

Metaheurísticas. Trabajo de Evaluación

Frédéric Roux, PhD

November 9, 2019

1 Problema de la asignación cuadrática y problema de la partición del grafo

El problema de asignación cuadrática (QAP) y el problema de la partición de un grafo pertenecen al campo de la optimización combinatoria [1-2]. Los dos problemas son problemas computacionales complejos dicho de tipo "NP-hard", es decir se requieren algoritmos no-deterministas basados en heurísticas para encontrar una solución aproximada, ya que una solución exacta de este tipo de problema no es factible en tiempo computacional polynomial [3-4].

1.1 Descripción del problema de asignación cuadrática (QAP)

El QAP fue introducido por Koopmans y Beckman en 1957 en el contexto de la localización de actividades económicas indivisibles [5-6]. El problema consiste en asignar un conjunto de n instalaciones a un conjunto de n localizaciones de manera que se minimize el coste de utilización de las instalaciones. El coste de utilización entre dos instalaciones es dado por la suma del flujo entre estas instalaciones multiplicado por la distancia entre sus localizaciones asignadas. Si denotamos por $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(n))$ una posible asignación, y por f_{kl} el flujo entre las instalaciones k y l , y por d_{ij} la distancia entre las instalaciones i y j , el objetivo del QAP es de minimizar la función objetiva dado por:

$$f(\sigma) = \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{\sigma(i)\sigma(j)} \quad (1)$$

1.2 Descripción del problema de la bi-partición de grafo

El problema de partición de grafo pertenece al campo de los problemas de partición y puede ser resumido por el ejemplo sencillo siguiente [7]: dado un conjunto de enteros 2 10 3 8 5 7 9 5 3 2, cual es la partición del conjunto S de enteros que resulta en dos conjuntos $S1$ y $S2$ con sumas iguales? Un ejemplo de partición posible seria $S1 = 2 5 3 10 7$ y $S2 = 2 5 3 9 8$ dado que la suma de cada conjunto es 27 [6]. En el caso del problema de la bi-partición

del grafo (BIPART), el problema consiste en particionar un grafo no dirigido $G = (\chi, U)$ con $\chi \neq \emptyset$ y $U = \{(u, v) \mid u, v \in \chi\}$ en el que cada arista (u, v) tiene asociado un peso $p_{(u, v)}$ y el numero de vertices es par en dos particiones. Por lo tanto, en el problema BIPART se trata de dividir el conjunto de vértices en dos subconjuntos $S1$ y $S2$ iguales, de forma que se minimice la suma de los pesos asociados a aristas que unen vértices de diferentes conjuntos. El objetivo del problema de partición de un grafo es de maximizar la función objetiva dado por:

$$f(G) = \sum_{i=1}^n p_{ij} \quad (2)$$

2 Propuesta de solución basada en búsqueda local

Con el objetivo de aproximar soluciones para el problema QAP y el problema BIPART, el trabajo presente ha utilizado como primer enfoque metaheurísticas dichas "sin memoria" (memory-less). Mas concretamente, para resolver el problema QAP se ha utilizado la búsqueda local con multiarranque, mientras que para resolver el problema BIPART se ha utilizado el greedy randomized adaptive search procedure (GRASP).

En el trabajo presente, la búsqueda local se ha implementado según los pasos descritos en el pseudo-código siguiente:

Seleccionar una solución inicial $e_0 \in \mathcal{E}$
Repetir
 Elegir $e \in V(e_0)$ tal que $f(e) < f(e_0)$
 Asignar e a e_0
hasta $f(e) \geq f(e_0) \quad \forall e \in V(e_0)$
 e_0 es la aproximación a la solución óptima

Basicamente, la búsqueda local empieza con una solución inicial y busca en su vecindad una mejor solución. Si la solución encontrada es mejor que la solución actual, la solución actual es remplazada por la encontrada y continua el proceso, hasta que no se puede mejorar la solución actual. Merece destacar que el diseño de la vecindad es crucial para el desempeño de la búsqueda local (y de muchos otros algoritmos de optimización).

La ventaja principal de la búsqueda local es de encontrar soluciones rápidamente, mientras que su principal desventaja es que queda fácilmente en mínimos locales y que su solución final depende de la solución inicial. Dado una misma entrada, la búsqueda local regresa siempre a la misma salida, propiedad que se puede ver como un pozo de atracción.

2.1 Multiarranque y búsqueda local para QAP

El diseño de la vecindad y el método mutiarranque en el trabajo presente ha sido implementado según el pseudo-código siguiente:

```
possibleNeighbours

  Let n be the number elements in a solution
  Let c be a matrix with  $n(n-1)/2$  rows and 2 columns

  x<-0
  FOR i=1 TO n
    FOR j=i+1 TO n-1
      x <- x+1
      c[x] <- (i,j)
    NEXT
  NEXT

multistart

  Let maxStart be the maximum number of starts allowed
  BEGIN
    WHILE k < maxStart
      Generate an initial random solution s0
      c <- possibleNeighbours(s0)
      s <- localSearch(c)
      IF s > s0
        s0 <-s
      ENDIF
    ENDWHILE
  STOP
```

2.2 GRASP y búsqueda local para BIPART

Falta de tiempo.

3 Propuestas de solución basadas en algoritmos poblacionales

A continuación se describe el enfoque elegido en este trabajo para resolver los problemas QAP y BIPART mediante algoritmos poblacionales. En los dos casos se utilizó el algoritmo genético simple con diferentes operadores de cruce, de mutación y de selección.

3.1 QAP y algoritmo genetico simple

Para resolver el problema QAP se utilizo un operador de cruce de correspondencia parcial (PMX). Con este operador una parte de la ristra representando a un padre se hace corresponder a otra parte, de igual tamaño, a la ristra del otro padre, intercambiándose la información restante. Por ejemplo, si consideramos las dos soluciones del problema QAP siguientes: (1 2 3 4 5 6 7 8) y (3 7 5 1 6 8 2 4) el operador PMX crea las ristras de los descendientes de manera siguiente:

- 1) Selección con probabilidad uniforme de dos puntos de corte en las ristras de los padres.
- 2) Se establece una correspondencia biunívoca entre los elementos que forman parte de las subristas comprendidas entre los puntos de corte.
- 3) La subrista del primer padre se copia en el segundo hijo, y de forma análoga, la subrista del segundo padre se copia en el primer hijo.
- 4) Cada i -ésimo descendiente se rellena copiando elementos del i -ésimo padre, teniendo en cuenta la correspondencias biunivocas.

Aplicando estos pasos obtenemos el descendiente 1 y el descendiente 2 para el ejemplo dado:.

1. (1 2 3 | 4 5 6 | 7 8) y (3 7 5 | 1 6 8 | 2 4)
2. 1= 4, 8=6, 6=5
3. (x x x | 1 6 8 | x x) y (x x x | 4 5 6 | x x)
4. (4 2 3 | 1 6 8 | 7 5) y (3 7 8 | 4 5 6 | 2 1)

Por otro lado, en este trabajo se ha utilizado el operador de mutación `mutShuffleIndexes` para resolver el problema QAP. El operador `mutShuffleIndexes` funciona de la manera siguiente:

1. Se fija un probabilidad p de mutación para la población.
2. Selección aleatoria de un valor x entre 0 y 1 para cada descendiente.
3. Si $x \leq p$ se permuten los elementos de una ristra de manera aleatoria.

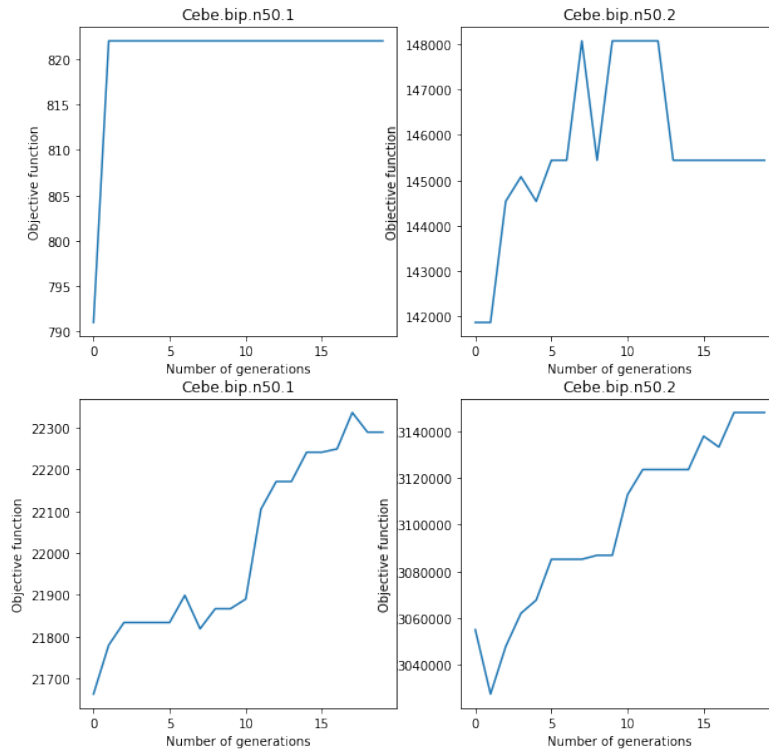
Finalmente, se ha utilizado la selección por torneo para evolucionar la población. En la selección por torneo se escogen al azar un número k (tamaño del torneo) de individuos de la población y se selecciona el mejor individuo de este grupo. El proceso se repite hasta que el número de individuos seleccionados esta igual al tamaño de la población original.

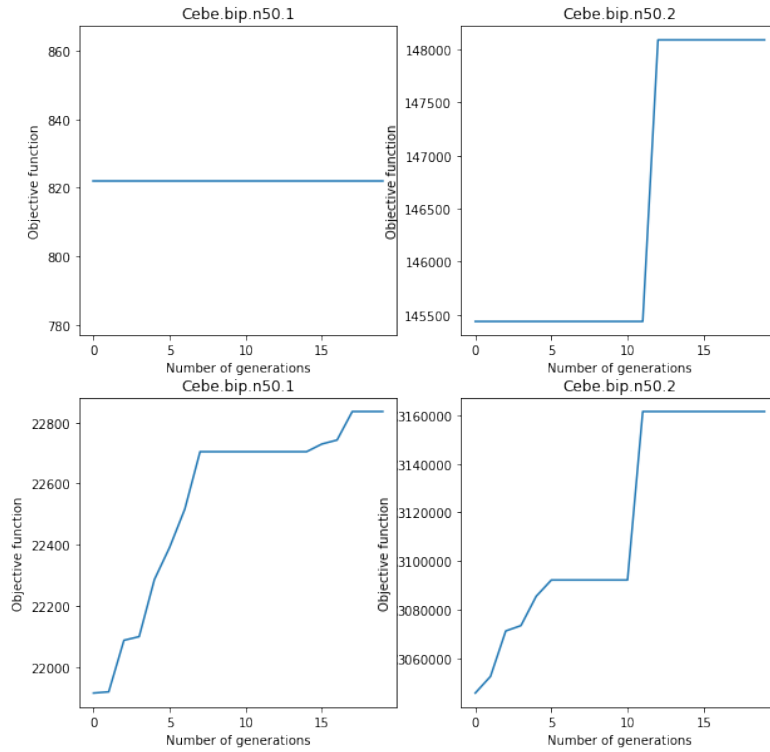
3.2 BIPART y algoritmo genetico simple

Falta de tiempo.

4 Experimentos

Creo haber observado que para el problema BIPART, el algoritmo genetico produce fluctuaciones para pequeñas valores del tamaño del torneo. A continuación se muestren los resultados obtenidos con el algoritmo genético simple para el problema BIPART con tamaño de torneo $k=2$ y $k=10$ y los operadores de cruce en dos puntos y de mutación mutFlipBit.





5 Referencias

- [1] Cormen TH., & Cormen, TH. (2001). Introduction to Algorithms. Cambridge, Mass: MIT Press.
- [2] Gendreau M. & Potvin JY. (2010). Handbook of Metaheuristics. Springer.
- [3] Feynman R. (1982), Simulating Physics with Computers, Int'l J. Theoretical Physics, 21(6-7):467-488.
- [4] Mertens S. (2002), Computacional Complexity for Physicists, Computing in Science & Engineering, 4(3):31-47.
- [5] Anstreicher KM. (2003), Recent Advances in the Solution of Quadratic Assignment Problems. Mathematical Programming Series B, 97:27-42.
- [6] Koopmans TC. & Beckmann MJ. (1957), Assignment Problems and the Location of Economic Activities. Econometrica, 25:53-76.
- [7] Hayes B. (2002), The Easiest Hard Problem, American Scientist, 90(2):113-

117.