

Universidad Técnica Federico Santa María
Departamento de Informática

Instrucciones Entregable #2 DRP

Inteligencia Artificial

Sebastián Gallardo D.

sebastian.gallard.14@sansano.usm.cl

Noviembre 2020

1. Problemas a resolver

El objetivo general de esta segunda entrega es resolver el problema DRP: *Defibrillator Rellocation Problem*.

Su algoritmo debera resolver instancias del DRP en su variante *flexible location*; esto quiere decir, que existen AEDs previamente posicionados (conocidos), y se deben ubicar nuevos AEDs o cambiar de posición los ya existentes para maximizar la cobertura de OHCA's, y sin exceder el presupuesto asignado.

Como parámetro del programa, se entrega el presupuesto disponible n , el número total de eventos OHCA m , y el radio de cobertura r , fijo para cualquier AED ubicado en el mapa. Se considera que cambiar de posición un AED cuesta 0,2 y posicionar uno nuevo tiene costo 1.

La idea es que puedan comparar los resultados obtenidos con la literatura del MCLP; para ello, existen 2 versiones de cada instancia: en una de ellas **no hay ningún AED posicionado**, mientras que en la otra si. De esta forma, usando la primera versión de cada instancia pueden obtener resultados comparables a la literatura del MCLP, y con la segunda versión observar y discutir como cambia la solución.

Finalmente, lo que se busca es que puedan extraer conclusiones en base a lo obtenido, teniendo en cuenta las características de cada instancia.

Objetivo

El objetivo es maximizar la cobertura de eventos OHCA's existentes en el mapa, donde se considera que un evento esta cubierto si existe un AED a menos de r metros de distancia, donde r es un parámetro de entrada de la instancia.

Restricciones

- En primer lugar, cabe destacar que a modo de reducir el espacio de búsqueda, solo se pueden ubicar AEDs en posiciones ocupadas por un OHCA. Esto quiere decir que no se pueden buscar posiciones intermedias o “entre” OHCA's. Por ejemplo: si existe un evento en la posición (1,2) del mapa, y otro en la posiciones (3,4), un AEDs podría ser ubicado en (1,2) o en (3,4), pero no en una posición ubicada entre ambas.
- No se puede ubicar más de un AED en la misma posición
- Todos los AEDs tienen el mismo radio de cobertura r definido como parámetro de entrada.
- No se debe exceder el presupuesto n .
- Recordar que el costo de mover un AED es de 0,2, mientras que el costo de ubicar uno nuevo es de 1.

2. Instancias de prueba

Se entregan 2 carpetas con instancias.

La carpeta *SJC-DRP* corresponde a instancias extraídas desde <http://www.lac.inpe.br/~lorena/instancias.html> (sección *Max-Covering*, código SJC)[1]. Tal como se dijo previamente, estas instancias no tienen ningún AED previamente posicionado. La idea es que puedan comparar el desempeño obtenido con el trabajo citado y otros presentes en la literatura.

La segunda carpeta (*SJC-DRP-AEDs*), corresponde a las mismas instancias, pero con algunos AEDs ya instalados, y donde además puede cambiar el presupuesto. El número de eventos, la posición de los OHCA y el radio de cobertura se mantienen exactamente igual.

Todos los archivos tienen extensión *.txt*

2.1. Formato de instancias de prueba

El formato es el siguiente:

```
<#eventosOHCA> <presupuesto> <Radio de Cobertura>
<coordenada-x-OHCA #1> <coordenada-y-OHCA #1> <existeAED? #1>
<coordenada-x-OHCA #2> <coordenada-y-OHCA #2> <existeAED? #2>
<coordenada-x-OHCA #3> <coordenada-y-OHCA #3> <existeAED? #3>
...
<coordenada-x-OHCA #<#eventosOHCA>> <coordenada-y-OHCA #<#eventosOHCA>> <existeAED? #<#eventosOHCA>>
```

La primera línea indica información de la instancia en general: número total de eventos, presupuesto disponible, y el radio de cobertura de cada uno.

Luego, se tienen m líneas (donde m es el número total de eventos), con las coordenadas cartesianas de cada evento, y un tercer valor **booleano**, el cual tiene valor 1 **si es que existe un AED previamente instalado en esa posición**, y 0 en caso contrario. Se puede observar que en el caso de las instancias sin AEDs previamente instalados, este valor existe y es 0 para todos los casos, por lo que no deben preocuparse de separar los 2 tipos de instancias (su algoritmo las tratará de forma transparente para cualquier instancia).

2.2. Características de instancias de prueba

1. El valor de `<#eventosOHCA>` indica la cantidad de líneas siguientes.
2. Todos los valores son números enteros positivos o 0, salvo `<presupuesto>` que es un número de punto flotante.

3. Especificaciones del input del programa

El programa debe recibir como input (en línea de comandos) los parámetros (en este orden):

1. La ruta del archivo que contiene la instancia, por ejemplo: “instancias/SJC324-3.txt”.

Usted puede añadir otros parámetros adicionales, tal como cantidad de iteraciones máxima, tiempo de ejecución máximo, temperatura, cantidad de restarts, etc. Si lo hace, **debe incorporarlos junto a una descripción en el archivo README.txt, estableciendo al menos un ejemplo de como ejecutar su programa con estos parámetros.**

4. Especificaciones del output del programa

Su programa debe mostrar por pantalla, para ambos casos:

1. Número de eventos cubiertos (función objetivo).

2. Porcentaje de eventos cubiertos.
3. Tiempo de ejecución en segundos.
4. Cantidad de iteraciones alcanzada (si aplica).
5. Para cada AED:
 - a) Coordenadas x e y donde fue ubicado.
 - b) Indicar si corresponde a un nuevo AED o uno reposicionado, y si es lo segundo, indicar también su posición original.
6. Indicar presupuesto sobrante.

Si usan una técnica completa, deben mostrar por pantalla lo que su algoritmo construyó hasta el momento en que decidan terminar la ejecución abruptamente, con un Ctrl+C.¹ Si desea mostrar otros datos que considere relevantes (y que puedan visualizarse fácilmente), puede añadirlos al final del output, **especificando en el archivo README.txt su significado y que lo motivo a añadirlos a la salida del programa.**

5. Recordatorios y recomendaciones

1. Recuerde que su programa debe ser realizado en C/C++ con un Makefile para poder compilarlo.
2. Recuerde que debe comentar su código.
3. Recuerde incluir un README.txt donde se especifique:
 - a) Instrucciones de ejecución
 - b) Parámetros necesarios para ejecutar el algoritmo, con una explicación de cada uno (significado, objetivo del mismo, etc.)
 - c) Ejemplo(s) de comando(s) de ejecución.
 - d) En caso de requerirlo, el detalle de datos adicionales incluidos en el output
 - e) Cualquier supuesto que considere relevante.
4. Para realizar mediciones de tiempo de su programa hágalo usando “usr/bin/time²”, donde el tiempo de ejecución total de su programa será “user + sys”. Se recomienda usar un archivo bash para automatizar la ejecución de su programa con distintos parámetros usando time y guardar los resultados en un archivo.
5. Si tiene problemas con algunas instancias específicas producto de, por ejemplo, cómputos muy grandes o demasiada memoria ram utilizada que saturó su pc, usted debería hacer lo siguiente:
 - a) Explicar en el informe cuáles instancias fueron las que originaron el problema
 - b) Explicar cuál fue el problema que usted detectó: tamaño de la instancia muy grande, error aleatorio, etc.
6. Se recomienda leer la sección de experimentos de [2], en donde comparan diversos algoritmos de aproximación tomando como base las mismas instancias con las que usted trabajará.

¹<https://stackoverflow.com/questions/1641182/how-can-i-catch-a-ctrl-c-event>

²Para más información revisar la documentación oficial

Referencias

- [1] Luiz AN Lorena and Marcos A Pereira. A lagrangean/surrogate heuristic for the maximal covering location problem using hillman's edition. *International Journal of Industrial Engineering*, 9:57–67, 2002.
- [2] Andrej PODRRADSKY. Stable marriage problem algorithms. Diplomova prace, Masarykova univerzita, Fakulta informatiky, 2011 [cit. 2014-06-12].