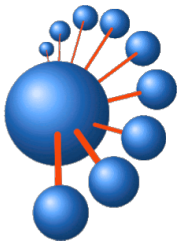

UNIVERSIDAD DE CONCEPCIÓN

Departamento de Ingeniería Informática y

Ciencias de la Computación

Maximum Independent Set Problem

Entrega N°1: Heurísticas Greedy y Greedy Aleatorizado

Profesor/a: Pedro Pinacho Davidson
Asignatura: Sistemas Adaptativos
Estudiantes:

- Dreyko Paredes Mansilla
- Joaquín San Martín Vargas
- Pablo Sanhueza Yévenes

Concepción, 25 de Septiembre de 2025

1. Desarrollo

El problema *Maximum Independent Set* (MIS) es de naturaleza NP-hard, lo que implica que resolver instancias de gran tamaño de forma exacta resulta inviable. Frente a esta dificultad, una alternativa es recurrir a heurísticas Greedy, las cuales permiten construir soluciones válidas de manera incremental y en tiempos significativamente menores. En este trabajo se presentan dos variantes: una determinista y otra aleatorizada.

1.1. Heurística Greedy determinista

La estrategia implementada consiste en seleccionar de manera iterativa el vértice con menor grado dentro del grafo. Este criterio busca reducir la cantidad de conflictos al incorporar el vértice al conjunto independiente, ya que un nodo con menos adyacencias restringe en menor medida las futuras elecciones. Una vez que un vértice es agregado a la solución, tanto él como todos sus vecinos se descartan del proceso de selección, garantizando así que el conjunto resultante se mantenga independiente.

Algorithm 1 Greedy determinista para MIS

```
1: Entrada: Grafo G = (V, E)
2: Ordenar los vértices de V por grado ascendente
3: S ← ∅                                          // Conjunto independiente
4: Marcar todos los vértices como no usados
5: for cada vértice u en el orden do
6:     if u no está marcado then
7:         S ← S ∪ {u}
8:         Marcar u y todos sus vecinos
9:     end if
10: end for
11: Salida: Conjunto S
```

1.2. Heurística Greedy Aleatorizado

En nuestra versión aleatorizada, se construye lo que se conoce como lista restringida de candidatos (un *RCL*, por sus siglas en inglés), que está compuesto por los k nodos con menor grado que aún no han sido descartados.

Entre los candidatos se elige un nodo de manera aleatoria y se incorpora al conjunto solución del MIS. Luego de añadir el nodo a la solución, tanto él como sus vecinos quedan marcados y posteriormente eliminados, del proceso de selección.

Esto introduce diversidad en las soluciones obtenidas con la finalidad de encontrar una configuración mejor para el problema.

Algorithm 2 Greedy aleatorizado para MIS

Entrada: Grafo $G = (V, E)$, parámetro k
Calcular el grado de cada vértice y almacenar en *degrees* como pares $\{grado, indice\}$
Ordenar *degrees* por grado ascendente
$S \leftarrow \emptyset$ // Conjunto independiente
Inicializar arreglo *marked* con todos los vértices marcados como falso
$unmarked \leftarrow |V|$ // Contador de vértices no marcados
while $unmarked > 0$ **do**
  Inicializar $RCL \leftarrow \emptyset$
  $counter \leftarrow 0$
  for $i = 1$ hasta $|degrees|$ y $counter \neq k$ **do**
    $node \leftarrow degrees[i].second$ // Obtener índice del vértice
    if $marked[node] =$ falso **then**
      Agregar *node* a RCL
      $counter \leftarrow counter + 1$
    end if
  end for
  Seleccionar índice aleatorio idx desde $[0, |RCL| - 1]$
  $node \leftarrow RCL[idx]$
  if $node = 0$ **then**
    continue // Saltar nodo 0
  end if
  if $marked[node] =$ falso **then**
    $S \leftarrow S \cup \{node\}$
    $marked[node] \leftarrow$ verdadero
    $unmarked \leftarrow unmarked - 1$
    for cada vecino *neighbor* de *node* **do**
      if $marked[neighbor] =$ falso **then**
        $marked[neighbor] \leftarrow$ verdadero
        $unmarked \leftarrow unmarked - 1$
      end if
    end for
  end if
end while
Salida: Conjunto S

2. Resultados

N	densidad (p)	Greedy		Greedy Aleatorio	
		Media	promedio tiempo (ms)	Media	promedio tiempo (ms)
1000	0.1	50.0	0.00021	48.47	0.00078
	0.2	27.93	0.00024	26.5	0.00060
	0.3	18.93	0.00024	17.93	0.00050
	0.4	14.33	0.00023	13.33	0.00039
	0.5	11.57	0.00023	10.2	0.00035
	0.6	9.63	0.00023	8.0	0.00032
	0.7	7.63	0.00021	6.4	0.00027
	0.8	6.2	0.00019	5.07	0.00024
	0.9	4.93	0.00017	3.97	0.00020
2000	0.1	57.07	0.00049	54.17	0.00171
	0.2	31.0	0.00051	29.13	0.00125
	0.3	20.77	0.00052	19.4	0.00100
	0.4	15.63	0.00051	14.73	0.00083
	0.5	12.1	0.00048	11.43	0.00074
	0.6	9.83	0.00046	8.97	0.00069
	0.7	8.37	0.00045	7.1	0.00060
	0.8	6.73	0.00040	5.63	0.00052
	0.9	5.3	0.00035	4.0	0.00042
3000	0.1	60.47	0.00073	58.57	0.00287
	0.2	32.03	0.00076	30.57	0.00197
	0.3	21.73	0.00076	20.6	0.00168
	0.4	16.37	0.00076	15.0	0.00131
	0.5	12.7	0.00073	11.77	0.00123
	0.6	10.17	0.00071	9.13	0.00108
	0.7	8.43	0.00067	7.23	0.00093
	0.8	6.83	0.00061	5.6	0.00077
	0.9	5.33	0.00054	4.33	0.00068

3. Análisis

3.1. Selección del parámetro k

El parámetro k controla el equilibrio entre **explotación** (comportamiento similar al Greedy determinista cuando k es pequeño) y **exploración** (mayor diversidad de soluciones cuando k es grande).

El valor de k se adaptó dinámicamente según el **tamaño del grafo** (n) y su **densidad** (p), siguiendo un criterio empírico específico. Para grafos de 1000 nodos, se utilizaron valores de $k = 20$ cuando la densidad era baja ($p \leq 0{,}3$), $k = 12$ para densidades medias ($0{,}3 < p \leq 0{,}6$), y $k = 8$ para densidades altas ($p > 0{,}6$). En el caso de grafos de 2000 nodos, estos valores se incrementaron a $k = 24$, $k = 14$ y $k = 8$ respectivamente para los mismos rangos de densidad. Finalmente, para los grafos más grandes de 3000 nodos, se emplearon $k = 30$ para densidades bajas, $k = 16$ para densidades medias, y $k = 10$ para densidades altas.

3.2. Resultados y Discusión

Lo primero que se observa es que, en todos los casos, el promedio de solución entregado por el algoritmo determinista resulta superior al obtenido con la versión aleatorizada. Esto ocurre porque el determinista no se arriesga en explorar soluciones potencialmente peores, pero dentro de ese promedio, no se descarta que una solución del algoritmo aleatorio de un mejor resultado que el determinista.

En cuanto al tiempo de ejecución, ambos algoritmos demuestran eficiencia y viabilidad. Sin embargo, el determinista es más eficiente que el aleatorizado, muy probablemente porque no debe construir ni utilizar el RCL.

Finalmente, se observa una tendencia clara en ambos casos: Mientras más densos son los grafos, más pequeño se hace el MIS obtenido. Esto se explica porque, al elegir candidatos según su grado, un grafo más denso implica que al seleccionar un nodo, se elimine un número mayor de vecinos, por lo que se reducen los candidatos posibles posteriores.

En conclusión, la implementación Greedy determinista es más consistente y rápido, pero el aleatorizado se arriesga más al explorar soluciones que inicialmente parecen infactibles. Si miramos más allá del promedio, puede ser ventajoso iterar con el algoritmo aleatorizado hasta que de una solución mejor que la del algoritmo determinista. Como línea de mejora, se plantea la idea de buscar un equilibrio entre la explotación y la exploración del algoritmo aleatorizado con una ponderación de los candidatos del RCL.