

ALGORITMOS Y ESTRUCTURAS DE DATOS III

## Trabajo Práctico N°3

De Sousa Bispo Mariano	389/08	marian_sabianaa@hotmail.com
Grosso Daniel	694/08	dgrosso@gmail.com
Livorno Carla	424/08	carlalivorno@hotmail.com
Raffo Diego	423/08	enanodr@hotmail.com

Junio 2010

# Índice

<b>Introducción</b>	<b>2</b>
<b>1. Situaciones de la vida real</b>	<b>2</b>
<b>2. Algoritmo exacto</b>	<b>2</b>
2.1. Optimizacion . . . . .	2
2.2. Complejidad temporal . . . . .	2
<b>3. Heurística constructiva</b>	<b>3</b>
3.1. Complejidad temporal . . . . .	3
<b>4. Búsqueda local</b>	<b>4</b>
4.1. Complejidad temporal . . . . .	4
<b>5. Tabu-Search</b>	<b>5</b>
5.1. Complejidad temporal . . . . .	5
<b>6. Resultados</b>	<b>6</b>
6.1. Parametros de la heurística tabú . . . . .	6
6.2. Comparacion de tiempos . . . . .	6
6.3. Comparacion de calidad . . . . .	6
<b>7. Compilación y ejecución de los programas</b>	<b>6</b>
<b>8. Conclusiones</b>	<b>6</b>

# Introducción

Este trabajo tiene como objetivo la aplicación de diferentes técnicas algorítmicas para la resolución de tres problemas particulares, el cálculo de complejidad teórica en el peor caso de cada algoritmo implementado, y la posterior verificación empírica.

El lenguaje utilizado para implementar los algoritmos de todos los problemas fue C/C++

## 1. Situaciones de la vida real

## 2. Algoritmo exacto

### 2.1. Optimización

Dado que se trata de un algoritmo de backtracking, la optimización se basa en podar las ramas en las que estamos seguros que no va a aparecer el óptimo. Para esto tenemos que poder predecir, dado un estado actual, si es posible mejorar el óptimo encontrado hasta el momento.

Por un lado, podemos podar las ramas que no forman un grafo completo, ya que no es solución.

Por otro lado, evaluamos en cada paso del algoritmo la cantidad de vértices que falta explorar. Es decir, calculamos el tamaño de la clique máxima que podríamos formar considerando los vértices que ya están incluidos en la solución actual. Si la cantidad de vértices que todavía no fueron evaluados más la cantidad de vértices ya pertenecientes a la clique actual es menor a la cantidad de vértices de la clique máxima encontrada hasta el momento, no tiene sentido seguir explorando esa rama ya que el tamaño de la clique máxima que se puede encontrar por ese camino es menor al de la máxima encontrada. Por este motivo, podemos podar esta rama.

### 2.2. Complejidad temporal

### **3. Heurística constructiva**

#### **3.1. Complejidad temporal**

## 4. Búsqueda local

### 4.1. Complejidad temporal

## 5. Tabu-Search

### 5.1. Complejidad temporal

## 6. Resultados

### 6.1. Parametros de la heurística tabú

### 6.2. Comparacion de tiempos

### 6.3. Comparacion de calidad

## 7. Compilación y ejecución de los programas

## 8. Conclusiones