Práctica 2.a Búsquedas Multiarranque para el Problema de la Asignación Cuadrática

Alejandro Alcalde*

Tercer curso Grado en Ingeniería Informática, Granada, grupo de los miércoles a las 17.30

algui91@gmail.com

Algoritmos: Generacional con pos y PMX, Estacionario con pos y PMX

Índice

*Template by howtoTeX.com, elbauldelprogramador.com

Resumen

Memoria de la tercera práctica de la asignatura metaheurísticas del tercer curso del Grado en Ingeniería Informatica de la facultad de Granada. La practica consiste en la implementación de dos tipos de algoritmos genéticos, uno generacional y otro elitista. Además, se han implementado dos tipos de operadores de cruce, PMX y basado en posición, lo que hace un total de cuatro implementaciones de algoritmos genéticos.

I. Descripción del problema

El problema de la asignación cuadrática (QAP) es un problema estándar en teoría de localización. En éste se trata de asignar N unidades a una cantidad N de sitios o localizaciones en donde se considera un costo asociado a cada una de las asignaciones. Este costo dependerá de las distancias y flujo entre las unidades, además de un costo adicional por asignar cierta unidad a cierta localización específica. De este modo se buscará que este costo, en función de la distancia y flujo, sea mínimo.

I.1. Definición matemática

$$\min_{S \in \prod_N} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \cdot d_{S(i)S(j)} \right)$$

Donde $\prod_N$ es el conjunto de todas las permutaciones posibles de $N = 1, 2, \ldots, n$

I.2. Representación del problema

En éste problema las soluciones se pueden representar como permutaciones de un conjunto. Si el problema es de tamaño cuatro, por ejemplo, una solución vendría dada por la permutación $N = \{3, 2, 1, 4\}$. Si tomamos los índices de éste conjunto como las unidades, y el valor en dicho índice como localizaciones, la localización 3 estaría asignada a la unidad 1, la localización 2 a la unidad 2 etc.

El objetivo del problema es **minimizar** la expresión mostrada anteriormente. Ésta será la función objetivo:

$$\min_{S \in \prod_N} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \cdot d_{S(i)S(j)} \right)$$

Hará falta un mecanismo para **generar la solución inicial**, en nuestro caso, la permutación N inicial sobre la que lanzar los distintos algoritmos será aleatoria.

De igual modo será necesario un esquema de **generación de soluciones vecinas**. Para ello, se realizará un intercambio entre dos localizaciones, cambiando sus respectivas unidades por la otra.

II. Composición de los algoritmos

II.1. Esquema de representación

Como se ha mencionado anteriormente, el esquema elegido para representar una solución ha sido una permutación del tipo $\{1, 2, 3, 4\}$ en la que los elementos representan localizaciones y la posición que ocupan las unidades a las que han sido asignadas.

II.2. Descripción de la función objetivo

La función objetivo descrita con su fórmula en apartados anteriores puede representarse con el siguiente pseudocódigo:

```
i = 0 .. Tamaño del problema
    j = 0 .. Tamaño del problema
        acumular el coste producido al asociar el flujo existente entre
        la unidad i y j y la distancia existente entre las localizaciones
        i y j
```

II.3. Generación de soluciones aleatorias

El mecanismo de generación de soluciones aleatorias tanto para ILS como para BMB ha sido el siguiente:

```
solucion aleatoria = []
mientras el tamaño de la solución no sea correcto
    generar un número aleatorio entre 0 y el tamaño de la solución
        si el número ya está en la solución descartarlo y generar uno nuevo
```

II.4. Descripción de la BL

La búsqueda local usada utiliza el mecanismo Dont Look Bits para ahorrar tiempo de exploración innecesario. Se realizan un total de 10000 evaluaciones:

```
mientras no se cumpla el criterio de parada
    recorrer la DLB
        Si el bit está a 0
            Explorar el entorno
                Realizar un intercambio
                Si se mejora aplicar el intercambio
            Si se explora todo el entorno poner el bit de la DLB a 1
```

El operador de vecino simplemente intercambia una posición de la solución con otra, y luego se aplica la siguiente función de factorización para ahorrar el cálculo de la función objetivo:

$$\sum_{k \neq r,s} \begin{bmatrix} f_{rk} \cdot (d_{\pi(s)\pi(k)} - d_{\pi(r)\pi(k)}) + f_{sk} \cdot (d_{\pi(r)\pi(k)} - d_{\pi(s)\pi(k)}) + \\ f_{kr} \cdot (d_{\pi(k)\pi(s)} - d_{\pi(k)\pi(r)}) + f_{ks} \cdot (d_{\pi(k)\pi(r)} - d_{\pi(k)\pi(s)}) \end{bmatrix}$$

III. ESTRUCTURA DEL MÉTODO DE BÚSQUEDA

Para el algoritmo BMB se ha seguido el siguiente esquema:

```
Generar 25 soluciones aleatorias
    Para cada una de las soluciones
        Aplicar búsqueda local
        Si la solución dada por la BL mejora la mejor hasta el momento
            Actualizar mejor solución
```

Para el algoritmo ILS se ha seguido el siguiente esquema:

```
Generar una solución aleatoria inicial
Aplicarle BL a dicha solución
Si mejora actualizar la mejor solución
Desde 0 .. 23
    mutar la solución actual
    Aplicar BL a la solución mutada
    Si mejora actualizar la mejor solución encontrada
```

El operador de mutación consistente en seleccionar una cadena consecutiva de asignaciones y reasignarlas aleatoriamente, la cadena seleccionada deberá ser lo suficientemente grande para que el cambio sea significativo, en este caso t = n/4. El procedimiento seguido es el siguiente:

```
Generar un número aleatorio entre 0 y el tamaño del problema
Calcular cuantos elementos en total habrá que cambiar
Guardar el índice superior
Guardar el índice actual
Para cada uno  de los elementos a cambiar
    generar un nuevo índice entre el índice inferior y el superior
    intercambiar el índice generado con el actual
    actualizar el índice actual a una unidad más
```

IV. Algoritmo de comparación, Greedy

Consiste simplemente en asociar unidades de gran flujo con localizaciones céntricas en la red y viceversa. Para ello se calcula el potencial de flujo y de distancia. A mayor potencial de flujo, más peso tendrá dicha unidad en el intercambio de flujos y, a menor flujo de distancia, más céntrica será la localización. Por tanto el algoritmo selecciona la unidad disponible con mayor potencial de flujo y le asignará la localización de menor potencial de distancia.

```
    Calculo potenciales.
    Inicialzar N a 0.
    para i = 1 hasta el tamaño del problema
        Seleccionar unidad de mayor potencial de flujo
        Seleccionar localización de menor potencial de distancia
        Añadir la localización seleccionada al índice correspondiente a la unidad en N
```

V. Procedimiento considerado

La implementación se ha realizado en Python, basándose en las explicaciones de clase y los documentos proporcionados por el profesorado. Para ejecutar el programa:

```
$ python QAP.py -d <datos del problema> -a [ils|bmb] -s semilla -v<verbose>
```

VI. Experimentos y análisis de resultados

Para esta pactica se ha usado todos los casos, salvo los bur. La semilla usada fue 2704647398.
A continuación se muestra la tabla con los resultados para los distintos algoritmos.

Algoritmo	Desviación	Tiempo
Greedy	81.4492028061	0.0012543466
BMB	5.9915214967	5.6860943106
ILS	5.9334093738	5.2941380077

Cuadro 1

BMB			
Caso	Coste	Desv	Tiempo
Els19	17997928	4.56	0.6709640026
Chr20a	2670	21.81	0.4514892101
Chr25a	4976	31.09	1.0165770054
Nug25	3828	2.24	0.8726320267
Tai30a	1890498	3.98	1.870486021
Tai30b	650861550	2.16	2.8595149517
Esc32a	144	10.77	2.0284779072
Kra32	92860	4.69	1.902946949
Tai35a	2513334	3.77	2.9234039784
Tai35b	287760798	1.57	4.9026348591
Tho40	246460	2.47	5.5052878857
Tai40a	3266848	4.06	4.134128809
Sko42	16070	1.63	6.004180193
Sko49	23804	1.79	9.6663868427
Tai50a	5138272	4.04	8.1470592022
Tai50b	470134271	2.47	12.7287528515
Tai60a	7491980	3.97	11.8476259708
Lipa90a	363473	0.79	24.8171489239

Cuadro 2

Ambos algoritmos son bastante parecidos en cuanto a tiempo de ejecución y resultados. Es posible que éstos resultados estén más influenciados por la búsqueda local, la cual converge muy rápidamente, y debido a ésto los resultados son muy similares para ambos casos.

ILS			
Caso	Coste	Desv	Tiempo
Els19	19612684	13.94	0.3921871185
Chr20a	2870	30.93	0.373308897
Chr25a	4964	30.77	0.8017849922
Nug25	3758	0.37	0.8408501148
Tai30a	1860086	2.31	1.3652291298
Tai30b	654878555	2.79	1.889136076
Esc32a	134	3.08	1.4711718559
Kra32	90570	2.11	1.5701570511
Tai35a	2458870	1.52	2.496491909
Tai35b	294069830	3.80	2.8392338753
Tho40	244544	1.67	3.84405303
Tai40a	3249646	3.51	3.4696700573
Sko42	16012	1.26	4.3035180569
Sko49	23676	1.24	6.9809579849
Tai50a	5082294	2.91	6.7509710789
Tai50b	460949433	0.46	10.516602993
Tai60a	7451524	3.41	11.595993042
Lipa90a	363213	0.72	33.7931668758

Cuadro 3