PNV-3512 Planejamento de Sistemas Logísticos

As atividades de PNV-3512 associadas ao tópico de roteirização de veículos consistem em implementações computacionais que deverão ser feitas em grupos limitados a 2 participantes. Envie a composição do seu grupo para andbergs@usp.br, para receber o número do problema que irá resolver.

Atividades 1

1.1. Geração Exaustiva de Rotas & Permutações

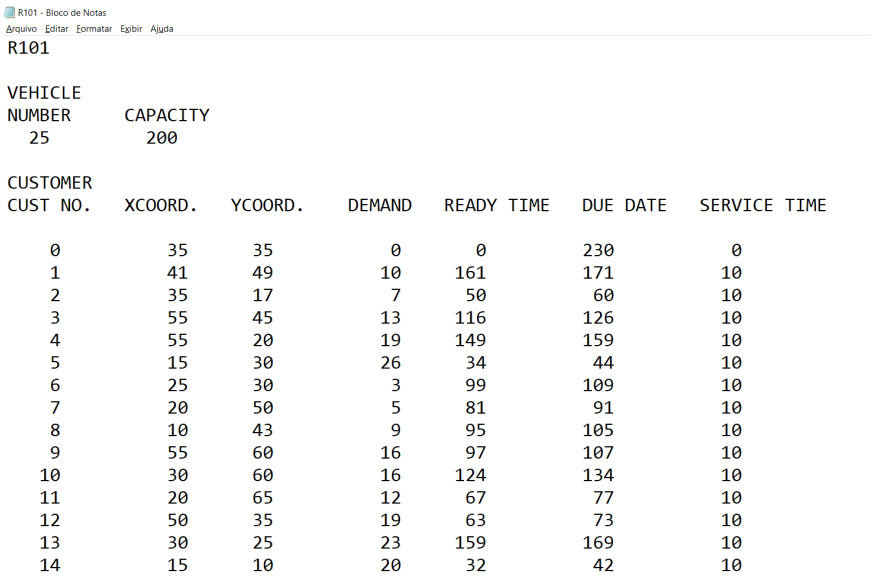
Dado o conjunto $N = \{1, \dots, n\}$ de clientes, gerar todas as possíveis rotas pela geração exaustiva de todas as combinações de clientes $\binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{n}$, aplicando o pseudo-código apresentado em aula, ou por meio da biblioteca itertools do Python. OBS: Conhecendo a demanda d_i de cada cliente i, é possível saber a priori se a combinação de clientes é tal que a capacidade do veículo é excedida.

Para cada combinação gerada, também é necessário resolver o problema do caixeiro viajante associado, para determinar a menor distância da rota. Isto equivale a avaliar todas as permutações possíveis dos clientes presentes na rota.

Exemplo: se uma combinação é formada pelos clientes {3,5,7}, então 3! permutações devem ser geradas e avaliadas: 3-5-7, 3-7-5, 5-3-7, 5-7-3, 7-3-5, 7-5-3. No caso da rede ser simétrica ($c_{ij} = c_{ji}$), algumas permutações possuirão a mesma distância total. As permutações poderão ser feitas por meio da implementação do pseudo-código apresentado em aula, ou por meio da biblioteca itertools do Python.

Esta atividade consiste em implementar o algoritmo de geração de combinações e de permutações para a base de clientes do seu grupo. No e-disciplinas estão disponibilizadas as bases de dados R101 a R113, para os grupos 1 a 13, respectivamente.

Cada arquivo possui a estrutura abaixo indicada.



R101 - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda

R101

VEHICLE
NUMBER CAPACITY
25 200

CUSTOMER

CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE TIME
0	35	35	0	0	230	0
1	41	49	10	161	171	10
2	35	17	7	50	60	10
3	55	45	13	116	126	10
4	55	20	19	149	159	10
5	15	30	26	34	44	10
6	25	30	3	99	109	10
7	20	50	5	81	91	10
8	10	43	9	95	105	10
9	55	60	16	97	107	10
10	30	60	16	124	134	10
11	20	65	12	67	77	10
12	50	35	19	63	73	10
13	30	25	23	159	169	10
14	15	10	20	32	42	10

Para esta atividade, adotar velocidade = 1 (o tempo é igual à distância euclidiana); capacidade = 60. Apenas serão necessárias as coordenadas (X,Y), e a demanda. Teste os algoritmos para 2 casos: com os primeiros 20 clientes (mais o depósito) e com os primeiros 40 clientes (mais o depósito).

A entrega deve incluir: arquivo de entrada de dados, código fonte em Python, e uma apresentação sucinta indicado as atividades realizadas.

Entrega: 4/7 no e-disciplinas.

1.2. Algoritmo Clarke & Wright

Implementar o algoritmo Clarke & Wright (versão sequencial, apresentada em aula) para a base de dados associada ao seu grupo (100 clientes). Para esta atividade, adotar velocidade = 1 (o tempo é igual à distância euclidiana); capacidade = 200. Por hora, apenas serão necessárias as coordenadas (X,Y) e a demanda.

Proponha também uma versão probabilística do algoritmo Clarke & Wright, e compare os resultados com a versão determinística.

A entrega deve incluir: arquivo de entrada de dados, código fonte em Python, e uma apresentação powerpoint sucinta indicado as atividades realizadas e a solução obtida (função objetivo, rotas e tempo de processamento).

Entrega: 4/7 no e-disciplinas.

1.3. Heurística Construtiva Probabilística

Usando o procedimento baseado em múltiplos recomeços, desenvolver um algoritmo que gera as rotas dos veículos de forma probabilística, seguindo o procedimento abaixo:

1) Estimar quantos veículos são necessários para atender toda a demanda m = $\lceil$soma da demanda total dividida pela capacidade do veículo$\rceil$.

2) Escolher m clientes que necessariamente estarão em rotas distintas. Por exemplo, achar os m clientes que estão mais afastados do depósito, e afastados entre si. Alocar cada um destes clientes a uma rota distinta, e tirá-los da fila de clientes não roteirizados.

3) Para cada rota, elaborar a lista ordenada dos clientes mais próximos (à rota), que ainda estão pendentes de alocação. Dentre os p clientes mais próximos (p:1...5), sortear um cliente e adicionar à rota. Este procedimento pode ser feito para uma rota até que ela fique completa (não comporte mais nenhum cliente), ou pode ser feita sequencialmente (adiciona um cliente à rota 1, depois um cliente à rota 2, etc., e retorna à rota 1, até que todos os clientes estejam alocados). OBS: Ao escolher o cliente que será adicionado à rota, o mesmo deve ser inserido na posição que resulta no menor acréscimo de custo.

4) Repetir o processo até que todos os clientes estejam alocados. Se não for possível alocar todos os clientes, repetir o algoritmo, a partir do passo 2, fazendo $m = m+1$.

O procedimento de múltiplos recomeços deverá ser rodado 1000 vezes, variando a semente do gerador de números aleatórios.

Framework - Heurística Construtiva Probabilística c/Múlt. Recomeços

```
procedure MultiStart
  f* ← ∞;
  while stopping criterion not satisfied do
    Construct feasible solution:
      S ← ConstructRandomSolution;
    if f(S) < f* then
      S* ← S;
      f* ← f(S);
    end
  end
return S*;
```

Fonte: Marti et al. (2012)

Framework - Heurística Construtiva Probabilística

```
procedure ConstructRandomSolution
  Initialize solution: S ← ∅;
  Initialize candidate set:
    C ← {s ∈ E\S | S∪{s} is not infeasible};
  while C ≠ ∅ do
    Select s ∈ C at random;
    Add s to solution: S ← S∪{s};
    Update candidate set:
      C ← {s ∈ E\S | S∪{s} is not infeasible};
  end
return S;
```

Fonte: Marti et al. (2012)

Testar o algoritmo para a base de dados do seu grupo, para os 100 clientes. Para esta atividade, adotar velocidade = 1 (o tempo é igual à distância euclidiana); capacidade = 200.

Reportar os tempos de processamento, uma saída gráfica visual das rotas da melhor solução, e o valor da função objetivo (mínimo, média, máximo, distribuição).

A entrega deve incluir: arquivo de entrada de dados, código fonte em Python, e uma apresentação powerpoint sucinta indicado as atividades realizadas. Reportar o tempo de processamento, o valor da função objetivo (mínimo, média, máximo, distribuição), e as rotas da solução de menor custo.

Entrega: 4/7 no e-disciplinas.