Práctica 3.a: Algoritmos Genéticos 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 tamaño de la población para esta práctica debe ser de 50, por lo tanto es necesario generar 50 soluciones iniciales aleatorias. Para así disponer de una primera población. El mecanismo seguido para obtener la población inicial, dada una semilla inicial, es el siguiente:

```
desde i=0 hasta el tamaño de la población
    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
    añadir la solución aleatoria a la población
```

Éste proceso se realiza al principio de la ejecución del algoritmo.

II.4. Mecanismo de selección

El mecanismo de selección escogido ha sido el torneo binario, éste método consiste en escoger aleatoriamente dos individuos de la población, quedarse con el mejor de ambos y luego volver a coger otros dos y repetir el proceso. Los ganadores de los torneos formarán parte de la población en la siguiente generación. La diferencia entre el esquema generacional y el estacionario reside en que el primero realiza tantos torneos como individuos existan en la población, mientras que el último realiza dos veces el torneo para seleccionar únicamente dos individuos.

Para la versión generacional la selección se ha implementado de la siguiente forma:

```
desde i = 0 hasta el tamaño de la población
    seleccionar aleatoriamente dos individuos de la población distintos
    añadir a la siguiente generación el mejor individuo de los dos
```

La versión estacionaria es un caso particular de la anterior en la que se realiza el torneo dos veces.

II.5. Operador de cruce

Una vez se ha obtenido una nueva población mediante la selección anterior es necesario realizar un cruce entre sus miembros bajo una cierta probabilidad. Para evitar la generación de múltiples números aleatorios se ha optado por usar la esperanza matemática para calcular de antemano cuantos individuos cruzarán.

El cruce consiste en combinar dos padres para obtener dos hijos que preserven de alguna manera la calidad de ambos padres. Se han considerado dos tipos de operadores, basado en posición y PMX. La descripción de ambos se muestra a continuación:

```
# Basado en posición
desde i = 0 hasta esperanza matemática del cruce
    - Se seleccionan dos padres consecutivos de la población
    - Se comprueba qué elementos coinciden en los dos padres y se añaden al hijo1 y hijo2
    - Los elementos que no coinciden se desordenan y se añaden al hijo1
    - Se vuelven a desordenar los elementos que no coindicen y se añaden al hijo2
    - Se sustituye a los dos padres seleccionados con los dos hijos
    - Marcar los dos nuevos individuos para su posterior evaluación
```

```
# PMX
desde i = 0 hasta esperanza matemática del cruce
    - Se seleccionan dos padres consecutivos de la población
    - Generar dos puntos de corte aleatorios, c1 entre 0 y n, y c2 entre c1 y n
    - Copiar en el hijo1 los elementos entre los puntos de corte del padre2
    - Copiar en el hijo2 los elementos entre los puntos de corte del padre1
    - Generar las correspondencias existentes de los elementos de ambos padres
      entre los puntos de corte
    - Obtener una lista de índices sin elementos asignados en los hijos
    - Obtener qué elementos ya existen en el hijo1 y hijo2
    - Recorrer los índices si rellenar en el hijo1
        - Si el elemento a insertar del padre1 ya existe en el hijo1
            - Sustituirlo por la correspondencia adecuada
        - Si no, agregar el elemento del padre1 al hijo1
    - Recorrer los índices si rellenar en el hijo2
        - Si el elemento a insertar del padre2 ya existe en el hijo2
            - Sustituirlo por la correspondencia adecuada
        - Si no, agregar el elemento del padre2 al hijo2
    - Sustituir a los dos padres seleccionados por los dos hijos
    - Marcar los dos nuevos individuos para su posterior evaluación
```

Hay una pequeña diferencia respecto al esquema generacional y el estacionario. El generacional realiza tantos cruces como su esperanza matemática indique. El estacionario debe cruzar con una probabilidad de 1, ya que únicamente selecciona dos padres.

II.6. Operador de mutación

La aplicación de la mutación también se basa en una probabilidad. Al igual que en el cruce, para evitar la generación innecesaria de números aleatorios, se calculan con antelación la cantidad exacta de elementos a mutar en base a esa probabilidad. El operador de mutación es el mismo usado en prácticas anteriores, escoger dos posiciones aleatorias e intercambiarlas.

En esta caso el procedimiento es el mismo para ambos esquemas:

```
desde i = 0 hasta la esperanza matemática para la mutación
    - Generar un número aleatorio que seleccionará el elemento de la población a mutar
    - Generar dos números más para decidir qué genes (elementos) del individuo intercambiar
    - Aplicar el intercambio y marcar el individuo para su posterior re-evaluación
```

En el estacionario, la selección del individuo se realizará sobre dos elementos.

II.7. Evaluación

Para evitar realizar cálculos innecesarios, durante una generación se marcan los individuos que necesitarán ser re-evaluados. Calculando únicamente el coste de dichos individuos en el caso de estar marcados para re-evaluación. En esta etapa se guarda el mejor individuo de la población.

III. Estructura del método de búsqueda

III.1. Esquema de evolución y reemplazamiento generacional

Éste esquema consiste en seleccionar una población de igual tamaño a la población genética. En nuestro caso la población es de 50. Por tanto, en cada iteración (generación) se guarda la población anterior como población vieja y se sustituye por una nueva con el mecanismo de selección, también de tamaño 50. A continuación se muestra el esquema de evolución:

```
Inicializar población
Evaluar población

mientras no se cumpla el criterio de parada
    Intercambiar la población vieja por una nueva
    Seleccionar individuos de la población vieja para formar parte de la nueva
    Cruzar
    Mutar
    Reemplazar
    Evaluar
```

En el esquema de reemplazamiento se sustituye la generación vieja por la nueva. Pero se pretende conservar el elitismo. Para ello, si el mejor individuo de la población vieja no está en la nueva, se sustituirá la peor solución de la población actual con éste individuo. El pseudocódigo es el siguiente:

```
Si el mejor individuo de la población anterior no está en la nueva
    - Buscar la peor solución de la población actual y sustituirla por el
      individuo de la población anterior.
```

III.2. Esquema de evolución y reemplazamiento estacionario

En éste esquema se seleccionan únicamente dos padres, los cuales se cruzarán, mutarán. El reemplazamiento se realizará únicamente en el caso que se produzca una mejora, es decir, si algunos de los individuos generados es mejor que los peores de la población.

```
Inicializar población
Evaluar población

mientras no se cumpla el criterio de parada
    Seleccionar 2 individuos de la población
    Cruzarlos
    Mutar con probabilidad
    Reemplazar
    Evaluar
```

El reemplazamiento:

```
Si el peor de la población es peor que el primer individuo generado
    - Sustituirlo
Si el segundo peor de la población es peor que el segundo individuo generado
    - Sustituirlo
Si el peor de la población es peor que el segundo individuo generado
    - Sustituirlo
Si el segundo peor de la población es peor que el segundo generado
    - Sustituirlo
```

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 \
[elitist_gg_pos|elitist_gg_pmx|stacionary_gg_pos|stacionary_gg_pmx] -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. Haciendo pruebas se descubrió un valor cercano al óptimo para la instancia nug25. El valor óptimo de ésta instancia es 3744, con la semilla 1411423505 se obtuvo un coste de 3764.

A continuación se muestra la tabla con los resultados para los distintos algoritmos.

Algoritmo	Desviación	Tiempo
Greedy	81.4492028061	0.0012273788
AGG-pos	10.425309152	18.3893676599
AGG-PMX	10.1836523477	17.2611667977
AGE-pos	44.6100486995	14.826227612
AGE-PMX	43.8634597082	14.8372759422

Cuadro 1

Genético Generacional POS			
Caso	Coste	Desv	Tiempo
Els19	17535402	1.88	10.4644160271
Chr20a	2838	29.47	8.3315520287
Chr25a	5942	56.53	12.3692309856
Nug25	3904	4.27	16.6173238754
Tai30a	1913302	5.23	15.8454630375
Tai30b	709753517	11.40	15.0750370026
Esc32a	160	23.08	10.9133908749
Kra32	91770	3.46	12.6483390331
Tai35a	2559336	5.67	15.4616510868
Tai35b	305457393	7.82	14.1183140278
Tho40	248844	3.46	16.4316048622
Tai40a	3278940	4.45	16.9408519268
Sko42	16410	3.78	16.6719579697
Sko49	24372	4.22	20.7642681599
Tai50a	5295072	7.21	21.1576180458
Tai50b	486611071	6.06	21.464343071
Tai60a	7815502	8.46	27.776927948
Lipa90a	364988	1.21	57.9563279152

Cuadro 2

Observando las tablas queda patente que la diferencia entre los operadores de cruce no es muy significativa. El PMX es ligeramente mejor, probablemente debido a que realiza cortes en las soluciones padres y las pega en los hijos, lo que ayuda a conservar gran parte del valor de la solución del padre en el hijo. El PMX intenta combinar los dos padres conservando en la medida de lo posible las posiciones de ambos en sus hijos.

Una razón por la cual el operador basado en posición pueda actuar un poco peor es que se basa únicamente en conservar las posiciones coincidentes en ambos padres, haciendo así que la porción de solución que se conserva tienda a ser menor que en PMX. El factor de aleatoriedad que se aplica en este operador para reasignar las posiciones sobrandes puede proporcionar tanto soluciones malas como buenas. Por tanto el PMX tiende a conservar más las propiedades de los padres.

En cuanto a los tos esquemas, generacional o estacionario, está claro que el estacionario, por su forma se seleccionar población siempre va a explorar menor cantidad en el espacio de soluciones, ya que por cada generación se seleccionan únicamente dos elementos para cruzar y mutar, mientras que la versión generacional explorará a un ritmo de 50 individuos por generación. Realizándole además mutaciones y cruces a los individuos correspondientes.

Genético Generacional PMX			
Caso	Coste	Desv	Tiempo
Els19	17937024	4.21	6.3874239922
Chr20a	2896	32.12	7.4710178375
Chr25a	6084	60.27	8.6566560268
Nug25	3920	4.70	9.7244329453
Tai30a	1889940	3.95	12.336566925
Tai30b	691796918	8.58	12.4235701561
Esc32a	156	20.00	10.8866541386
Kra32	94510	6.55	11.614978075
Tai35a	2518224	3.97	13.8358850479
Tai35b	302709491	6.85	13.8314809799
Tho40	253804	5.52	15.98488307
Tai40a	3311914	5.50	15.962831974
Sko42	16434	3.93	17.0838091373
Sko49	23950	2.41	21.2035839558
Tai50a	5181558	4.92	21.168197155
Tai50b	469403774	2.31	21.6532700062
Tai60a	7673216	6.48	28.1585299969
Lipa90a	364356	1.03	62.3172309399

Cuadro 3

Genético Estacionario POS			
Caso	Coste	Desv	Tiempo
Els19	24385666	41.67	4.5428960323
Chr20a	5414	146.99	4.8091681004
Chr25a	11466	202.05	6.5609920025
Nug25	4404	17.63	6.5850667954
Tai30a	2072420	13.99	8.5201299191
Tai30b	817177373	28.26	8.8779101372
Esc32a	294	126.15	9.3563039303
Kra32	118480	33.57	9.7311291695
Tai35a	2773072	14.50	10.8184459209
Tai35b	370803500	30.88	10.9540109634
Tho40	302924	25.95	13.8538200855
Tai40a	3578576	13.99	13.3199338913
Sko42	18734	18.48	14.8727619648
Sko49	27392	17.13	19.4602730274
Tai50a	5666064	14.73	19.6195249557
Tai50b	646546149	40.91	20.1130871773
Tai60a	8246186	14.44	27.0534820557
Lipa90a	366624	1.66	57.8231608868

Cuadro 4

Genético Estacionario PMX			
Caso	Coste	Desv	Tiempo
Els19	24094508	39.98	4.6574339867
Chr20a	5386	145.71	4.9204189777
Chr25a	10854	185.93	6.6915891171
Nug25	4322	15.44	6.609251976
Tai30a	2067496	13.71	8.4478189945
Tai30b	870376734	36.61	8.62398386
Esc32a	284	118.46	9.2911510468
Kra32	118060	33.10	9.5825548172
Tai35a	2783710	14.93	10.8890759945
Tai35b	392475721	38.53	11.1784491539
Tho40	303358	26.13	13.6819250584
Tai40a	3606472	14.88	13.4146249294
Sko42	18566	17.42	14.84770298
Sko49	27258	16.56	19.39554286
Tai50a	5659200	14.59	19.5065131187
Tai50b	649205197	41.49	19.9771239758
Tai60a	8246650	14.44	26.7352631092
Lipa90a	366480	1.62	58.6205430031

Cuadro 5