tener accesos rápidos y con todas las optimizaciones posible que puede hacer el compilador.

CONCLUSIONES

- El algoritmo de Brelaz no es correcto.
- El algoritmo de Brelaz corregido es un algoritmo que no descarta soluciones óptimas.
- El algoritmo de Brelaz corregido representa una mejora sustancial al backtracking de todo el espacio de búsqueda.
- El problema de coloración de grafos es un problema con un árbol de búsqueda extremadamente grande y difícil de podar.
- El problema de coloración de grafos necesita algoritmos muy eficientes, posiblemente con paralelismo para obtener soluciones rápidas.