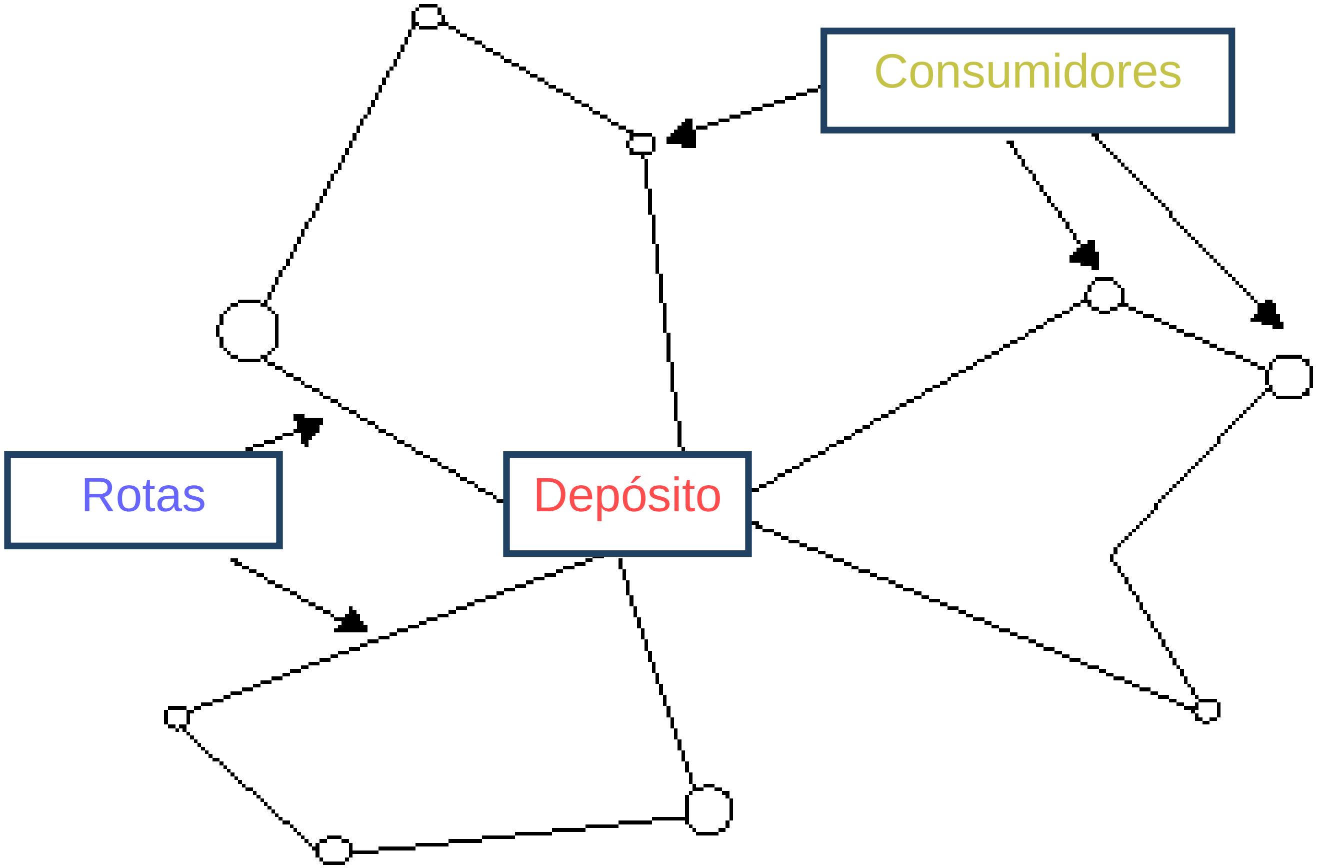
MÉTODO DE PESQUISA EM VIZINHANÇA VARIÁVEL APLICADO À RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS

Dárlinton B. Feres Carvalho
darlinton@enut.ufop.br

Guilherme Arantes de Oliveira
guilhermeaol@yahoo.com.br

Marcone Jamilson Freitas Souza
marcone@iceb.ufop.br

Exemplo:



Formulação do Problema do Roteamento de Veículos (PRV):

- Seja $G = (V, E)$ um grafo não direcionado, onde $V = \{v0, v1,..., vn\}$ é o conjunto dos vértices e $E = \{(vi, vj): vi ,vj \in V, i < j\}$ é o conjunto de arestas.
  - O vértice $v0$ representa o depósito, sendo este a base de uma frota de veículos idênticos de capacidade Q, enquanto os vértices remanescentes correspondem às cidades ou consumidores.
- Cada consumidor vi tem uma demanda não negativa qi e $q0 = 0$.
- Supõe-se que existe um número ilimitado de veículos no depósito.
- A cada aresta (vi, vj) está associada uma distância não negativa cij que representa a distância entre os consumidores.

Formulação do Problema do Roteamento de Veículos (PRV):

- O Problema de Roteamento de Veículos consiste em determinar o conjunto de rotas que deverão ser feitas pelos veículos minimizando os custos de transporte, dado pela distância e respeitando as seguintes condições:

a) Cada rota começa e termina no depósito;

b) Toda cidade de $V \setminus \{v0\}$ é visitada somente uma vez por somente um veículo;

c) A demanda total de qualquer rota não deve superar a capacidade Q de um veículo.

Nossa proposta:

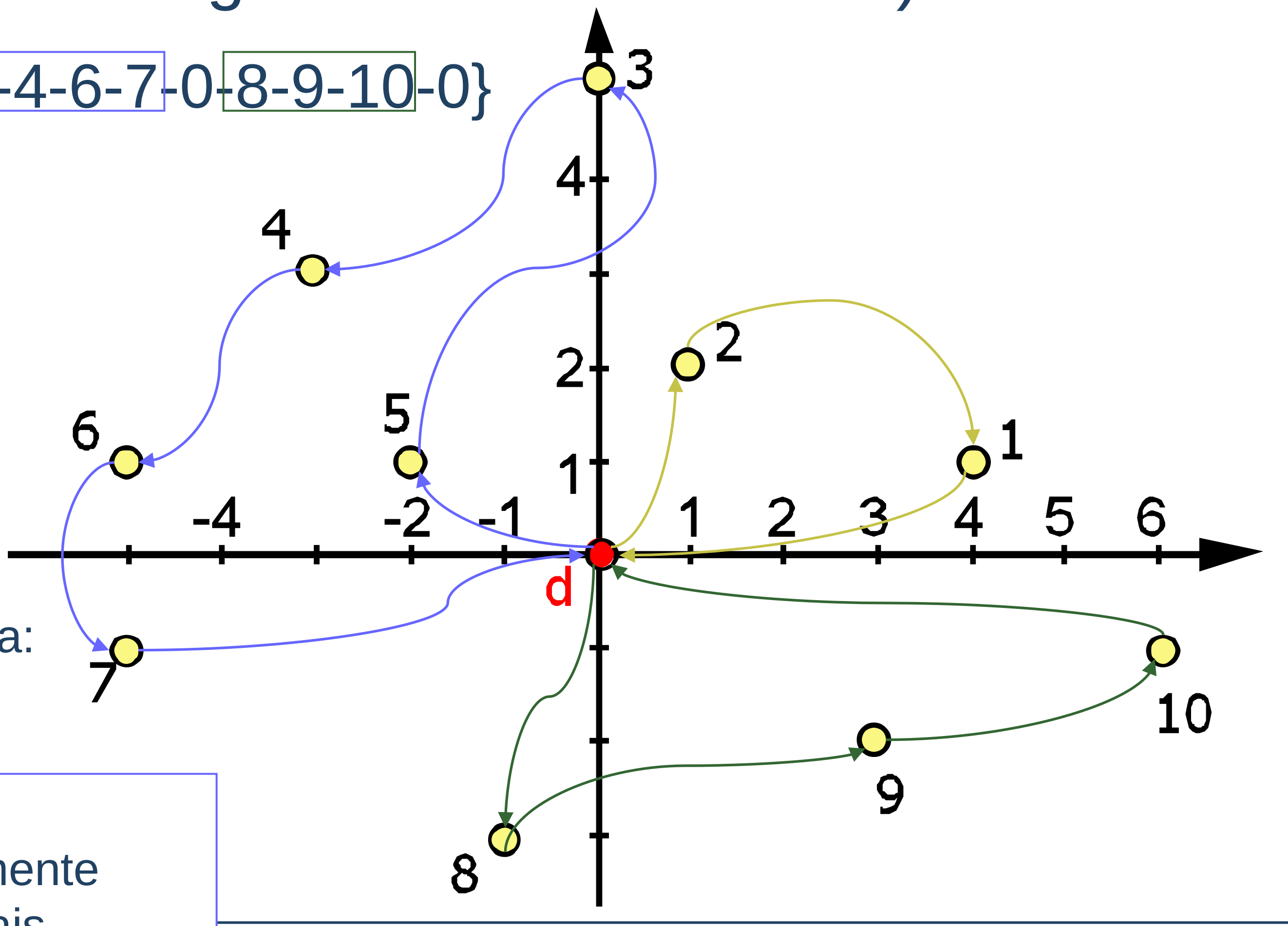
- Um método de duas fases para a resolução do PRV:
  - Fase de construção GRASP.
  - Refinamento usando o Método de Pesquisa em Vizinhança Variável (VNS).
    - Função de avaliação objetivando minimizar as distâncias percorridas.
    - As estruturas de vizinhança simples capazes de escapar de ótimos locais.
    - Método de Descida em Vizinhança Variável (VND) para busca local no VNS.

Representação do PRV:

- Assumimos a representação usada por Pradenas & Parada (1999). Uma solução do PRV é representada por meio de uma permutação de cidades, numeradas de 1 a *n*, separadas em tantas partições quantos forem o número de veículos usados.
  - O elemento separador é representado pelo valor **zero** e **indica** o **depósito**.
- Por exemplo, se há 6 consumidores, 3 veículos e a solução *s* é {0-3-4-0-1-5-2-0-6-0} então as rotas dos veículos, denominadas pétalas, são {0-3-4-0}, {0-1-5-2-0} e {0-6-0}.

Construção de uma solução inicial pelo método GRASP (Procedimento de busca adaptativa gulosa e randomizada). :

s ={0-2-1-0-5-3-4-6-7-0-8-9-10-0}



Critério para nova rota:

Ex: q2+q1+q3>Q

Critério de escolha:
Selecionar aleatoriamente dentre os vizinhos mais próximos do último escolhido.

Movimentos e estruturas de vizinhança

- Consideramos seis estruturas de vizinhança, a saber: N^1, N^2, N^3, N^4, N^5, N^6
- O primeiro movimento consiste na troca de dois números inteiros em uma mesma pétala da solução.
  - Estes números representam apenas os consumidores.
- O segundo, terceiro, quarto, quinto e sexto movimentos consistem em efetuar uma, duas, três, quatro ou cinco trocas, respectivamente, entre quaisquer elementos da solução.

Movimentos e estruturas de vizinhança:

- s ={0-3-4-0-1-5-2-0-6-0}
- s' ={0-3-4-0-1-**2**-**5**-0-6-0} $\in N^1(s)$.
- s' ={0-3-**5**-0-1-**4**-2-0-6-0} $\in N^2(s)$.
- s' ={0-3-**5**-0-**6**-**4**-2-0-**1**-0} $\in N^3(s)$.
- s' ={0-**2**-**5**-0-**6**-**4**-**3**-0-**1**-0} $\in N^4(s)$.
- s' ={0-**2**-**5**-**6**-**0**-**4**-**3**-0-**1**-0} $\in N^5(s)$.
- s' ={0-**2**-**5**-**6**-**0**-**4**-**3**-**1**-**0**-0} $\in N^6(s)$.

Função Objetivo:

- Função objetivo baseada em penalização.
- Seja $f1(s)$ representando a função objetivo pura da solução *s:*
  - Soma das distâncias percorridas por todos os veículos.
- $O(s)$ o total das sobrecargas dos veículos associada a esta solução, caso exista.
- Função objetivo $f(\mathrm{s}) = f1(s) + \beta \times O(s)$
  - β é um fator de penalidade não negativo.

```
procedimento VNS(s, t);
    enquanto ( tempo_execução < t )
        k ← 1;
        enquanto ( k <= 10 )
            escolha(k)
                1: s'←vizinho_qualquer(s, N^1);
                2: s'←vizinho_qualquer(s, N^2);
                3: s'←vizinho_qualquer(s, N^3);
                4: s'←vizinho_qualquer(s, N^4);
                5: s'←vizinho_qualquer(s, N^5);
                6: s'←vizinho_qualquer(s, N^5);
                7: s'←vizinho_qualquer(s, N^6);
                8: s'←vizinho_qualquer(s, N^6);
                9: s'←vizinho_qualquer(s, N^6);
               10: s'←vizinho_qualquer(s, N^6);
            fim escolha;
            s'←VND(s');
            se ( f(s') < f(s))
                s ← s';
                k ← 1;
            senão
                k ← k + 1;
            fim-se;
        fim enquanto;
    fim enquanto;
    Retorne s;        {Retorne a melhor solução}
fim VNS;
```

Figura 1: Método VNS aplicado ao PRV.

```
procedimento VND(s);
    s'← s;
    k ← 1;
    enquanto ( k <= 2 )
        escolha(k)
            1: s'←descida_1opt(s');
            2: s'← descida_2opt (s');
        fim escolha;
        se ( f(s') < f(s))
            s ← s';
            k ← 1;
        senão
            k ← k + 1;
        fim-se;
    fim enquanto;
    Retorne s;        {Retorne a melhor solução}
fim VND;
```

Figura 2: Método VND aplicado ao PRV.

Método proposto:

procedimento GRASP_VNS(α, *t*);
$s \leftarrow$ ConstruaSolucaoInicial(α);
$s \leftarrow$ VNS(*s*, *t*);
Retorne *s*; {Retorne a melhor solução}
fim GRASP_VNS;

Figura 3: Método proposto.

Alguns testes:

Teste	Instância	# cid	Cap. vei.	Melhor Valor conhecido	Tempo CPU (seg)	α	GRASP_VNS				
							Melhor Valor	Valor Médio	Desvio (%)	# veic *	Média # veic
1	c50.dat	50	160	524.61	150	0,01	524.61	532.87	1.57	5	5
2	c50.dat	50	160	524.61	150	0,35	527.67	545.99	4.07	5	5.44
3	c50.dat	50	160	524.61	300	0,35	524.61	541.24	3.17	5	5.28
4	c75.dat	75	140	835.26	300	0,01	842.96	865.56	3.62	10	10.96
5	c75.dat	75	140	835.26	150	0,35	844.44	875.54	4.82	11	10.92

Todos os experimentos foram realizados em um PC com processador Pentium IV, de 1.8 GHz, 512 MB de RAM rodando a plataforma Windows XP.

$$\text{Desvio} = \frac{\left(\bar{f} - f^*\right) \times 100}{f^*}$$

http://ina.eivd.ch/collaborateurs/etd/problemes.dir/vrp.dir/vrp.html

Conclusões:

- O método proposto é simples e requer poucos parâmetros, basicamente o fator de aleatoriedade relativo à construção de uma solução inicial, o tempo de processamento e a seqüência de estruturas de vizinhança.
- A partir desses parâmetros pode-se representar características do problema que irão guiar o método e muitas vezes determinar a qualidade da solução.