Curso: Heurística – CM0439
Profesor: Juan Carlos Rivera Agudelo
Enero 24 de 2023
Métodos constructivos y aleatorizados

METODOS CONSTRUCTIVOS Y ALEATORIZADOS

Teniendo en cuenta el problema asignado para trabajar durante el semestre (Multitrip Vehicle Routing Problem – mtVRP –, ver Anexo 1), se deben implementar tres algoritmos: un algoritmo constructivo y dos algoritmos constructivos aleatorizados (construcción GRASP, construcciones con ruido, construcciones basadas en colonias de hormigas). Los algoritmos deben poderse parametrizar para ejecutarse con diferentes valores de cada parámetro según sea el caso (ver Anexo 4 – ejemplo de definición de parámetros).

El algoritmo debe ser desarrollado en Python o Matlab (otros lenguajes pueden ser utilizados sujetos a previa aprobación por parte del profesor).

Las pruebas del algoritmo deben ser realizadas con las instancias de prueba disponibles en las librerías para el problema seleccionado. Las instancias están disponibles en Interactiva Virtual con el formato descrito en el Anexo 2.

Cuando los algoritmos desarrollados sean basados en ideas de otros autores, i.e. método de los ahorros, NEH u otros, se debe indicar explícitamente en la descripción del método.

Formato de Entrega:

1. Realizar una presentación oral en la que se incluya:
    - Descripción de los algoritmos implementados.
    - Descripción de los resultados obtenidos, incluyendo:
        - Comparación con una cota inferior.
        - Comparaciones utilizando diferentes valores para cada parámetro.
        - Comparación entre los métodos implementados.
        - Tiempo de cómputo.
    - Conclusiones.
    - Se deben incluir todas las referencias bibliográficas que hayan servido de apoyo.
    - Se pueden incluir otras secciones en el informe.
2. Enviar por Interactiva Virtual los archivos correspondientes al código, los archivos de resultados y la presentación (con ayudas audiovisuales, formato libre).

La presentación deberá ser realizada en español o en inglés. La presentación se realizará por fuera de clase en horario acordado entre profesor y estudiante. Otros estudiantes son libres de asistir.

Se debe adjuntar los archivos correspondientes a los códigos de los algoritmos. Éstos deben poderse ejecutar sin necesidad de ingresar datos o modificaciones adicionales de forma manual. El algoritmo debe producir, para cada instancia disponible, un archivo de resultado con el nombre y formato descrito en el Anexo 3.

FECHA DE ENTREGA: La fecha límite de entrega es el miércoles 22 de marzo de 2023 (vía Buzón de Interactiva Virtual).

Curso: Heurística – CM0439
Profesor: Juan Carlos Rivera Agudelo
Enero 24 de 2023
Métodos constructivos y aleatorizados

UNIVERSIDAD EAFIT

ANEXO 1

DEFINICIÓN Y FORMULACIÓN DEL PROBLEMA

Vehicle routing problem refers to a family of combinatorial optimization problems where a set of demand nodes scattered over a geographical region by a fleet of capacitated homogeneous vehicles based on a depot. The multitrip variant, i.e. multitrip vehicle routing problem (mtVRP), is a version where each vehicle can perform several trips to fulfill node demands. Multitrip flexibility is especially needed when total demand exceeds total capacity.

The problem can be defined on an undirected complete graph G = (V; E). The node-set $V = \{0, \dots, n\}$ includes a depot-node 0 and a subset $V' = V\{0\}$ of n required sites, also called demand nodes. In the sequel, it is assumed that G is encoded as a symmetric directed graph, with travel times $w_{ij} = w_{ji}$ for each arc (i, j) in E. A fleet of R identical vehicles of capacity Q and autonomy T_H is based at the depot and each node $i \in V'$ has a known demand q_i. It is assumed without loss of generality that $\sum_{i \in V'} q_i \geq R \cdot Q, n \geq R$, and $q\_i \leq Q$ for all $i \in V'$.

The objective is to identify a set of trips such that each site is visited exactly once, and the sum of travel times of all trips is minimized. Note that a trip can be defined as a circuit, starting and ending at the depot, in which the total demand serviced does not exceed the vehicle capacity Q. Every trip must be assigned to exactly one vehicle and, if it is necessary, vehicles can perform more than one trip. Moreover, the trips assigned to each vehicle must be performed in a sequential ordered, that is, trip cannot be performed in parallel by the same vehicle. The set of successive trips performed by one vehicle is called a multitrip.

The described problem can be modeled by the 0-1 mixed integer linear program (MILP) in (1) to (9) based on four types of decision variables: The binary variables x_{ij} are equal to 1 if and only if arc (i, j) is traversed by a vehicle. The binary variables y_{ij} refers to replenishment arcs and take the value of 1 if and only if a trip finishes at node i and the following trip performed by the same vehicle starts at node j. Note that replenishment arc is used here as a trick to transform a multitrip to a single trip (without depot copies). The variables F_{ij} define the flow on each arc (i, j), that is, the vehicle load on each traversed arc. Finally, the variables t_i identify the arrival times at each node i.

$$\min f = \sum_{(i,j) \in A} w_{ij} \cdot x_{ij} + \sum_{(i,j) \in A} (w_{i0} + w_{0j}) \cdot y_{ij} \quad (1)$$

subject to,

$$\sum_{j \in V'} x_{0j} = R \tag{2}$$

$$\sum_{j \in V} x_{ij} + \sum_{j \in V'} y_{ij} = 1 \qquad \forall\, i \in V' \tag{3}$$

$$\sum_{i \in V} x_{ij} + \sum_{i \in V'} y_{ij} = 1 \qquad \forall\, j \in V' \tag{4}$$

$$F_{ij} \leq Q \cdot x_{ij} \qquad \forall\, i \in V', j \in V \tag{5}$$

$$F_{0j} \leq Q \cdot \left(x_{0j} + \sum_{i \in V'} y_{ij} \right) \qquad \forall\, j \in V' \tag{6}$$

$$\sum_{j \in N} (F_{ji} - F_{ij}) = q_i \qquad \forall\, i \in V' \tag{7}$$

$$t_j \geq w_{0j} \cdot x_{0j} \qquad \forall\, j \in V' \tag{8}$$

$$t_j \geq t_i + w_{ij} - M \cdot (1 - x_{ij}) \qquad \forall\, (i,j) \in E \tag{9}$$

$$t_j \geq t_i + (w_{i0} + w_{0j}) - M \cdot (1 - y_{ij}) \qquad \forall\, (i,j) \in E \tag{10}$$

$$t_j \leq T_H \qquad j \in V \tag{11}$$

$$x_{ij}, y_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in E \tag{12}$$

$$F_{ij}, t_{ij} \geq 0 \qquad \forall\, (i,j) \in E \tag{13}$$

The objective function in (1) aims to minimize the total traveled time. Note that in this type of models the traveled time, traveled distance and traveled cost are frequently assumed as proportional, so they are interchangeable in the objective function. Equation (2) guarantees that only available vehicles are used. Equations in (3) ensure that each node is visited exactly once, while constraints in (4) enforce to leave a node once it is visited. Constraints in (5) and (6) limit the maximum load when traversing any arc to the vehicle capacity. Equations in (7) force the model to ensure a flow conservation along the used arcs and prevent subtours. Expressions (8), (9) and (10) allows to compute arrival times at nodes. Constraints (11) limits vehicle autonomy or multitrips length to T_H. Constraints in (12) and (13) define the domain of decision variables. Note that variables F_{ij} could be integer if q_i is integer, but given the structure of the formulation, a continuous domain will lead to integer values.

The problem can also be formulated by different models; Cattaruzza et al (2016) present a survey with four different formulations for the mtVRP.

Cattaruzza, Diego; Absi, Nabil and Feillet, Dominique (2016). Vehicle routing problems with multiple trips. 4OR, 14: 223-259.

Curso: Heurística – CM0439
Profesor: Juan Carlos Rivera Agudelo
Enero 24 de 2023
Métodos constructivos y aleatorizados

ANEXO 2

FORMATO DE DATOS DE LAS INSTANCIAS

En el archivo "mtVRP Instances.zip" (disponible en EAFIT Interactiva) se encuentran 12 instancias con datos para el mtVRP, donde el número de nodos varía entre 50 y 200. El formato de cada instancia es el siguiente:

La primera fila indica el número de nodos de demanda (sin incluir el depósito) n, el número de vehículos R, la capacidad de cada vehículo Q y la máxima distancia que puede recorrer cada vehículo T_H.

Las siguientes $n + 1$ filas indican, para cada nodo (incluyendo el depósito), el índice del nodo correspondiente, las coordenadas (x, y) y la demanda. La demanda del depósito se asume igual a cero para cada instancia del problema.

La distancia entre cada par de nodos se calcula utilizando la distancia euclídea.

A continuación se presenta un ejemplo con $n = 5$:

5	2	8	35
0	0	0	0
1	1	8	3
2	5	7	6
3	2	10	2
4	3	4	7
5	10	2	1

Curso: Heurística – CM0439
Profesor: Juan Carlos Rivera Agudelo
Enero 24 de 2023
Métodos constructivos y aleatorizados

ANEXO 3

FORMATO DE ARCHIVO DE RESULTADOS

Los resultados obtenidos se deben registrar en un archivo de Excel en el cual cada hoja contenga los resultados de cada instancia. El nombre del archivo debe ser mtVRP_<nombre_estudiante>_<método>.xlsx, donde <nombre_estudiante> debe ser reemplazado por el nombre (nombre, o apellido, o iniciales) de cada estudiante, y <método> debe ser reemplazado por un nombre indicador del método utilizado para hallar los resultados (constructivo, GRASP, ACO, LS, VNS, GA,…). Cada hoja del archivo de Excel debe ser nombrada con el nombre de la instancia respectiva (usar los mismos nombres utilizados en el archivo de datos).

Los resultados deben tener el siguiente formato: Cada solución se debe componer de un conjunto de multitrips. Así cada hoja debe contener R filas, una por cada vehículo, con la siguiente información: número total de nodos en el multitrip (incluyendo todas las réplicas necesarias del depósito), la secuencia ordenada de nodos visitados (incluyendo todas las réplicas necesarias del depósito), la distancia total recorrida por el multitrip y un indicador con valor cero en caso de que el multitrip satisfaga la restricción de longitud o uno en caso contrario. La última fila ($R + 1$) debe contener el valor de la función objetivo (la suma de las distancias totales recorridas) y el tiempo de cómputo.

Todos los valores no enteros deben ser redondeados a dos cifras decimales.

El siguiente es un ejemplo de una solución factible para la instancia de ejemplo presentada en el Anexo 2.

0	4	5	0	22.48	0		
0	1	3	0	2	0	37.70	1
60.18	0.01	1					

En la solución anterior $n = 5$. La primera ruta inicia en el nodo 0 y visita los nodos 4 y 5, y regresa al depósito, con una demanda total de 8 unidades y un tiempo total de 22.48 unidades. La segunda ruta está compuesta de dos multrips: el primer trip visita los nodos 1 y 3 con una demanda total de 5 unidades, y el segundo visita el nodo 2 con una demanda de 6 unidades. La distancia recorrida por el segundo multitrip es 37.7, excediendo el tiempo límite de 35 unidades. El tiempo total recorrido es de 22.48+37.70=60.18 unidades y el tiempo de cómputo es 0.01 segundos.

UNIVERSIDAD EAFIT

ANEXO 4

EJEMPLO DE DEFINICIÓN DE PARÁMETROS

La siguiente figura muestra un algoritmo en lenguaje de programación Python donde luego de cargar librerías y funciones se definen los valores de los 4 parámetros a utilizar: nsol, alpha, K y r. En los algoritmos implementados los parámetros deben tener valores constantes; no deben ser leídos de algún archivo, ni solicitados al usuario en cada ejecución.

```
10 import xlwt
11 from xlwt import Workbook
12 
13 from Constructive import Constructive
14 
15 from GRASP1 import GRASP1
16 from GRASP2 import GRASP2
17 from Noise import Noise
18 
19 
20 # Data reading
21 from Read import read_fsspsc
22 
23 
24 wb = Workbook()
25 sheet1 = wb.add_sheet('Randomized')
26 
27 nsol=100   # Number of solutions
28 alpha=0.05 # GRASP parameter
29 K=5        # Maximum RCL size
30 r=5        # Noise
31 
32 for id in range(1,21):
33     print(id)
34     n,m,L,dur = read_fsspsc(id)
35 
36     Z0,S0,I0,F0,Fi0,t0=Constructive(n,m,dur,L,id)
37     print("C:\t",Z0,"\t",t0)
38 
39     Z1,S1,I1,F1,Fi1,t1=GRASP1(n,m,dur,L,id,alpha,nsol)
40     print("GRASP1:\t",Z1,"\t",t1)
41 
42     Z2,S2,I2,F2,Fi2,t2=GRASP2(n,m,dur,L,id,K,nsol)
43     print("GRASP2:\t",Z2,"\t",t2)
44 
45     Z3,S3,I3,F3,Fi3,t3=Noise(n,m,dur,L,id,r,20)
       print("Noise:\t",Z3,"\t",t3)
```