

5

CAPÍTULO

BUSCA EM VIZINHANÇA VARIÁVEL PARA LOCALIZAÇÃO E ROTEAMENTO

Silva, Lorrany Cristina da ¹ *;
Queiroz, Thiago Alves de ²

¹ Unidade de Matemática e Tecnologia, Universidade Federal de Goiás – Regional Catalão

² Unidade de Matemática e Tecnologia, Universidade Federal de Goiás – Regional Catalão

* **email: cristina_lorrany@yahoo.com.br**

Resumo: Apresenta-se uma heurística baseada na estratégia de busca em vizinhança variável para o problema de localização e roteamento. A heurística possui uma fase inicial para gerar uma solução aleatória, que posteriormente é melhorada por um procedimento de busca local. Na busca local, cinco operadores de vizinhança são aplicados. Os operadores são baseados em trocas, inserções e exclusões de clientes/depósitos considerando rotas e depósitos já estabelecidos. A heurística foi codificada na linguagem C considerando a versão capacitada do problema. Experimentos computacionais foram realizados em instâncias de outros trabalhos, validando os resultados em comparação a outras propostas da literatura.

Palavras-chave: Problema de Localização e Roteamento; Busca em Vizinhança Variável; Heurística

1. Introdução

O problema de localização de facilidades e roteamento de veículos é referenciado na literatura como Problema de Localização e Roteamento – PLR (Laporte e Norbert, 1981). No problema de localização de facilidades, deseja-se determinar, a um custo mínimo, a quantidade e a localização de facilidades (depósitos, empresas, hospitais, etc.) com o objetivo de suprir a demanda de clientes. No roteamento de veículos, deseja-se encontrar rotas de custo mínimo, partindo de depósitos, para que veículos atendam a demanda por mercadorias de clientes.

O PLR considera $I=\{1, \dots, n\}$ como o conjunto de clientes e $J=\{d_1, \dots, d_m\}$ sendo o conjunto de depósitos. Cada depósito $d_j \in J$ possui uma capacidade de estoque b_j , um custo fixo f_j de abertura/localização e possui um único veículo associado, ou seja, assume-se que existe no máximo uma rota associada a cada depósito. Por outro lado, cada cliente $i \in I$ tem uma demanda q_i , a qual deve ser atendida a partir de algum depósito. O custo para viajar entre i e r é denotado por c_{ir} , com $i, r \in I \times J$.

O objetivo do PLR é minimizar o custo total incorrido da localização de um subconjunto de depósitos, para atender a demanda total dos clientes, e das rotas partindo dos depósitos. As restrições que foram consideradas são: cada cliente é atendido por exatamente um depósito, a capacidade de estoque de cada depósito aberto não pode ser excedida, a demanda total dos clientes deve ser atendida. Além disso, não existe custo para usar um veículo.

O PLR é um problema NP-difícil, uma vez que possui dois subproblemas que são NP-difíceis, conforme Garey e Johnson (1979). Assim, não se espera algoritmos exatos de tempo polinomial para resolvê-lo. Alguns trabalhos que utilizam estratégias exatas foram propostos em Belenguer et al. (2011), Laporte et al. (1986) e em Laporte e Nobert (1981). Belenguer et al. (2011) fizeram uma formulação matemática, resolvida por um algoritmo branch-and-cut, capaz de lidar com instâncias de até 50 clientes e 10 depósitos. Para tanto, várias rotinas de separação especializadas e novas desigualdades válidas foram propostas.

Por outro lado, existem as heurísticas, que podem retornar soluções muito boas ou até mesmo ótimas, porém em um tempo computacional que é viável para aplicações práticas. Jarboui et al. (2013) desenvolveram heurísticas baseadas no método de Busca em Vizinhança Variável. O algoritmo de Jarboui et al. (2013) melhorou os resultados publicados em Albareda-Sambola et al. (2005) e Derbel et al. (2010). Tuzun e Burke (1999) consideraram o PLR com rotas capacitadas e depósitos incapacitados, resolvendo-o usando uma busca Tabu. Restrições práticas que podem ser aplicadas em problemas de roteamento, incluindo o PLR, podem ser obtidas nos trabalhos de Queiroz e Miyazawa (2013, 2014) e Gonçalves e Queiroz (2014).

Nesse trabalho, apresenta-se uma heurística baseada no método de busca em vizinhança variável, seguindo o trabalho de Jarboui et al. (2013), para resolver o PLR. A Seção 2 apresenta detalhes da heurística, enquanto a Seção 3 discute os testes numéricos realizados. Conclusões e direções para trabalhos futuros são dados na Seção 4.

2. Heurística de Busca em Vizinhança Variável

O algoritmo da Figura 1 traz o método de Busca em Vizinhança Variável (BVV), que atua em diferentes regiões do espaço de busca. Esta estratégia foi proposta por Mladenović e Hansen (1997) e lida com a exploração do espaço de busca por meio da avaliação de estruturas de vizinhança de uma dada solução. A ideia é obter uma solução que seja ótima global em relação a todos os operadores de vizinhança.

O algoritmo geral da BVV mostra uma estrutura em três etapas: (i) obter uma solução por meio da aplicação do operador P_k (linha 5); (ii) realizar uma busca local determinística partindo da solução gerada na etapa anterior (linha 6); (iii) e, aceitar a nova solução da busca local somente se ela for melhor comparada a melhor solução atual (linhas 7 e 8). Deseja-se no PLR sempre a solução de menor custo.

```

1  Gerar uma solução inicial  $s$ ;
2  Repetir
3     $k \leftarrow 1$ ;
4    Enquanto  $k \leq k_{max}$  faça
5       $s' \leftarrow$  aplicar o operador de vizinhança  $P_k$  em  $s$ ;
6       $s'' \leftarrow$  Aplicar uma rotina de busca local em  $s'$ ;
7      Se  $s''$  é melhor do que  $s$  então
8         $s \leftarrow s''$ ;  $k \leftarrow 1$ ;
9      Senão
10        $k \leftarrow k + 1$ ;
11 Até alcançar a condição de parada
12 Retorne a solução  $s$ ;
```

Figura 1: Algoritmo geral da BVV.
Fonte: Jarboui et al. (2013).

Em linhas gerais, a BVV gera uma solução inicial qualquer s , muitas vezes de forma aleatória. A cada interação, uma solução s' é obtida a partir da aplicação do operador P_k em s , para em seguida, uma busca local ser realizada sobre s' . A busca local explora a vizinhança de s' em busca de uma solução melhor, isto é, s'' . Se s'' for melhor do que s , então, faz-se a atualização de s e o processo de aplicação dos operadores reinicia a partir do operador P_1 . Caso contrário, considera-se um novo operador de vizinhança, incrementando o valor de k .

2.1 Operadores de Vizinhança e Busca Local

Utilizam-se cinco operadores de vizinhança, P_1 , P_2 , P_3 , P_4 e P_5 , representadas nas Figuras 3, 4, 5, 6 e 7, respectivamente. Esses operadores contêm operações que foram propostas por Jarboui et al. (2013) no contexto do PLR, baseando-se em operações para excluir, inserir e trocar clientes de rotas ou fechar e abrir depósitos. É importante mencionar que tais autores apenas descreveram as estruturas, sem dar detalhes da codificação computacional. Dessa forma, as estruturas descritas adiante estão baseadas na codificação desenvolvida nesse trabalho.

Os quatro primeiros operadores, P_1 a P_4 , lidam com a estrutura do problema de roteamento, isso é, modificam apenas as rotas da solução em que foram aplicados. Para cada um deles, Jarboui et al. (2013) fizeram uma proposta de mudança intra-rota, em que o

operador trabalha na modificação de uma mesma rota atendida por um depósito, e uma mudança inter-rota, pelo qual a mudança ocorre entre duas rotas diferentes, isto é, cada qual sendo atendida por um depósito. A Figura 2 traz as rotas usadas como padrão para a aplicação dos operadores, sendo que as Figuras 3 a 7 mostram as rotas modificadas.

O último operador, P_5 , atua na questão de abrir ou fechar um depósito, associando a rota no novo depósito aberto. Por isso, não há proposta para uma mudança intra-rota ou inter-rota para P_5 .

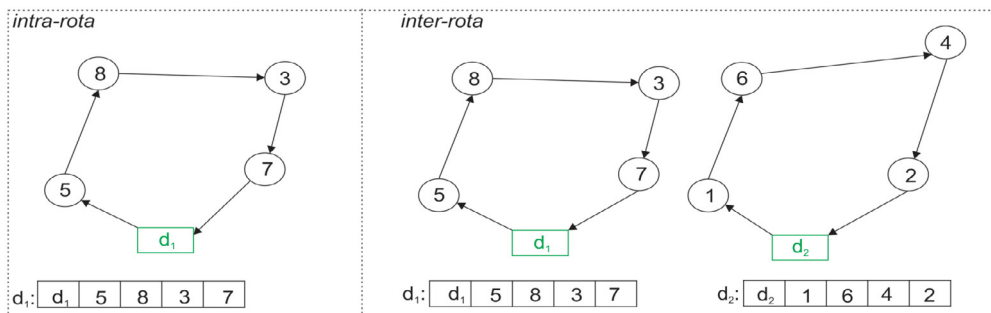


Figura 2: Exemplo de rotas.
Fonte: Elaborada pela Autora.

Os operadores P_1 e P_2 descrevem operações de inserção e troca de clientes em rotas, respectivamente. Na Figura 3 está a intra-rota do operador P_1 , que realiza o movimento de remover um cliente de uma dada posição i e inseri-lo depois de outra posição r . Nesta figura também é apresentada a operação inter-rota, a qual busca inserir um cliente i de uma dada rota, após um cliente r de outra rota. A escolha dos clientes, posições e rotas são feitas de forma aleatória.

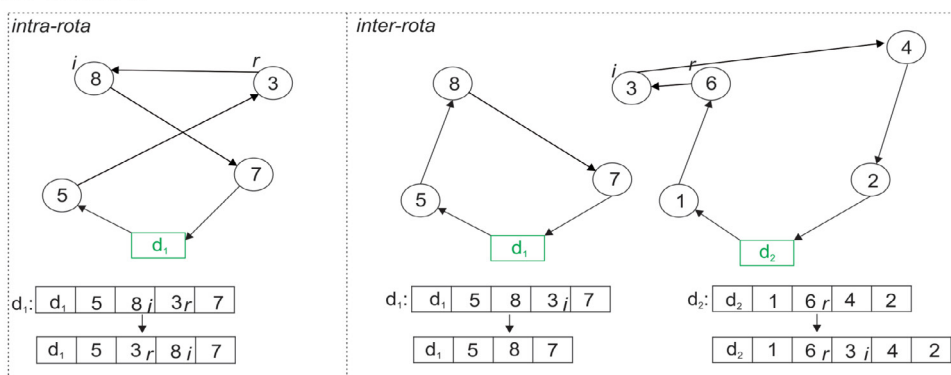


Figura 3: Representação do operador P_1 .
Fonte: Elaborada pela Autora.

A Figura 4 apresenta as operações realizadas pelo operador P_2 na intra-rota e inter-rota. Na intra-rota ocorre a troca no vetor ordenado entre os clientes das posições i e r , enquanto que na inter-rota é feita a troca tomando duas rotas distintas e envolvendo os clientes das posições i em uma rota, e r da outra rota.

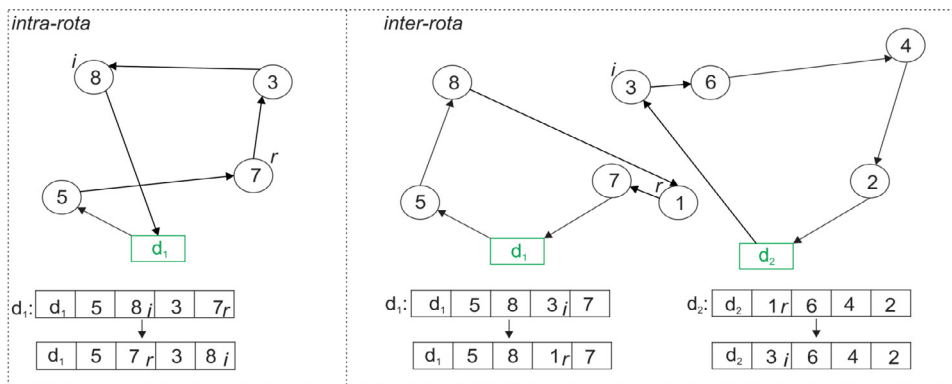


Figura 4. Representação do operador P_2 .
Fonte: Elaborada pela Autora.

O operador P_3 , que está ilustrado na Figura 5, trabalha com a inserção de uma sequência de clientes, iniciando na posição i e que é inserida após a posição r . Este operador generaliza o operador P_1 , o qual considera a inserção de um único cliente. No caso da intra-rota, a inserção ocorre na mesma rota, enquanto que na inter-rota é observado duas rotas distintas, uma para obter a seqüência e a outra para indicar a posição r .

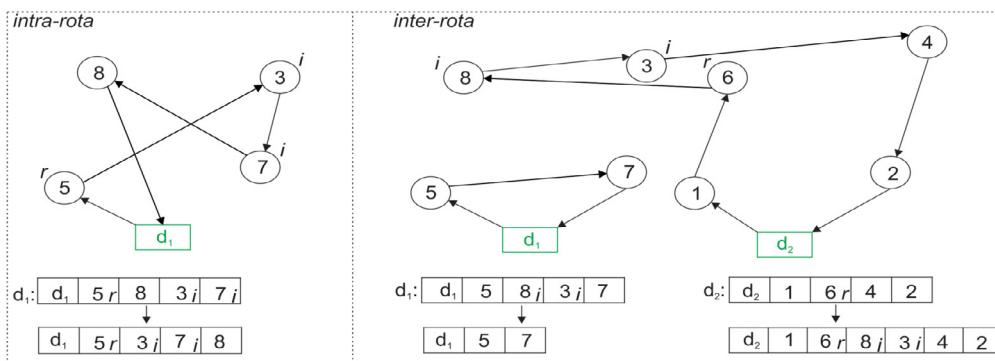


Figura 5. Representação do operador P_3 .
Fonte: Elaborada pela Autora.

Na Figura 6 está um exemplo de aplicação do operador P_4 , que é em uma modificação de P_3 . Em P_4 , a única diferença comparado a P_3 ocorre durante a inserção dos clientes, que agora é feita em ordem inversa, tanto na intra-rota como na inter-rota.

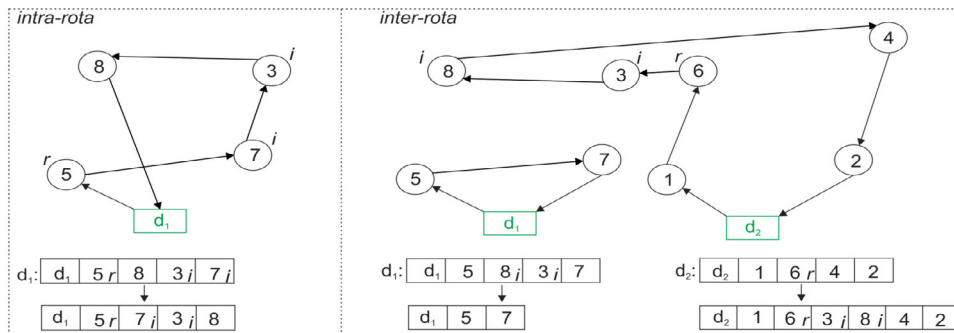


Figura 6. Representação do operador P₄.
 Fonte: Elaborada pela Autora.

O operador P₅, representado na Figura 7, trabalha no fechamento de um depósito e na localização de outro ainda não aberto. Com este operador, espera-se diversificar a busca, abrindo novos depósitos e fechando outros já abertos. A escolha de qual depósito abrir é feita de forma aleatória.

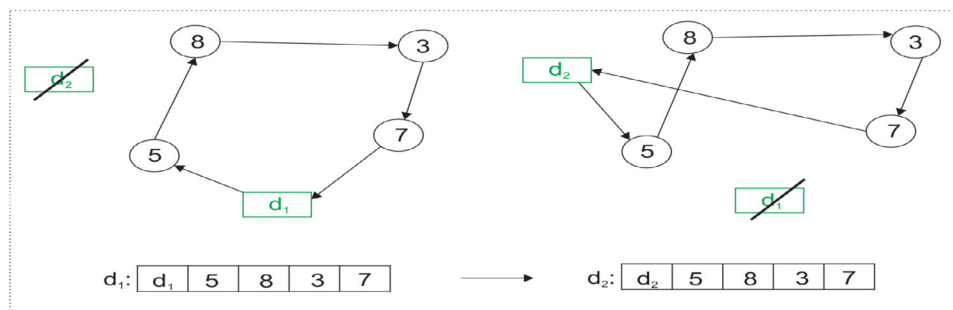


Figura 7. Representação do operador P₅.
 Fonte: Elaborada pela Autora.

O algoritmo usado para realizar a busca local é descrito na Figura 8 e consiste na DVV (método da Descida em Vizinhança Variável). Nele, os operadores P₁ a P₅ são aplicados sequencialmente. Observe que sempre que uma solução melhor que a melhor atual é obtida, reinicia-se a aplicação dos operadores fazendo k receber 1. A rotina de busca local finaliza quando nenhum dos operadores consegue melhorar a melhor solução atual.

Na versão da DVV considerada por Jarboui et al. (2013), um operador P_k é aplicado até não conseguir melhorar mais a solução, enquanto a versão da Figura 8 aplica o operador uma única vez (linha 3) e já parte para o teste condicional da linha 4. Outras versões de busca local usando a DVV podem ser encontradas no trabalho de Jarboui et al. (2013).


```

1 Faça  $k \leftarrow 1$  e seja  $s$  uma solução fornecida;
2 Enquanto  $k \leq 5$  faça
3      $s' \leftarrow$  aplique o operador  $P_k$  em  $s$ ;
4     Se  $s'$  é melhor que  $s$  então
5          $s \leftarrow s'$ ;
6          $k \leftarrow 1$ ;
7     Senão
7          $k \leftarrow k + 1$ ;
8 Retornar a solução  $s$ ;

```

Figura 8. Busca local usando o método DVV.
Fonte: Elaborada pela Autora.

Considera-se a função $f(s)$, sendo s a solução em avaliação, para o cálculo do valor de uma solução. Uma rota iniciada em um depósito d_i visita uma sequência de clientes e retorna ao mesmo depósito. A quantidade de carga transportada em cada rota é igual $q(d_i)$ e corresponde a soma da demanda de cada cliente na rota de d_i . Igualmente, o custo da rota, representado por $c(d_i)$, é dado pelo custo de localizar o depósito d_i mais o custo c_{ir} de viajar de i para r observando toda a rota. Para o caso de um depósito fechado, tem-se $q(d_i)=0$ e $c(d_i)=0$.

Jarboui et al. (2013) definiram a aptidão (ou custo) da rota d_i como sendo: $f(d_i) = c(d_i) + \alpha_j \max\{0, q(d_i) - b_j\}$, em que α_j representa o fator de penalidade a ser aplicado quando se excede a capacidade de estoque b_j do depósito. Por fim, chega-se na função de aptidão $f(s)$ calculada como: $f(s) = \sum_{j=1}^m f(d_j)$.

2.2 Heurística Para o PLR

A heurística desenvolvida para o PLR consiste no algoritmo BVV da Figura 3, em que a fase de busca local é o algoritmo DVV da Figura 8. O algoritmo geral envolve a inicialização com uma solução gerada de forma aleatória.

Uma solução é gerada aleatoriamente efetuando os seguintes passos: (i) gera-se uma chave aleatória inteira para cada cliente; (ii) seleciona-se um depósito d_i de forma aleatória; (iii) seleciona-se o cliente de menor chave e atribui-o no depósito d_i ; (iv) o passo (iii) é repetido até que a capacidade de d_i não seja violada; (v) se a capacidade de d_i for violada, volta-se ao passo (ii). Clientes que já foram atribuídos a algum depósito têm sua chave atualizada para infinito.

O algoritmo BVV executa até alcançar a condição de parada, que pode ser um número máximo de iterações ou tempo. Nesse trabalho, foi utilizado como condição de parada, o número máximo de iterações.

3. Testes Numéricos

Como mencionado anteriormente, a heurística foi codificada na linguagem C. A estrutura principal usou a heurística de Busca em Vizinhança Variável com a busca local sendo realizada pelo o método de Descida em Vizinhança Variável.

Foram consideradas 79 instâncias de três autores diferentes, sendo 13 instâncias fornecidas em Barreto (2004), 30 instâncias de Prins et al. (2006) e 36 instâncias de Tuzun e Burke (1999). Cada instância fornece o número de clientes, o número de depósitos, as coordenadas de cada depósito, as coordenadas de cada cliente, a capacidade de carga do veículo e a capacidade de estoque do depósito. Elas podem ser obtidas em http://prodhonc.free.fr/Instances/instances_us.htm.

Os testes computacionais ocorreram em um computador com processador Intel® Core™ i3 1.90 GHz, 4GB de memória RAM e sistema operacional Linux Ubuntu 12.04. Limitou-se o algoritmo com o número máximo de 10.000 iterações, obtido por meio de testes de calibração.

3.1 Resultados

As Tabelas 1, 2 e 3 possuem os resultados para as instâncias de Barreto (2004), Prins et al. (2006) e Tuzun e Burke (1999), respectivamente. Nessas tabelas estão representados o Valor da Solução de cada instância obtida com a heurística, o número de Veículos Usados, o Tempo gasto em segundos, o valor da Melhor Solução conhecida, obtido em Jarboui et al. (2013), e o Desempenho da heurística comparado com a melhor solução, em porcentagem.

Inicialmente, foi assumido que cada depósito possuía apenas um veículo associado. Porém, como foi informada a capacidade dos veículos, decidiu-se obter o número de veículos requeridos para atender aquela instância. Para saber quantos veículos é preciso em cada instância, dividiu-se a demanda atendida por cada depósito pela capacidade do veículo, que são todos de mesma capacidade, ou seja, a frota é considerada homogênea.

O cálculo do desempenho da heurística em comparação com a melhor solução conhecida foi feita como: $100 - (((\text{Valor da Solução} - \text{Melhor Solução conhecida}) \div \text{Melhor Solução conhecida}) \times 100)$.

A Tabela 1 traz os testes e resultados sobre as instâncias de Barreto (2004). Observou-se que cinco instâncias obtiveram valor igual ao da Melhor Solução conhecida e apenas três instâncias tiveram desempenho menor do que 90%, sendo que elas possuem maior número de clientes e de depósitos. O pior desempenho ocorreu para a instância Min92-(134x5) com desempenho de 78,17 % para a heurística. O tempo gasto na resolução das instâncias variou entre 3 a 74 segundos, pelo qual os maiores tempos correspondem às instâncias com maior número de clientes e depósitos.

Tabela 1. Solução das instâncias de Barreto (2004).

Instâncias- (n x m)	Veículos Usados	Tempo (s)	Valor da Solução	Melhor Solução conhecida	Desempenho (%)
Christofides69-(50x5)	5	12	461,98	450,90	97,54
Christofides69-(50x5)	9	26	580,92	568,29	97,78
Christofides69- (100x10)	8	42	709,82	677,47	95,22
Daskin95-(88x8)	6	32	338,76	296,82	85,87
Daskin95-(150x10)	11	74	44.189,22	40.033,25	89,62
Gaskell67-(21x5)	5	3	370,73	370,73	100,00
Gaskell67-(22x5)	3	4	519,96	519,96	100,00
Gaskell67-(29x5)	3	5	434,26	434,26	100,00
Gaskell67-(32x5)	4	6	441,22	441,22	100,00
Gaskell67-(32x5) bis	3	6	441,22	441,22	100,00
Gaskell67-(36x5)	4	7	426,90	410,00	95,88
Min92-(27x5)	4	5	2.496,70	2.484,04	99,49
Min92-(134x5)	11	59	5.983,34	4.911,32	78,17
Total:	76	281	-	Desempenho Médio:	95,35

Na Tabela 2 estão as instâncias de Prins et al. (2006) com os devidos resultados obtidos. Elas possuem 20, 50, 100 e 200 clientes divididos na quantidade de 5 e 10 depósitos. Os valores da solução obtidos foram satisfatórios para 25 das 30 instâncias, com o desempenho chegando a 100% em duas instâncias. Cinco instâncias apresentaram desempenho inferior a 90%, sendo que quatro destas possuem 100 clientes e 10 depósitos. O tempo gasto variou de 3 a 128 segundos, correspondendo a menores tempos para instâncias com poucos clientes e depósitos.

Tabela 2. Solução das instâncias de Prins et al. (2006).

Instâncias (n x m)	Veículos Usados	Tempo (s)	Valor da Solução	Melhor Solução Conhecida	Desempenho (%)
20-5-1 (20x5)	5	3	21.367,00	21.360,00	99,97
20-5-1b (20x5)	3	3	15.662,00	15.662,00	100,00
20-5-2 (20x5)	5	3	22.929,00	14.708,00	44,11
20-5-2b (20x5)	3	4	14.095,00	14.095,00	100,00
50-5-1 (50x5)	11	12	15.136,00	15.076,00	99,60
50-5-1b (50x5)	6	12	15.136,00	15.077,00	99,61
50-5-2 (50x5)	12	11	29.574,00	29.562,00	99,96
50-5-2b (50x5)	7	11	29.574,00	29.562,00	99,96
50-5-2bbis (50x5)	12	11	16.744,00	17.889,00	99,94
50-5-3 (50x5)	6	11	17.903,00	11.002,00	99,92
50-5-3b (50x5)	11	12	11.014,00	11.002,00	99,89
100-5-1 (100x5)	6	12	11.014,00	133.254,00	99,89
100-5-1b (100x5)	23	36	133.329,00	133.254,00	99,94
100-5-2 (100x5)	12	35	133.329,00	97.032,00	99,94
100-5-2b (100x5)	23	39	97.069,00	97.029,00	99,96
100-5-3 (100x5)	11	39	97.069,00	86.610,00	99,96
100-5-3b (100x5)	23	37	86.662,00	86.610,00	99,94
100-10-1 (100x10)	12	37	86.662,00	155.407,00	99,94
100-10-1b(100x10)	24	35	196.785,00	155.412,00	73,37
100-10-2 (100x10)	14	36	196.785,00	141.414,00	73,38
100-10-2b(100x10)	24	35	189.083,00	141.404,00	66,29
100-10-3 (100x10)	12	35	189.083,00	136.499,00	66,28
100-10-3b(100x10)	23	37	139.864,00	136.531,00	97,53
200-10-1 (200x10)	12	38	139.864,00	236.756,00	97,56
200-10-1b(200x10)	45	127	236.990,00	236.764,00	99,90
200-10-2 (200x10)	22	127	236.990,00	276.903,00	99,90
200-10-2b(200x10)	45	128	277.124,00	276.902,00	99,92
200-10-3 (200x10)	22	128	277.124,00	235.110,00	99,92
200-10-3b(200x10)	45	127	235.295,00	235.119,00	99,92
Total:	501	1308	-	Desempenho Médio:	93,88

A Tabela 3 apresenta as instâncias de Tuzun e Burke (1999) bem como os resultados da heurística. Elas possuem 100, 150 e 200 clientes (11, 13 e 12, respectivamente) divididos na quantidade de 10 e 20 depósitos (12 e 22, respectivamente). Por serem instâncias maiores, com mais clientes e depósitos comparado as outras, a heurística teve menor desempenho, variando de 64,36% a 91,68%. Apenas uma instância, a 113122 (100x20), apresentou desempenho superior a 90%. Dessa forma, a heurística foi eficaz na resolução das instâncias, retornando uma boa solução para a maioria dos casos e gastando um tempo computacional relativamente pequeno. Por outro lado, várias instâncias ainda tiveram solução com diferença da Melhor Solução conhecida superior a 20%.

Tabela 3. Solução das instâncias de Tuzun e Burke (1999).

Instâncias (n x m)	Veículos Usados	Tempo (s)	Valor da Solução	Melhor Solução Conhecida	Desempenho (%)
111112 (100x10)	11	42	974,05	881,00	89,44
111122 (100x20)	10	49	906,48	817,00	89,05
111212 (100x10)	10	43	869,59	785,00	89,22
112122 (100x20)	10	47	659,22	574,00	85,15
112212 (100x10)	11	41	414,74	359,00	84,47
112222 (100x20)	10	48	425,54	367,00	84,05
113112 (100x10)	10	41	654,79	563,00	83,70
113122(100x20)	11	48	649,91	600,00	91,68
113212 (100x10)	10	41	495,19	446,00	88,97
113222 (100x20)	10	48	504,14	449,00	87,72
121112 (200x10)	20	147	1.347,18	1.055,00	72,31
121122 (200x20)	21	160	1.321,47	1.045,00	73,54
121212 (200x10)	20	148	1.331,97	1.103,00	79,24
121222 (200x20)	20	160	1.373,99	1.061,00	70,50
122112 (200x10)	21	142	941,57	751,00	74,62
122122 (200x20)	20	156	1.055,25	778,00	64,36
122212 (200x10)	20	143	632,70	491,00	71,14
122222 (200x20)	21	157	627,45	473,00	67,35
123112 (200x10)	21	142	952,94	758,00	74,28
123122 (200x20)	20	153	1.061,60	819,00	70,38
123212 (200x10)	21	142	642,83	483,00	66,91
123222 (200x20)	21	153	686,35	511,00	65,68
131112 (300x10)	15	86	1.170,48	1.021,00	85,36
131122 (300x20)	15	96	1.129,96	963,00	82,66
131212(300x20)	16	87	1.122,59	959,00	82,94
131222 (300x20)	15	96	1.170,37	968,00	79,09
132112 (300x10)	15	84	766,85	632,00	78,66
132122 (300x20)	15	94	792,57	649,00	77,88
132212 (300x10)	15	84	510,98	403,00	73,21
132222 (300x20)	16	93	466,41	363,00	71,51
133112 (300x10)	15	83	812,65	686,00	81,54
133122 (300x20)	15	93	706,82	611,00	84,32
133212 (300x10)	16	83	531,48	428,00	75,82
133222 (300x20)	15	93	576,19	484,00	80,95
Total:	554	3413	-	Desempenho Médio:	79,07

4. Conclusão e Trabalhos Futuros

A heurística desenvolvida para o Problema de Localização e Roteamento resume-se em: obter uma solução inicial aleatória para, em seguida, realizar a fase de agitação, em que uma solução s' é computada a partir de uma estrutura de vizinhança definida. Depois, com busca local, utiliza-se o algoritmo de Descida em Vizinhança Variável para a obtenção de uma nova solução. Aceita-se a nova solução da busca local somente se ela for melhor do que a melhor solução atual. Como o problema em questão envolve a minimização de custos, deseja-se a solução de mínimo valor. A fase de agitação e a busca local são repetidas até atingir a condição de parada de 10.000 iterações.

Testes foram realizados com três grupos de instâncias, isto é, dos autores Barreto (2004), Prins et al. (2006) e Tuzun e Burke (1999). Observou-se bons resultados da heurística para instâncias com poucos clientes e depósitos. Por meio dos testes, o desempenho médio foi de 95,35% para as instâncias de Barreto (2004), 93,88% para Prins et al. (2006) e 79,07% para Tuzun e Burke (1999). O pior desempenho ocorreu para as instâncias maiores (com mais clientes e possibilidades para localizar depósitos), melhorando à medida que as instâncias ficam menores. Vale destacar que a heurística encontrou solução igual a melhor conhecida para 7 das 79 instâncias testadas, mostrando que ela pode ser aperfeiçoada para lidar com os casos maiores.

Por fim, os operadores e a heurística desenvolvida são consistentes para o problema em estudo, apesar de que melhorias podem ser feitas. Trabalhos futuros devem incluir o aperfeiçoamento nos operadores de vizinhança e novas estratégias para resolver instâncias maiores a um tempo computacional aceitável, ou seja: criar um operador que generaliza P_2 , isto é, que considera a troca de duas sequências de clientes, tanto na intra-rota quanto na inter-rota; criar um operador similar ao P_5 ; e, combinar o uso da heurística com a resolução de um modelo de programação inteira, no intuito de desenvolver uma math-heurística;

5. Agradecimentos

Os autores agradecem o apoio financeiro recebido das agências CNPq (471351/2012-1), FAPEG e FAPESP (2014/16906-1).

Variable Neighborhood Search for Location and Routing

Abstract: This research presents a variable neighborhood search based heuristic for the location-routing problem. The heuristic has an initial phase to generate a random solution, that is further improved by a local search procedure. In the local search, five neighborhood structures are applied, each one with operations based on swaps, insertions, or exclusions of customers/depots from routes and depots already established. The heuristic was coded in the C language considering the capacitated version of this problem. Computational experiments were performed over instances from other authors, validating the results compared to other proposals from the literature.

Keywords: Location-Routing Problem; Variable Neighborhood Search; Heuristic.

Referências bibliográficas

- ALBAREDA-SAMBOIA, M.; DIAZ, J.; FERNADEZ, E. **A compact model and tight bounds for a combined location-routing problem.** Computers and Operations Research, v. 32, n. 3, p. 407-428, 2005.
- BARRETO, S. S. **Análise e Modelização de Problemas de Localização-Distribuição.** 2004. 357 f. Tese de Doutorado – Universidade de Aveiro, Aveiro, 2004.
- BELENGUER, J. M. et al. **A Branch-and-Cut method for the Capacitated Location-Routing Problem.** . Computers and Operations Research, v. 38, n. 6, p. 931–941, 2011.
- DERBEL, H. et al. **An iterated local search for solving a location-routing problem.** Electronic Notes in Discrete Mathematics, v. 36, n. 1, p. 875-882, 2010.
- GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the theory of NP-Completeness.** San Francisco: Freeman, 1979.
- GONÇALVES, R. F.; QUEIROZ, T. A. **The Knapsack Problem with Three Practical Constraints.** Procedia Computer Science, v. 29, p. 2192-2200, 2014.

JARBOUI, B. et al. **Variable neighborhood search for location routing.** *Computers and Operations Research*, v. 40, n. 1, p. 47-57, 2013.

LAPORTE, G.; NORBERT, Y. **An exact algorithm for minimizing routing and operating costs in depot location.** *European Journal of Operational Research*, v. 6, n. 2, p. 224-226, 1981.

LAPORTE, G.; NORBERT, Y.; ARPIN, D. **An exact algorithm for solving a capacitated location-routing problem.** *Annals of Operations Research*, v. 6, n. 9, p. 293-310, 1986.

MLADENOVI, N.; HANSEN, P. **Variable neighborhood search.** *Computers and Operations Research*, v. 24, n. 11, p.1097-1100, 1997.

PRINS, C.; PRODHON, C. CALVO, R. W. **Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking.** *4OR: A Quarterly Journal of Operations Research*, v. 4, n. 3, p. 221–238, 2006.

QUEIROZ, T. A.; MIYAZAWA, F. K. **Two-dimensional strip packing problem with load balancing, load bearing and multi-drop constraints.** *International Journal of Production Economics*, v. 145, p. 511-530, 2013.

QUEIROZ, T. A.; MIYAZAWA, F. K. **Order and static stability into the strip packing problem.** *Annals of Operation Research*, In press, 2014, DOI: 10.1007/s10479-014-1634-2.

TUZUN, D.; BURKE, L. I. **A two-phase tabu search approach to the location routing problem.** *European Journal of Operational Research*, v. 116, n.1, p. 87-99, 1999.