Trabalho de Otimização Combinatória (2016-2)

Entregas

- ??/??/????: relatório com formulação matemática e testes com o GLPK.
- ? (CVRP), ? (SDVRP), ? (SOP): relatório dos resultados da mateheurística (incluir também os resultados do GLPK).

Relatório

Para cada combinação problema vs. metaheurística, apresentar um relatório com aproximadamente 6 páginas contendo no mínimo, as seguintes informações:

- Introdução;
- Descrição clara do problema e formulação matemática do problema;
- Descrição com detalhes do algoritmo proposto:
  - Representação do problema;
  - Principais estruturas de dados;
  - Heurística construtiva;
  - Vizinhança e a estratégia de escolha de vizinhos;
  - Parâmetros do método (combinações testadas e valores usados nos experimentos);
  - Critério de parada;
- Tabela de resultados com no mínimo as seguintes colunas para cada instância:
  - Valor da solução inicial heurística;
  - Valor da melhor solução encontrada pelo seu algoritmo (S);
  - Tempo de execução (em segundos);
  - Solução do GLPK (reportar a melhor solução encontrada no tempo disponível para execução do GLPK, mesmo quando não ótima);
  - Tempo do GLPK (com limite de tempo de 1h [parâmetro --tmlim 3600"do GLPK] por instância - ou mais tempo);
  - Desvio percentual ($100\frac{MC-S}{MC}$) da melhor solução conhecida MC.
  - Análise dos resultados;
  - Conclusões;
  - Bibliografia pesquisada.

Os resultados das metaheurísticas deve ser uma média de no mínimo 10 rodadas. Além do relatório, os alunos devem entregar (no moodle) um zip contendo os códigos e relatório.

Implementação

- Todas as implementações devem aceitar uma instância no formato do problema na entrada padrão (stdin) e imprimir a melhor solução encontrada, bem como o tempo utilizado na saída padrão (stdout);
- Os principais parâmetros do método devem ser definíveis pela linha de comando;
- Critérios básicos de engenharia de software: documentação, legibilidade, etc;

- Critérios como qualidade das soluções encontradas e eficiência das implementações serão considerados na avaliação (ex: quando fizer uma modificação na solução, recalcular a diferença com o vizinho e não a solução inteira novamente);
- Comentários no código ajudam na correção;
- Vizinhança: uma boa vizinhança é aquela que permite alcançar qualquer solução;
- Vizinhos com mesmo valor de solução: Utilizar escolhas aleatórias como critério de desempate podem trazer bons resutados.

Dicas de referência para metaheurísticas:

1. VNS: A Tutorial on Variable Neighborhood Search, by Pierre Hansen (GERAD and HEC Montreal) and Nenad Mladenovic (GERAD and Mathematical Institute, SANU, Belgrade), 2003. `http://www.dei.unipd.it/~fisch/ricop/VNS_tutorial.pdf`;
2. GRASP: `http://www.research.att.com/export/sites/att_labs/techdocs/TD_100315.pdf`;
3. GA: A genetic algorithm tutorial, by D. Whitley, Statistics and computing 4 (2), 65-85;
4. Tabu Search: Tabu Search: A Tutorial, by Fred Glover (1990), Interfaces;
5. SA: Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning, by D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon, Operations research 37 (6), 865-892, 1989;

1 Capacitated Vehicle Routing Problem (CVRP)

O Problema de Roteamento de Veículos ou *Vehicle Routing Problem (VRP)* é um nome genérico dado a toda uma classe de problemas em que um conjunto de clientes dispersos geograficamente devem ser atendidos por uma frota de veículos. Cada cliente possui uma demanda e cada veículo uma capacidade. Os veículos partem de um depósito e devem retornar ao mesmo no fim da rota.
A Figura 1 representa uma instância do problema (esquerda) e uma possível solução com 5 veículos(direita).

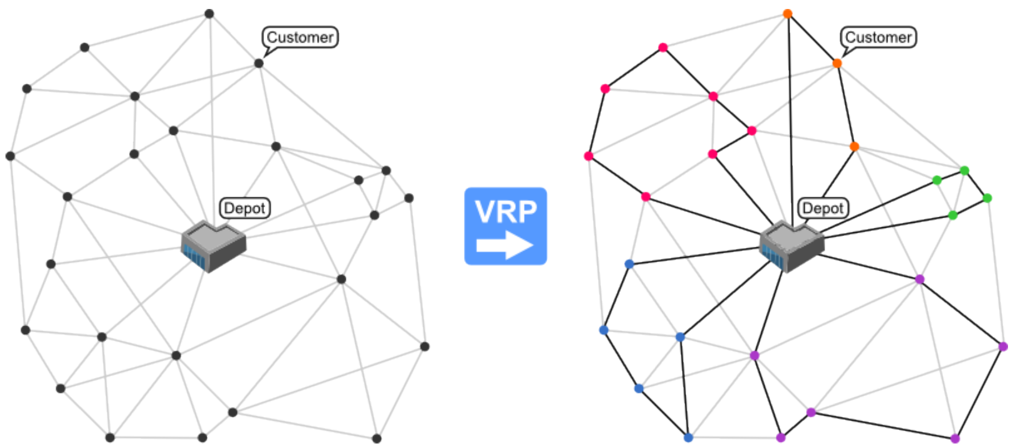

Figura 1: Representação de uma instância e uma possível solução do VRP.

Definição do problema: Seja $G = (N, A)$ um grafo direcionado completo, sendo $N = \{0, 1, \ldots, n\}$ o conjunto de vértices e $A = \{a = (i, j) : i, j \in N, i \neq j\}$ o conjunto de arestas ligando os vértices.O nó 0 corresponde ao depósito e cada nó $i \in N \backslash \{0\}$ é um cliente. Seja também $K = \{1, \ldots, k\}$ o conjunto de

k veículos (iguais) disponíveis com capacidade Q. O custo para atravessar um arco $a \in A$ é denotado por c_a, e a demanda de cada cliente $i \in N \backslash \{0\}$ é d_i.
Objetivo:
O problema consiste em definir no máximo k rotas de forma que todos os clientes sejam atendidos, sem que a capacidade de nenhum veículo seja excedida. Cada cliente deve ser atendido por somente 1 veículo, e cada veículo percorre somente uma rota. O objetivo é minimizar o custo total da(s) rota(s) (geralmente associado à distância percorrida).
Obs: não é obrigatório que todos os veículos tenham de executar uma rota.
INPUT: um conjunto de n pontos (depósito clientes), número máximo de veículos k, capacidade dos veículos, e uma matriz simétrica de custos entre cada par de vértices(arestas).
OUTPUT: a rota de cada veículo e o valor da solução final.
Instâncias:
As instâncias estão disponíveis na CVRPLIB - Capacitated Vehicle Routing Problem Library. A tabela 1 apresenta as instâncias que devem ser utilizadas no trabalho e suas características. *Instância* corresponde ao nome da instância, n é o número de pontos (depósito + clientes), k é o número de veículos, Q é a capacidade de cada veículo, e *Opt* é o valor da solução ótima.

Tabela 1: Instâncias CVRP

Instância	n	k	Q	*Opt*
Christofides, Mingozzi and Toth (1979)				
CMT1	51	5	160	524.61
CMT2	76	10	140	835.26
CMT3	101	8	200	826.14
CMT4	151	12	200	1,028.42
CMT5	200	17	200	1,291.29
Uchoa et al. (2014)				
X-n101-k25	101	25	206	27,591
X-n106-k14	106	14	600	26,362
X-n110-k13	110	13	66	14,971
X-n115-k10	115	10	169	12,747
X-n120-k6	120	6	21	13,332

As instâncias definem para o depósito e para cada cliente a sua coordenada no plano cartesiano. Dessa forma, o custo c_a deve ser calculado como sendo a distância entre os pontos i e j (distância Euclidiana).
Referências úteis: [TV02, Lap92, RKPT03, CGL+02]

2 Split Delivery Vehicle Routing Problem (SDVRP)

O SDVRP é muito semelhante ao VRP (ver sessão 1). A única diferença entre eles é que o SDVRP permite que a demanda de um cliente seja particionada, ou seja, um ou mais veículos podem fazer uma entrega no mesmo cliente.
Definição do problema: Seja $G = (N, A)$ um grafo direcionado completo, sendo $N = \{0, 1, \dots, n\}$ o conjunto de vértices e $A = \{a = (i, j) : i, j \in N, i \neq j\}$ o conjunto de arestas ligando os vértices.O nó 0 corresponde ao depósito e cada nó $i \in N \backslash \{0\}$ é um cliente. Seja também $K = \{1, \dots, k\}$ o conjunto de k veículos iguais com capacidade Q. O custo para atravessar um arco $a \in A$ é denotado por c_a, e a demanda de cada cliente $i \in N \backslash \{0\}$ é d_i.
Objetivo: Atribuir a cada veículo uma rota de entrega, passando por um sub-conjunto de clientes, de forma a minimizar o custo total dos arcos atravessados. Deve-se assegurar que a demanda de todos clientes seja atendida, e o total entregue por cada veículo não seja maior que sua capacidade.
Obs: não é obrigatório que todos os veículos tenham de fazer entregas.

INPUT: um conjunto de n clientes e depósitos, número máximo de veículos, capacidade dos veículos, e uma matriz simétrica de custos entre cada par de vértices. Os custos podem ser calculados como sendo a distância Euclidiana entre dois pontos, quando as coordenadas são fornecidas.
OUTPUT: a rota de cada veículo e o valor da solução final. Para cada rota, deve-se explicitar a fração de demanda atendida de cada cliente.
Instâncias: As instâncias estão disponíveis no seguinte link: `http://www.uv.es/belengue/SDVRP/sdvrplib.tar.gz`. Junto ao arquivo existe um README para maiores detalhes. A tabela 2 apresenta quais instâncias devem ser utilizadas no trabalho e suas características. Onde *Instância* é o nome da instância, n é o número de pontos (índice 0 corresponde ao depósito, e o restante aos clientes), k_{min} é o número mínimo de veículos necessários, Q é a capacidade de cada veículo, e BKV é o melhor valor da solução conhecido (*best known value*). Valores assinalados com * correspondem à solução ótima.

Tabela 2: Instâncias SDVRP

Instância	n	k_{min}	Q	BKV
eil51.sd	51	5	160	*521
eil30.sd	30	3	4500	*510
eilA76.sd	76	10	140	832
eil22.sd	22	4	6000	*375
S51D1.sd	51	3	160	458
eilB101.sd	101	14	112	1077
eil33.sd	33	4	8000	835
eil23.sd	23	3	4500	*569
eilA101.sd	101	8	200	817
S51D2.sd	51	9	160	726

Referências úteis: [AS08, TV02, CGL+02].

3 Sequential Ordering Problem - SOP

O Problema de Ordenação Sequencial é uma variação do Problema do Caixeiro Viajante assimétrico (*aTSP - Assymetric Travelling Salesman Problem*). Também conhecido como *Asymmetric traveling salesman problem with precedence constraints*, o objetivo do problema é igual ao do aTSP, ou seja, encontrar um ciclo Hamiltoniano (um caminho que passe por todos os pontos de um grafo somente uma vez) retornando ao ponto inicial, de forma que a distância total percorrida (custo das arestas) seja o menor possível. No SOP porém, deve-se considerar regras de precedência entre os pontos.
Definição do Problema:
Dado um grafo $G = (V, A)$ com $V = \{1, \ldots, n\}$ e $A = \{(i, j), \forall i, j \in V : i \neq j\}$, um custo c_{ij} para cada arco $a \in A$. Vértices 1 e n representam o mesmo nó (inicial e final) do ciclo. É dado também um dígrafo de precedência $P = (V, R)$, onde V é o mesmo conjunto de vértices de G, enquanto que um arco $(i, j) \in R$ representa uma relação de precedência, *i.e.* i deve ser visitado antes que j em qualquer caminho factível. O objetivo é minimizar o custo dos arcos $a \in A$ utilizados para visitar todos os vértices somente uma vez, respeitando as regras de precedência entre os mesmos.
INPUT: uma matriz de distâncias c_{ij}. Se $c_{ij} \geq 0$ o valor representa a distância entre os nós i e j. Se $c_{ij} = -1$ significa que j deve preceder i na rota. **OUTPUT**: um ciclo Hamiltoniano (sequência de pontos a serem visitados) que respeite as regras de precedência e o custo da solução.

Instâncias: As instâncias para testes estão disponíveis no seguinte link: `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/sop/`. As soluções ótimas encontram-se em `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/SOP.html`.
A tabela 3 apresenta quais instâncias devem ser utilizadas no trabalho e suas características. Onde *Instância* é o nome da instância, n é o número de pontos e *Opt* é o valor da solução ótima.

Referências úteis: [KK91, MKCS02, MSG08] **Dica:** buscar trabalhos sobre o aTSP pode auxiliar na formulação do problema.

Tabela 3: Instâncias SOP

Instância	n	*Opt*
ESC07	09	2125
ESC12	14	1675
ESC25	27	1681
ESC47	49	1288
ESC63	65	62
ESC78	79	18230
rbg048a	50	351
rbg109a	111	1038
br17.12	18	55
ft53.1	54	7531

Referências

[AS08] Claudia Archetti and Maria Grazia Speranza. The split delivery vehicle routing problem: A survey. In *The vehicle routing problem: Latest advances and new challenges*, pages 103–122. Springer, 2008.

[CGL+02] J-F Cordeau, M Gendreau, G Laporte, J-Y Potvin, and F Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53(5):512–522, may 2002.

[KK91] Mikio Kubo and Hiroshi Kasugai. The precedence constrained traveling salesman problem. *Journal of the Operations Research Society of Japan*, 34(2):152–172, 1991.

[Lap92] Gilbert Laporte. The Vehicle Routing Problem : An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59:345–358, 1992.

[MKCS02] Chiung Moon, Jongsoo Kim, Gyunghyun Choi, and Yoonho Seo. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European Journal of Operational Research*, 140(3):606 – 617, 2002.

[MSG08] R. Montemanni, D.H. Smith, and L.M. Gambardella. A heuristic manipulation technique for the sequential ordering problem. *Computers & Operations Research*, 35(12):3931 – 3944, 2008. Part Special Issue: Telecommunications Network Engineering.

[RKPT03] T.K. Ralphs, L. Kopman, W.R. Pulleyblank, and L.E. Trotter. On the capacitated vehicle routing problem. *Mathematical Programming*, 94(2):343–359, 2003.

[TV02] Paolo Toth and Daniele Vigo. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1?3):487 – 512, 2002.