

Práctica de Búsqueda Local

Centrales Eléctricas

Ariosa Hernández, Roberto

Bosch Llufríu, Júlia

Gómez Sánchez, Joaquín

QT - Curso 2018/19

Inteligencia Artificial

Índice

1. Elementos para la resolución del problema
2. Experimentación
 - a. Experimento 1 - Fijación de los operadores
 - b. Experimento 2 - Fijación de la estrategia para la gestión de las soluciones iniciales
 - c. Experimento 3 - Determinación de parámetros para el algoritmo Simulated Annealing
 - d. Experimento 4 - Evolución del tiempo de ejecución
 - e. Experimento 5 - Heurístico con penalización
 - f. Experimento 6 - Proporción del tipo de clientes
3. Comparación Hill-Climbing vs Simulated Annealing

Elementos para la resolución del problema

Para la resolución del problema dado, es decir, las asignaciones de un conjunto de clientes a centrales de una compañía eléctrica, de manera que los clientes con un suministro garantizado por contrato estén siempre suministrados, los que no tienen dicho contrato no lo estén necesariamente y que además se minimice la pérdida de producción energética y se maximicen los beneficios económicos, tenemos los siguientes elementos.

Implementación del Estado

En la implementación del estado hemos decidido, en primer lugar, tener dos Array Lists estáticos con la información esencial del problema i.e. el conjunto de clientes y el conjunto de centrales de los que dispone la compañía, generados aleatoriamente siguiendo unas proporciones de tipos.

En segundo lugar, para la representación del core del problema decidimos que lo esencialmente necesario para tener un control de las asignaciones y de la producción consumida por las centrales, con tal de no incumplir las restricciones del problema y además tener un trabajo fluido con los operadores, eran dos vectores enumerados a continuación:

- *representationClients* - Array simple de Java el cual mapea directamente los clientes del arraylist de clientes. Guarda por cada cliente la central que tiene asignado.
- *representationCentrales* - Array simple de Java el cual mapea directamente las centrales del arraylist de centrales. Guarda por cada central los MW que están siendo consumidos de esta.

Además de lo necesario para la representación del núcleo del problema, para agilizar el cálculo de la función heurística, nuestro estado incluye un parámetro *beneficio* con el beneficio económico para el estado actual y un parámetro *entropía* que contiene la penalización debido a la distancia entre los clientes y las centrales asignadas.

Debido a que es necesario actualizar dichos parámetros en función de los operadores aplicados, hay cuatro variables adicionales (*hCentral_old*, *hCentral_new*, *hCliente_old*, *hCliente_new*) que nos permiten hacerlo.

Operadores de búsqueda

El conjunto final de operadores que hemos decidido mantener debido a la variabilidad de estados que nos permite obtener, aumentado nuestra visión sobre el espacio de soluciones, es el enumerado a continuación:

- **Operador Assignment.** Conjunto de sub-operadores más sencillos.
 - **Un cliente de una central a otra.** Realiza un cambio de un cliente de una central a otra, permitiendo hacer cambios sencillos pudiendo llegarse a apagar centrales, maximizando las que están muy solicitadas y minimizando las que no lo están tanto.
 - **Un cliente a una central.** Asigna un cliente que no tiene central a una. Permite trabajar con los clientes no prioritarios además de compartir las mismas bondades que el sub-operador anterior.
- **Operador Swap.** Realiza el intercambio de clientes entre centrales. Permite jugar con las diferentes posibilidades del consumo de centrales (i.e. como el problema de la mochila) además de verse fuertemente afectado por el parámetro de la distancia entre los clientes y sus respectivas centrales.

Estrategias para las soluciones iniciales

Para las solución inicial hemos planteado dos estrategias iniciales, ambas solución dentro del espacio de soluciones.

- **Solo Prioritarios.** Se realiza una asignación a centrales únicamente de los clientes que tienen un suministro garantizado. Se van asignando los clientes a una central hasta que esta no puede suministrar más producción pasándose entonces a hacer asignaciones con la siguiente central en el arraylist de centrales.
- **Todos.** Se intenta realizar la asignación de todos los clientes. Igual que con la asignación de solo prioritarios, se van asignando los clientes a una central hasta que se cubre su producción y se pasa a hacer las asignaciones a la siguiente central hasta que se completa su producción y así sucesivamente. Se garantiza que al menos todos los clientes garantizados estén asignados.

Función Heurística

Partiendo del hecho de que nuestro objetivo es maximizar el beneficio, y dado que para ello debemos reducir los costes innecesarios y además ser eficientes en las asignaciones, es necesario para nuestros operadores que influya en el heurístico la pérdida en MW debida las distancias, haciendo que intente reducir este factor para poder a su vez ser más eficiente en las asignaciones y poder obtener más beneficio.

El heurístico que hemos creído conveniente para dicho criterio es el siguiente:

$$f_{heurística} = Beneficio_{Total} - Coste_{MW \text{ por Distancia}}$$

A rasgos más específicos el beneficio es la suma de las tarifas de los clientes menos los costes asociados a las centrales y sus producciones. El coste en MW por las distancias es el total de MWs perdidos debido a la distancia multiplicados por el coste que supone la producción de dichos MW.

En base a todo esto, la función heurística desarrollada correspondiente:

$$f_{heurística} = \left(\sum_{i=0}^n Tarifa_{Cliente\ Garantizado} + \sum_{i=0}^n Tarifa_{Cliente\ No\ Garantizado} - \sum_{i=0}^n Penalización_{Cliente\ no\ Garantizado\ no\ Suministrado} - \sum_{j=0}^m Coste_{Central\ en\ Marcha} - \sum_{j=0}^m Coste_{Central\ en\ Parada} - \sum_{j=0}^m Coste_{Producción\ MW\ de\ Central\ en\ Marcha} \right) - 5 \cdot \sum_{i=0}^n (Porcentaje_{Pérdida\ en\ función\ de\ la\ Distancia} \cdot Consumo_{Cliente_i})$$

Experimentación

Experimento 1 - Fijación de los operadores

Para este primer experimento con el algoritmo Hill Climbing y los parámetros fijos:

- Número de centrales: 5 (A), 10 (B) y 25 (C).
- Clientes: 1000.
- Proporción de clientes: 25% (XG), 30% (MG) y 45% (G).
- Proporción de clientes con suministro garantizado: 75%.
- Tipo de solución inicial: Todos.

En primer lugar, hemos realizado diez iteraciones utilizando únicamente el operador Assignment, después otras diez iteraciones únicamente con el operador Swap, y finalmente, el mismo número de iteraciones utilizando ambos operadores.

Las seed utilizadas para las diez iteraciones son: 1234, 101, 10001, 4, 51, 10, 2, 176, 234, 515.

Los promedios de los heurísticos obtenidos son los siguientes:

$$PM_{Assignment} = 156.585,2 \quad PM_{Swap} = 148.594,3 \quad PM_{Ambos} = 178.758,4$$

Podemos observar que utilizando ambos operadores adquirimos un heurístico promedio mayor ya que ampliamos nuestro espacio de soluciones pudiendo llegar a estados de mayor calidad que utilizando únicamente uno de los dos operadores.

Esto último era lo esperado dado que al utilizar más operadores, comprobando la validez de sus acciones, ampliamos la visión que el algoritmo tiene sobre el espacio de soluciones pudiendo moverse más libremente por él. Por ejemplo, para el escenario dado, con solo asignaciones podemos generar como máximo 40.000 estados, para solo swap máximo 500.000 estados, y para ambos la suma de los dos anteriores, es decir, 540.000 estados.

Experimento 2 - Fijación de la estrategia para la generación de las soluciones iniciales

Partiendo del escenario del experimento uno, procedemos a realizar una ejecución utilizando ambos operadores, ya que han dado mejor resultado promedio, pero en este caso utilizando la solución inicial que asigna únicamente los prioritarios.

Para esta solución obtenemos un heurístico promedio:

$$PM_{\text{Ambos-Prioritarios}} = 189.239,8$$

que es mayor que el heurístico obtenido en el experimento anterior.

Esta solución inicial es mejor debido a que da al algoritmo mayor margen de maniobra al dejar más clientes sin asignar y más centrales no ocupadas o no tan al límite.

Experimento 3 - Determinación de parámetros para el algoritmo Simulated Annealing

Para determinar los parámetros a utilizar para el algoritmo Simulated Annealing hemos obtenido diferentes heurísticos dejando fijos los steps y el stiter, modificando únicamente k y lambda, que son los parámetros que determinan la función de aceptación de estados.

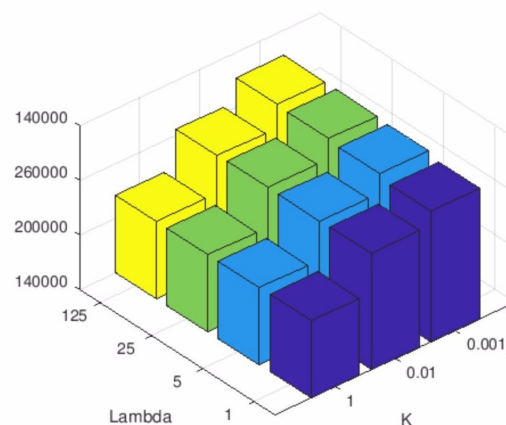


Figura 1 - Lambda vs K

Como podemos observar en el diagrama de barras anterior, para $k = 125$ y $\lambda = 0,001$ obtenemos un aumento considerable en el heurístico por lo que es interesante girar entorno a valores de ese estilo.

Tras diferentes pruebas el mejor resultado obtenido es 253.431 utilizando como parámetros: $Iteraciones = 50.000.000$, $Stitet = 100$, $k = 125$, $\lambda = 0,001$.

Experimento 4 - Evolución del tiempo de ejecución

Dado el escenario del experimento uno y modificando el número de clientes progresivamente obtenemos la siguiente representación del tiempo de ejecución:

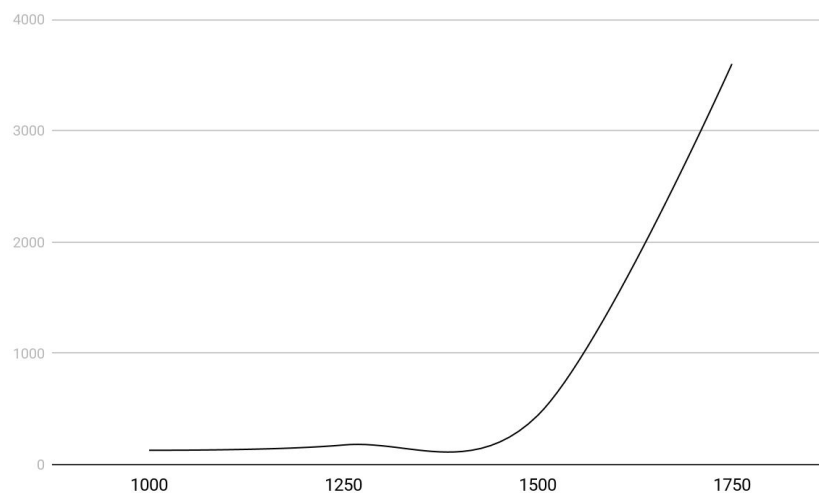


Figura 2 - Tiempos de ejecución con HC

Podemos observar que la función del tiempo de ejecución se aproxima a una función de tipo claramente exponencial, dado que cuando aumentamos el número de clientes aumentamos con carácter exponencial el tiempo de ejecución. Esto es debido al operador Swap, ya que realiza todos los intercambios posibles entre clientes por las centrales.

Experimento 5 - Heurístico con penalización

Para hacer este experimento hemos adaptado la función de evaluación para que sea capaz de añadir una penalización por cada cliente garantizado al cual no se le puede servir. Para analizar cómo terminan las ejecuciones y sus cambios dependiendo de la penalización, empezamos siempre con una solución (parcial) vacía.

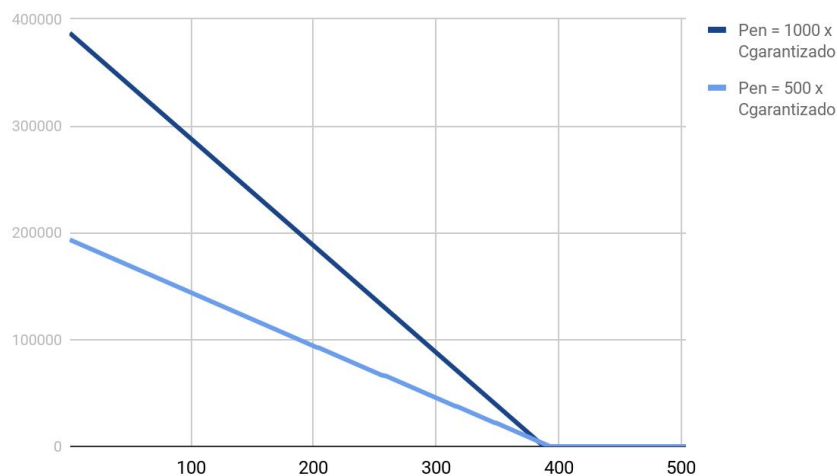


Figura 3 - Penalización por iteración del algoritmo

Debido a nuestro heurístico no hay diferencia en la solución ya que no contemplamos una posible afectación por la penalización.

Experimento 6 - Proporción del tipo de clientes

Un factor que puede ser determinante a la hora de intentar reducir distancias entre centrales y clientes es intentar introducir más centrales pequeñas pero más cercanas a los clientes.

Experimentando con este tipo de centrales HC obtenemos la tabla de valores:

A	B	C	Uso de A	Uso de B	Uso C	Tiempo Ejec (s)
5	10	25	5	10	25	418
5	10	50	5	10	43	12,29
5	10	75	5	10	43	14,75

Y para SA obtenemos:

A	B	C	Uso de A	Uso de B	Uso C	Tiempo Ejec (s)
5	10	25	5	10	25	5,75
5	10	50	5	10	47	5,92
5	10	75	5	10	71	6,233

Tal y como aparece en ambas tablas debido a los operadores que utilizamos y que no podemos garantizar la distribución de estas centrales más pequeñas, el número de centrales A y B no disminuye para el escenario planteado en el experimento uno.

Comparación Hill-Climbing vs Simulated Annealing

Con la representación del estado obtenida tras realizar todos los experimentos y con unos parámetros para el Simulated Annealing:

Iteraciones = 50.000.000

Iteraciones por cambio de temperatura = 100

$k = 10$

$\lambda = 0,001$

En promedio, los heurísticos obtenidos con el SA son mayormente superiores a los obtenidos con HC, pudiéndose obtener valores superiores a 200.000 que con HC con el escenario planteado en el experimento uno no se han podido obtener para ninguna seed.

Que los valores del heurístico para SA sean mayores es debido a que, aunque siempre mantengamos la misma seed, gracias al factor de temperatura y a los saltos que podemos realizar dentro del espacio de soluciones del problema somos capaces de llegar a soluciones que HC posiblemente no contempla debido a que siempre se encamina a mínimos/máximos locales próximos al heurístico del estado inicial.