UNIVERSIDAD SIMÓN BOLÍVAR

Departamento de Computación y Tecnología de la Información

CI-5651 - Diseño de Algoritmos I

Profesor: Guillermo Palma

PROYECTO 1

Elaborado Por:

Salvador Di Renzo 10-10296
Stefani Castellanos 11-11394

Sartenejas, Febrero 2017

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Para resolver el problema *Prize-Collecting Rural Postman Problem* se construyó un algoritmo que utiliza la estrategia *Greedy*, es decir, cada vez que se agrega una arista a la solución no es posible removerla.

El diseño de la solución está basada en el teorema del enunciado: un ciclo euleriano que pasa por el vértice depósito es una solución factible al problema PRPP. Para ello se pregunta si el conjunto de aristas cuyo costo al cruzarlas dos veces es positivo constituyen un ciclo euleriano que pasa por el depósito; de ser así se toma esa como la solución, de lo contrario se unen las componentes conexas de ese conjunto utilizando una Kruskall (*Minimum Spanning Tree*) y finalmente se convierte en un ciclo euleriano.

Sea G(V, E, d, c , b) el grafo original cuyo conjunto de vértices es V y conjunto de lados es E, el depósito es d, la función de costo es c y la de beneficio b. También se define la clasificación de aristas de la siguiente manera:

$$P = \{e \in E \mid c(e) - b(e) < 0\} \qquad R = \{e \in E \mid c(e) - 2 * b(e) \geq 0\}$$

$$Q = \{e \in E \setminus R \mid c(e) - b(e) \geq 0\}$$

```
-----------------------------------------------------------------------------
Gr <- Grafo conformado por las aristas R del G(V,E)
Si Gr es euleriano
    retornar camino que pasa por dichas aristas

sino
    Gmst <- El grafo (árbol) cobertor mínimo que conecta las componentes
            conexas de Gr
    Hacer que Gmst sea euleriano (todos los vertices de grado par)
    retornar camino que pasa por dichas aristas

-----------------------------------------------------------------------------
```

En cuanto a la implementación, para representar el Grafo se utilizó la estructura lista de aristas con su costo y beneficio. Para recorrer con mayor facilidad el camino que se debe devolver se utilizó la representación de lista de adyacencias agregando cada para de aristas que se encontraron que pertenecen al camino. Adicionalmente fue implementada la estructura “Conjuntos disjuntos” (*Union-Find*) con compresión de caminos que permite mejorar el tiempo amortizado de la unión de árboles que necesita el algoritmo Kruskall.

Orden del algoritmo

Mejor caso: $O(H) = clasificar + esEuleriano + camino$
Peor caso(Amortizado): $O(H) = clasificar + esEuleriano + mst + euleriano + camino$

Donde:

clasificar: clasificar las aristas según los conjuntos *P*, *R* y *Q*. *O(E*)
esEuleriano: determinar si Gr(Vr, R) es euleriano.
mst: algoritmo de Kruskall sobre Gr . *O(R log Vr)*
euleriano: convertir el grafo resultante del MST en un ciclo euleriano. $O((V+E)^2)$.
camino: devolver el camino. $O(V * E)$

RESULTADO EXPERIMENTAL Y ANÁLISIS

A continuación se presentan los resultados obtenidos comparados con la solución propuesta por Meza usando la heurística de corte de planos.

ALBAIDA

Instancia	Vo	%dHeurPlanos	%dHeur	tHeur (seg)
ALBAIDAA	6266	0	-100	0
ALBAIDAB	4372	0	-100	0

CHRISTOFIDES

Instancia	Vo	%dHeurPlanos	%dHeur	tHeur (seg)
P1	3	0	500	0
P2	66	0	0	0
P3	56	0	0	0
P4	45	0	0	0
P5	35	0	-200	0

P6	60	0	0	0
P7	89	0	0	0
P8	90	0	0	0
P9	46	0	-100	0
P10	41	0	0	0
P11	9	0	100	0
P12	10	0	0	0
P13	5	0	0	0
P14	128	0	-100	0
P15	43	0	-800	0
P16	113	0	0	0
P17	42	0	0	0
P18	21	0	-500	0
P19	90	0	0	0
P20	246	0	0	0
P21	258	0	0	0
P22	474	0	0	0
P23	360	0	0	0
P24	237	0	-100	0

DEGREE

Instancia	Vo	%dHeurPlanos	%dHeur	tHeur (seg)
D0	109	0	-300	0

D1	115	0	-200	0
D2	274	0	0	0
D3	172	0	300	0
D4	210	0	-200	0
D5	313	0	0	0
D6	166	0	-400	0
D7	260	0	200	0
D8	457	0	0	0
D9	160	0	500	0
D10	0	0	0	0
D11	398	0	0	0
D12	280	0	100	0
D13	717	0	100	0
D14	810	0	-100	0
D15	702	0	-100	0
D16	980	0	0	0
D17	1115	0	0	0
D18	515	0	-100	0
D19	509	0	0	0
D20	457	0	0	0
D21	1000	0	-100	0
D22	989	0	-100	0
D23	968	0	100	0

D24	1463	0	-100	0
D25	1317	0	0	0
D26	1625	0	0	0
D27	549	0	-300	0
D28	815	0	-100	0
D29	55	-800	-500	0
D30	1378	0	-100	0
D31	1503	0	0	0
D32	1066	0	0	0
D33	2074	0	0	0
D34	1806	0	0	0
D35	1901	0	0	0

GRID

Instancia	Vo	%dHeurPlanos	%dHeur	tHeur (seg)
G0	0	0	0	0
G1	0	0	0	0
G2	0	0	0	0
G3	2	0	-100	0
G4	0	0	0	0
G5	4	0	400	0
G6	9	0	100	0
G7	1	0	200	0

G8	4	0	0	0
G9	2	0	-800	0
G10	0	0	0	1
G11	4	0	900	0
G12	15	0	100	0
G13	11	0	0	0
G14	14	0	-100	0
G15	26	0	0	0
G16	20	0	-100	0
G17	24	0	0	0
G18	8	0	700	0
G19	6	0	0	0
G20	9	0	0	0
G21	32	0	-100	0
G22	32	0	0	0
G23	33	0	-100	0
G24	57	0	0	0
G25	46	0	-100	0
G26	57	0	0	0
G27	14	0	-300	0
G28	23	0	-100	0
G29	14	0	-300	0
G30	50	0	100	0

G31	54	0	-100	0
G32	57	0	-100	0
G33	92	0	0	0
G34	86	0	-100	0
G35	88	0	0	0

RANDOM

Instancia	Vo	%dHeurPlanos	%dHeur	tHeur (seg)
R0	1742	0	-100	0
R1	4253	0	-800	0
R2	5638	0	-400	0
R3	18453	0	0	0
R4	17316	0	200	0
R5	398	0	2100	0
R6	12478	0	-100	0
R7	9405	0	-400	0
R8	14847	0	200	0
R9	17405	0	100	0
R10	7125	0	500	0
R11	1493	0	-1700	0
R12	32453	0	100	0
R13	30732	0	0	0
R14	27791	0	-200	0

R15	10533	0	100	0
R16	4276	0	-700	0
R17	28462	0	200	0
R18	26873	0	0	0

OBSERVACIONES

Durante el desarrollo del proyecto, desafortunadamente, no se logró plasmar las ideas planteadas en la sección de descripción del algoritmo por razones de diversa índole, es por este motivo que los resultados en las tablas no permiten realizar conclusiones sobre el tiempo que toma realmente toma retornar una respuesta para el problema de *Prize-Collecting Rural Postman Problem* con la heurística presentada.