Inteligencia Computacional (TC–3023)
Problema de ruteo de vehículos

1. Introducción

1.1. Problema del vendedor viajero

En el problema del vendedor viajero, *traveling salesperson person* (TSP), se tienen N ciudades, y se desea la ruta que parte de una ciudad, visita todas las ciudades, regresa a la ciudad inicial, y que tiene un mínimo costo. En el TSP euclidiano, el costo es la distancia euclidiana entre las ciudades, y por lo tanto, el costo de ir de la ciudad A a la ciudad B es el mismo que el costo de ir de la ciudad B a la ciudad A. En TSP el espacio de búsqueda es de cardinalidad $N!$ En la figura 1 se muestra un pequeño ejemplo del TSP.

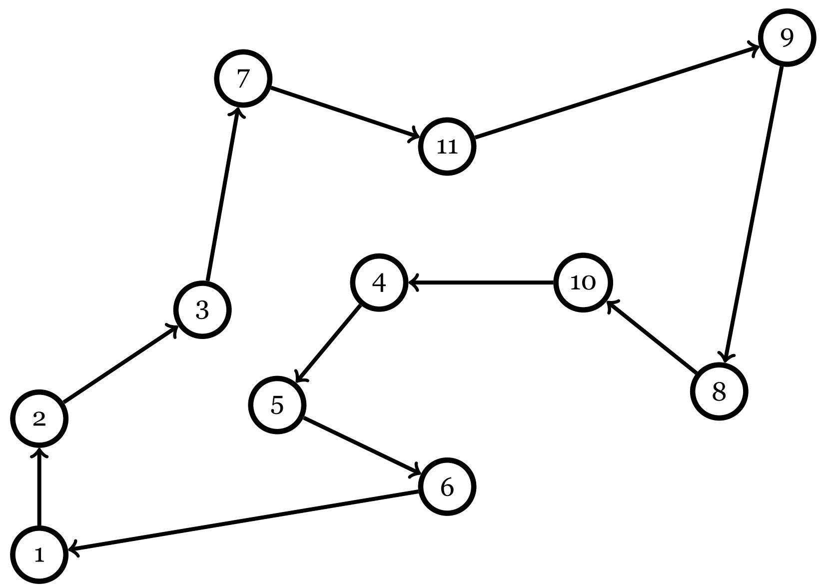

Figura 1: Ejemplo de una solución posible a un problema de vendedor viajero euclidiano de 11 ciudades. Los arcos representan los posibles tramos de calles entre las posiciones geográficas de los clientes y el almacén central.

1.2. Problema de ruteo de vehículos

En el problema de ruteo de vehículos, *vehicle routing problem* (VRP), se tienen N clientes que se deben visitar, y se tiene una flotilla de vehículos. Se tienen almacenes de donde parten y a donde regresan los vehículos, y se tiene una red de caminos donde pueden viajar estos camiones. Se desea encontrar las rutas que deben seguir los vehículos para que visitar a todos los clientes de manera que se minimice alguna definición de costo. Usualmente, el costo está relacionado con la distancia recorrida por los vehículos.

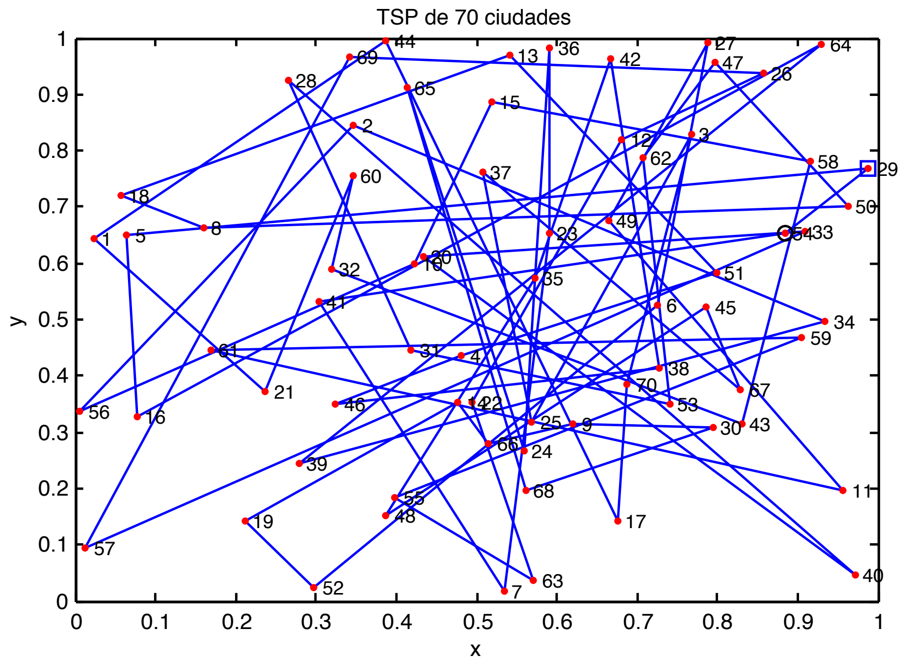

Figura 2: Problema de vendedor viajero. Estado inicial para recocido simulado.

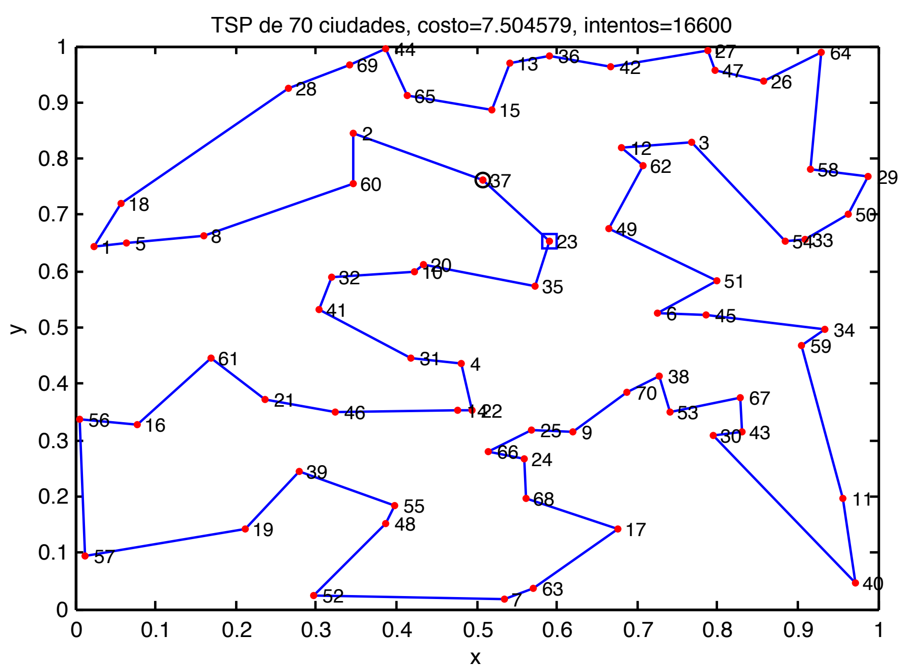

Figura 3: Problema de vendedor viajero. Estado final que se obtiene con recocido simulado después de 16600 evaluaciones de la función objetivo.

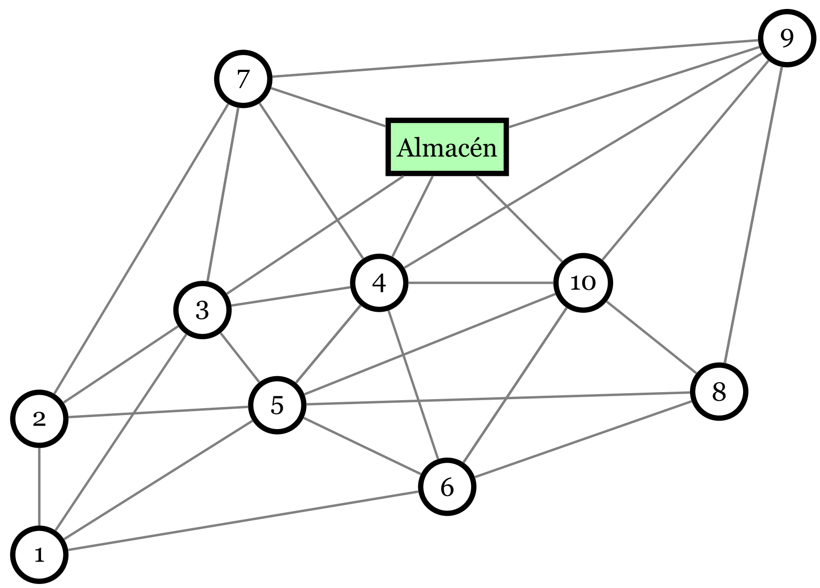

Figura 4: Ejemplo de un problema de ruteo de vehículos. Los arcos del grafo representan tramos de la red de caminos disponibles.

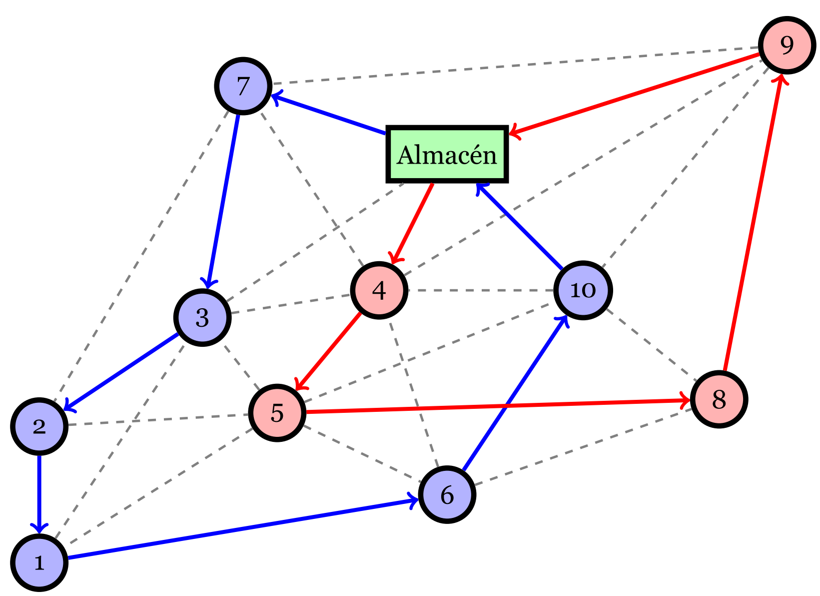

Figura 5: Ejemplo de una solución posible al problema de ruteo de vehículos de la figura 4. Se tienen dos rutas que parten de un almacén. Las líneas punteadas son tramos de la red de calles no utilizados.

1.2.1. Variantes de VRP

Existen muchas variantes del problema de ruteo de vehículo. A continuación se presentan algunas de las más usuales.

- (CVRP) Vehículos con capacidades máximas
  Los clientes tienen demandas de lo que se les debe entregar, y los vehículos tienen una capacidad máxima que pueden transportar. La solución no puede violar la capacidad máxima de los vehículos. La flotilla de vehículos puede ser uniforme o no uniforme, es decir, se pueden tener vehículo con diferentes capacidades.

- (MDVRP) Múltiples almacenes
  Se tienen múltiples almacenes desde donde se pueden atender las demandas de los clientes.

- (VRPTW) Clientes con ventanas de tiempo
  Los clientes establecen ventanas de tiempo en las cuales pueden recibir a los vehículos. Se debe tomar en cuenta el tiempo de recorrido, y el tiempo de servicio a cada cliente.

- Red de caminos no uniforme y flotilla no uniforme
  Se tiene una red de caminos de diferentes capacidades, y vehículos con diferentes características que no puede todos circular sobre todos los tramos de la red.

- Recolección y entrega
  En esta variante del problema, los vehículos deben entregar y recoger carga de los clientes. Es posible que también se tenga que tomar en cuenta el acomodo de la carga dentro de cada camión.

- Problemas estocásticos y no estacionarios
  En esta variante es posible que los tiempos de recorrido cambien dependiendo de la hora del día, o que sean estocásticos, es decir que el tiempo de recorrido de un punto pueda variar con una cierta distribución de probablidad.

En la página http://neo.lcc.uma.es/vrp/vrp-flavors/ se presetan descripciones de éstas y otras variantes del VRP.

1.2.2. Problemas de Solomon para CVRPTW

Solomon (1987) presentó 56 instancias del problema CVRPTW.[1] Sus base de datos de problema ha sido utilizada extensamente para probar algoritmos para el problema de VRPTW. En los problema de Solomon se tiene un solo almacén. Los vehículos tienen una capacidad máxima igual. Los clientes tienen demandas conocidas. Adicionalmente, los clientes tienen ventanas de tiempo definidas como hora más temprana y hora más tarde en la que pueden ser atendidos. Se desea minimizar, primero, el número de vehículos utilizados y el costo de operarlos, y después, el costo es la distancia total recorrida por los vehículos; es decir, una solución que utilice menos vehículos es preferible a cualquier otra, sin importar su distancia, que utilice más vehículos.

Para el cliente i se tienen los siguientes datos:

- s_i unidades de tiempo para entregar. (*Service time*)

- d_i demanda.

[1] Se pueden obtener de http://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/

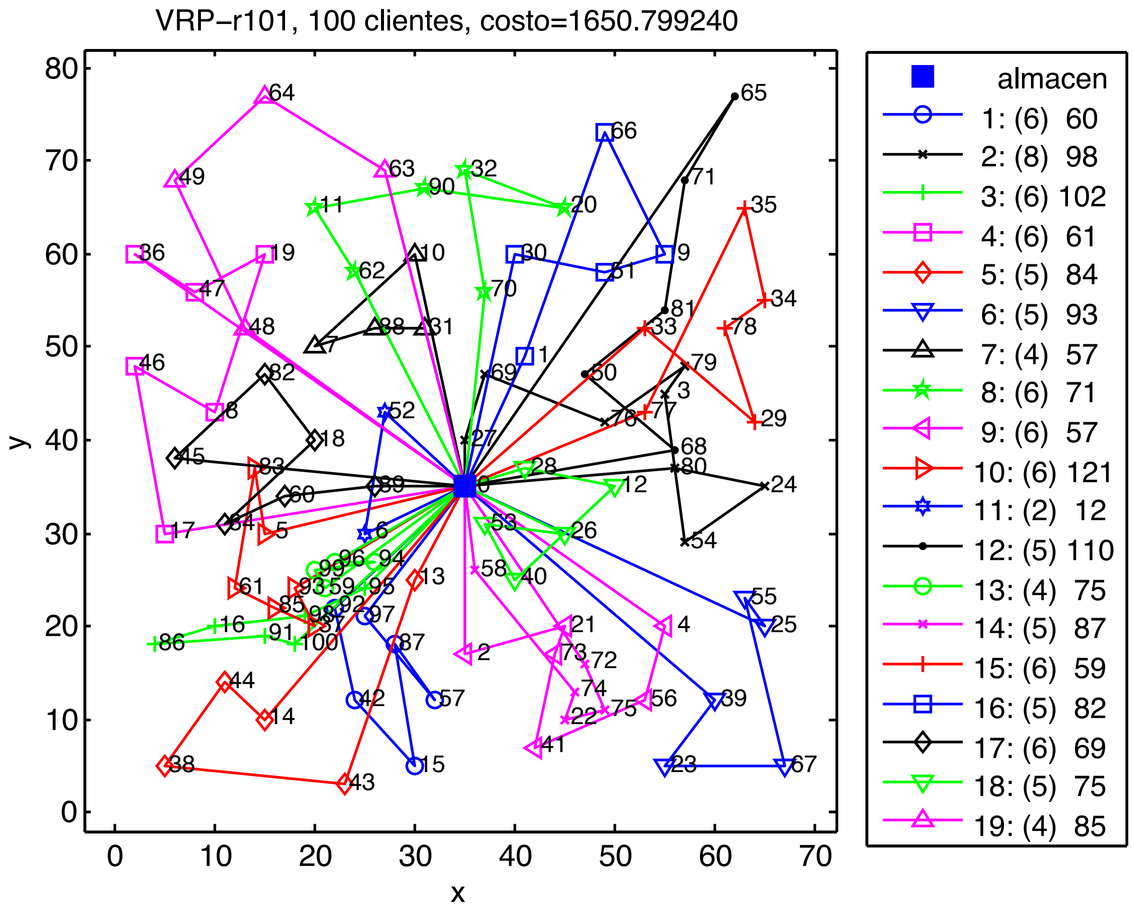

Figura 6: Mejor solución conocida para el problema R101 de Solomon

- $[e_i, l_i]$ ventana de tiempo para el cliente i, donde e_i es el tiempo más temprano de atención (*earliest*) y l_i es el tiempo más tarde de atención (*latest*). (También se les llama *ready time* y *due date*.)
- b_i tiempo en el que se inicia el servicio para el cliente i. (Esto es parte de la solución.)
- $b_j = \max\{e_j, b_i + s_i + t_{ij}\}$ Porque el cliente no va a recibir al camión antes de su ventana de tiempo.

Además, se deben tener los siguientes datos del problema:

- t_{ij} tiempo para viajar del cliente i al cliente j
- d_{ij} distancia directa entre cliente i y cliente j

Es posible que en lugar de la distancia y el tiempo entre clientes, se den las coordenadas (x_i, y_i) de la colocación geográfica del cliente i y la distancia se calcule como la distancia euclidiana. En este caso, el tiempo es la distancia multiplicada por una velocidad fija de 1.

En la figura 6 se muestra la mejor solución conocida para el problema R101 de Solomon.

Costo total: 1650.799240
Tiempo máximo: 219.055385
Capacidad máxima: 200

no.	clientes	terminación	capacidad	ruta
1:	(6)	160.20	60	92 42 15 87 57 97
2:	(8)	213.10	98	27 69 76 79 3 54 24 80
3:	(6)	219.04	102	95 98 16 86 91 100
4:	(6)	197.41	61	36 47 19 8 46 17
5:	(5)	186.27	84	14 44 38 43 13
6:	(5)	215.54	93	39 23 67 55 25
7:	(4)	159.50	57	31 88 7 10
8:	(6)	213.10	71	62 11 90 20 32 70
9:	(6)	184.00	57	2 21 73 41 56 4
10:	(6)	218.25	121	5 83 61 85 37 93
11:	(2)	120.18	12	52 6
12:	(5)	177.42	110	65 71 81 50 68
13:	(4)	160.26	75	59 99 94 96
14:	(5)	219.06	87	72 75 22 74 58
15:	(6)	208.70	59	33 29 78 34 35 77
16:	(5)	191.39	82	30 51 9 66 1
17:	(6)	200.06	69	45 82 18 84 60 89
18:	(5)	153.18	75	28 12 40 53 26
19:	(4)	202.80	85	63 64 49 48

Tabla 1: Mejor solución conocida para el problema R101 de Solomon

i	d_i	e_i	b_i	l_i	ruta	i	d_i	e_i	b_i	l_i	ruta
1:	10	161	166	171	16	51:	10	88	90	98	16
2:	7	50	50	60	9	52:	9	52	52	62	11
3:	13	116	116	126	2	53:	14	95	104	105	18
4:	19	149	149	159	9	54:	18	140	142	150	2
5:	26	34	34	44	10	55:	2	136	136	146	6
6:	3	99	99	109	11	56:	6	130	130	140	9
7:	5	81	90	91	7	57:	7	101	110	111	1
8:	9	95	104	105	4	58:	18	200	200	210	14
9:	16	97	107	107	16	59:	28	18	18	28	13
10:	16	124	124	134	7	60:	3	162	162	172	17
11:	12	67	76	77	8	61:	13	76	76	86	10
12:	19	63	63	73	18	62:	19	58	58	68	8
13:	23	159	165	169	5	63:	10	34	35	44	19
14:	20	32	32	42	5	64:	9	73	73	83	19
15:	8	61	61	71	1	65:	20	51	51	61	12
16:	19	75	77	85	3	66:	25	127	131	137	16
17:	2	157	157	167	4	67:	25	83	90	93	6
18:	12	87	87	97	17	68:	36	142	146	152	12
19:	17	76	77	86	4	69:	6	50	54	60	2
20:	9	126	126	136	8	70:	5	182	182	192	8
21:	11	62	70	72	9	71:	15	77	77	87	12
22:	18	97	97	107	14	72:	25	35	35	45	14
23:	29	68	68	78	6	73:	9	78	84	88	9
24:	3	153	162	163	2	74:	8	149	149	159	14
25:	6	172	172	182	6	75:	18	69	69	79	14
26:	17	132	132	142	18	76:	13	73	77	83	2
27:	16	37	37	47	2	77:	14	179	179	189	15
28:	16	39	39	49	18	78:	3	96	96	106	15
29:	9	63	63	73	15	79:	23	92	97	102	2
30:	21	71	71	81	16	80:	6	182	182	192	2
31:	27	50	50	60	7	81:	26	94	101	104	12
32:	23	141	147	151	8	82:	16	55	55	65	17
33:	11	37	37	47	15	83:	11	44	51	54	10
34:	14	117	117	127	15	84:	7	101	110	111	17
35:	8	143	143	153	15	85:	41	91	91	101	10
36:	5	41	41	51	4	86:	35	94	94	104	3
37:	8	134	134	144	10	87:	26	93	93	103	1
38:	16	83	90	93	5	88:	9	74	74	84	7
39:	31	44	44	54	6	89:	15	176	181	186	17
40:	9	85	87	95	18	90:	3	95	97	105	8
41:	5	97	104	107	9	91:	1	160	160	170	3
42:	5	31	39	41	1	92:	2	18	18	28	1
43:	7	132	132	142	5	93:	22	188	188	198	10
44:	18	69	69	79	5	94:	27	100	100	110	13
45:	16	32	32	42	17	95:	20	39	39	49	3
46:	1	117	124	127	4	96:	11	135	135	145	13
47:	27	51	59	61	4	97:	12	133	133	143	1
48:	36	165	165	175	19	98:	10	58	58	68	3
49:	30	108	108	118	19	99:	9	83	83	93	13
50:	13	124	124	134	12	100:	17	185	185	195	3

Tabla 2: Mejor solución conocida para el problema R101 de Solomon.

$$R_1: \quad 1 \;\; 2 \;\; 3 \;\; \boxed{4 \;\; 5 \;\; 6 \;\; 7 \;\; 8} \;\; 9 \;\; 10$$

$$\Downarrow$$

$$R_1': \quad 1 \;\; 2 \;\; 3 \;\; \boxed{8 \;\; 6 \;\; 7 \;\; 5 \;\; 4} \;\; 9 \;\; 10$$

Figura 7: 2-Opt (Una ruta). Se escogen dos posiciones aleatorias en una ruta, y se invierte la subruta entre esas dos posiciones.

$$R_1: \quad 1 \;\; 2 \;\; 3 \;\; \boxed{4 \;\; 5 \;\; 6} \;\; 7 \;\; 8 \;\; 9 \;\; 10$$

$$\Downarrow$$

$$R_1': \quad 1 \;\; 2 \;\; 3 \;\; 7 \;\; 8 \;\; \boxed{4 \;\; 5 \;\; 6} \;\; 9 \;\; 10$$

Figura 8: Or-Opt (Una ruta). Se escoge aleatoriamente una subruta y una posición a lo largo de la ruta. La subruta se coloca después de la posición escogida.

1.3. Funciones de vecindad para VRPTW

Las funciones para generar vecinos para el VRP se clasifican en aquellas que modifican a una sola ruta y las que modifican a dos (o más) rutas. Comúnmente se utilizan las siguientes para el VRPTW (Bräysy y Gendreau, 2005a): Funciones para una ruta:

- 2-opt
- Or-opt
- Intercambio intra-ruta

Funciones para dos rutas:

- 2-opt∗
- Operador de relocación (*Relocate operator*)
- Operador de intercambio (*Exchange operator*)
- Intercambio cruzado (*Cross exchange operator*)

Las figuras 7–13 dan ejemplos de estos operadores.

$$
\begin{array}{ll}
R_1: & 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8\ \ 9\ \ 10 \\
 & \Downarrow \\
R'_1: & 1\ \ 8\ \ 9\ \ 5\ \ 6\ \ 7\ \ 2\ \ 3\ \ 4\ \ 10
\end{array}
$$

Figura 9: Intercambio intra-ruta (Una ruta). Se escogen aleatoriamente dos subrutas y se intercambia su posición.

$$
\begin{array}{ll}
R_1: & 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8 \\
R_2: & 9\ \ 10\ \ 11\ \ 12\ \ 13\ \ 14\ \ 15\ \ 16 \\
 & \Downarrow \\
R'_1: & 1\ \ 2\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8 \\
R'_2: & 9\ \ 10\ \ 11\ \ 12\ \ 13\ \ 14\ \ 15\ \ 3\ \ 16
\end{array}
$$

Figura 10: Relocate Operator (Dos rutas). Para cada ruta se escoge aleatoriamente una posición. Se coloca el cliente de la posición en la primera ruta después de la posición de la segunda ruta.

$$
\begin{array}{ll}
R_1: & 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8 \\
R_2: & 9\ \ 10\ \ 11\ \ 12\ \ 13\ \ 14\ \ 15\ \ 16 \\
 & \Downarrow \\
R'_1: & 1\ \ 2\ \ 15\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8 \\
R'_2: & 9\ \ 10\ \ 11\ \ 12\ \ 13\ \ 14\ \ 3\ \ 16
\end{array}
$$

Figura 11: Exchange Operator (Dos rutas) Para cada ruta se escoge aleatoriamente una posición. Se intercambian de ruta los clientes en las posiciones.

$$
\begin{array}{ll}
R_1: & 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8 \\
R_2: & 9\ \ 10\ \ 11\ \ 12\ \ 13\ \ 14\ \ 15\ \ 16 \\
 & \Downarrow \\
R'_1: & 1\ \ 2\ \ 3\ \ 13\ \ 14\ \ 15\ \ 16 \\
R'_2: & 9\ \ 10\ \ 11\ \ 12\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8
\end{array}
$$

Figura 12: 2-Opt∗ (Dos rutas) Se escoge aleatoriamente una arista en cada ruta. Se intercambian las subrutas después de las aristas.

R_1 : 1 2 3 4 5 6 7 8

R_2 : 9 10 11 12 13 14 15 16

$\Downarrow$

R_1' : 1 12 13 14 15 5 6 7 8

R_2' : 9 10 11 2 3 4 16

Figura 13: Cross exchange (Dos rutas). Para cada ruta se escoge aleatoriamente una subruta. Se intercambian las subrutas.

2. Definición del problema

Se desea aplicar recocido simulado para resolver instancias del conjunto de problemas Solomon (1987).

2.1. Caso 1

Utilice el problema R101 de Solomon. Resuelva para problemas sin ventanas de tiempo, y con ventanas de tiempo.

2.2. Caso 2

Utilice el problema R102 de Solomon. Resuelva para problemas sin ventanas de tiempo, y con ventanas de tiempo.

3. Reporte

Utilice recocido simulado para resolver los casos que se explican arriba. Afine los parámetros del algoritmo para que se comporte de la mejor forma posible. Entregue un reporte que incluya la siguiente información:

- Descripción del problema y de la función objetivo.
- Descripción del estado y de la función de vecindad implementada.
- Valores de todos los parámetros del algoritmo.
- Curvas del mejor encontrado para los dos casos.
- Mejor solución encontrada,(rutas, costo total, etc.) para cada caso.
- Código de la función objetivo y función de vecindad.

Referencias

Bräysy, O., y Gendreau, M. (2005a). Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, *39*(1), 104–108. Descargado de http://dx.doi.org/10.1287/trsc.1030.0056

Solomon, M. M. (1987). Algorithms for the vehicle routing problem with time windows. *Operations Research*, *35*(2), 254–265.