

DESENVOLVIMENTO E APLICAÇÃO DE ALGORITMOS ADAPTATIVOS DE BUSCA TABU PARA A RESOLUÇÃO DE PROBLEMAS DE ROTEAMENTO DE VEÍCULOS PERIÓDICO (PRVP)

Renato Hallal

Vitória M. Pureza

Universidade Federal de São Carlos
Departamento de Engenharia de Produção

RESUMO

Esta pesquisa consiste no desenvolvimento de algoritmos para resolver o Problema de Roteamento de Veículos Periódico (PRVP), um problema pouco estudado. O objetivo do PRVP é elaborar um conjunto de rotas para atender a demanda dos clientes ao longo de um horizonte de planejamento. Cada cliente requer que as visitas aconteçam em uma combinação predefinida de dias. Inicialmente, utiliza-se uma heurística de construção inicial para atribuir os clientes aos dias. Em seguida, são realizadas trocas de combinações de dias de visitas na tentativa de melhorar a solução. Os processos de busca por soluções são controlados por uma heurística tabu adaptativa que determina as ações de intensificação e diversificação para cada dia do período. Posteriormente, aplica-se uma abordagem similar à heurística tabu adaptativa para o período como um todo. Testes computacionais envolvendo dez problemas foram realizados e os resultados foram comparados com cinco heurísticas competitivas encontradas na literatura sobre o assunto. Os resultados comprovam que o método proposto leva a bons resultados.

ABSTRACT

This research consists of the development of algorithms to solve the Periodic Vehicle Routing Problem (PVRP), a problem little studied. The objective of the PVRP is to elaborate a set of routes to attend to the customers' demand along a planning horizon. Each customer requests that the visits happen in a predefined combination of days. The proposed approach uses a initial construction heuristic to attribute the customers to days. After that, visit day combination are interchanged aiming to improve the solution. The search process is controlled by an adaptive tabu search, that determines intensification and diversification actions, for each day in the period. Later, it is applied a similar adaptive tabu search for the period as a whole. Computational tests involving ten problems were accomplished and the results were compared with five competitive heuristics found in the literature. The results prove that the method proposed lead to good results.

1. INTRODUÇÃO

Nos últimos anos, o nível de competição entre as empresas aumentou bastante. Os competidores, antes regionais, agora se encontram dispersos em mercados espalhados globalmente. O ambiente de competição empresarial passou a estar sujeito a incertezas crescentes, fazendo com que as empresas buscassem novas formas de sobrevivência, procurando tecnologias e técnicas capazes de mantê-las produzindo e obtendo lucros. Surge então a necessidade de uma reformulação empresarial e da busca por soluções que otimizem as várias funções de sistemas produtivos. Em especial, os consumidores esperam por produtos e serviços que satisfaçam suas expectativas de qualidade, tempo de entrega e custo (Sipper e Bulfin, 1997). A garantia de um nível de serviço diferenciado é hoje o objetivo de empresas que desejam permanecer no mercado e, diante disso, os empresários têm que revisar os setores das suas empresas para identificar novas oportunidades de reduzir custos e manter a competitividade. Dentre os vários setores, a área de distribuição de produtos finais recebeu atenção considerável.

O transporte a partir dos centros de distribuição até os consumidores (chamado de transporte local), usualmente é o mais caro da cadeia de distribuição (Galvão, 2001). Para que a empresa tenha êxito na realização desse transporte, ela deve planejar e executar as atividades de

distribuição de uma maneira racional, levando em consideração todos os fatores econômicos envolvidos. O gasto relacionado ao transporte rodoviário tem se tornado muito aparente devido ao aumento contínuo dos custos do combustível, alto custo do capital para a substituição dos veículos, manutenção, aumento do salário do trabalhador, entre outros fatores. Assim, uma economia pequena, tal como 2% a 5% destas despesas, pode resultar em ganhos substanciais no decorrer dos anos e permitir uma maior flexibilização dos preços finais praticados (Golden e Wasil, 1987). Como decorrência, maior competitividade pode ser alcançada. A busca da minimização dos custos logísticos nas atividades de distribuição passou a ser um objeto de intensa pesquisa e de grande interesse pelas empresas. Neste sentido, o tema central da presente pesquisa são os problemas de roteamento de veículos.

1.1. Resolução do PRV

A classe mais simples de problemas de roteamento com múltiplos veículos é o clássico Problema de Roteamento de Veículos (PRV). No PRV, admite-se a existência de um único depósito onde as rotas começam e terminam. Considera-se a frota homogênea e admite-se que informações relativas aos clientes (posições geográficas e demandas) sejam conhecidas. Cada veículo estabelece uma única rota e cada cliente tem sua demanda atendida por um único veículo, cuja capacidade total não pode ser excedida. O objetivo é minimizar a distância total percorrida sobre todo o roteamento em um único dia. Modelos e algoritmos para resolver o PRV podem ser pesquisados em Bodin *et al.* (1983). Um número substancial de métodos exatos foi elaborado com vistas à resolução do PRV. Levantamentos detalhados a este respeito podem ser encontrados em Christofides *et al.* (1979), Watson-Gandy e Foulds (1981), Bott e Ballou (1996), entre outros. Os algoritmos exatos apesar da qualidade da solução apresentada, não conseguem, entretanto, tratar PRVs de interesse prático devido à evolução exponencial do tempo computacional requerido em relação ao aumento do tamanho do problema, uma vez que se trata de um problema *NP-hard*. De forma a dar suporte a este tipo de problema surgem as heurísticas e meta-heurísticas. Veja, por exemplo, Pureza (1991), Laporte (1992), Osman (1993), Taillard (1993), Ficher (1995), Laporte e Osman (1995), Gendreau e Braysy (2001), Czech e Czarnas (2002), entre outros.

1.2. Resolução do PRVP

O Problema de Roteamento de Veículos Periódico é uma generalização do Problema de Roteamento de Veículos onde o período de planejamento é estendido de um único dia para T dias. Nestes problemas os clientes são atendidos com uma dada frequência (F_i) no período T . Neste período, os clientes não só especificam quantas vezes devem ser visitados, mas em geral estipulam as possíveis combinações (C_i) de dias de visitas. Por exemplo, suponha um período de $T = 5$ dias e que o cliente i tenha $F_i = 2$ e $C_i = \{(1,3), (2,4), (3,5)\}$. Então, este cliente deve ser visitado 2 vezes no período de 5 dias e estas visitas devem acontecer nos dias 1 e 3, ou nos dias 2 e 4, ou nos dias 3 e 5. Assim, o objetivo do PRVP é selecionar as combinações de dias de visitas para cada cliente e a ordem com que cada visita deve ser efetuada em cada dia t , de forma a minimizar os custos totais do roteamento sobre todo o período T .

Como se trata de uma generalização do PRV, considera-se o PRVP tão difícil quanto o primeiro, sendo classificado como um problema de otimização combinatória pertencente à classe *NP-hard*. Por esta razão, a maioria dos trabalhos reportados sobre o PRVP utilizam métodos heurísticos para sua resolução.

Uma das motivações para a escolha desta pesquisa é o fato de que, apesar de sua evidente relevância prática, o PRVP tem sido relativamente pouco estudado, o que se reflete no reduzido número de trabalhos publicados nas últimas décadas.

Os Problemas de Roteamento de Veículos Periódico são naturalmente encontrados em empresas de distribuição de refrigerantes, leite, gás industrial, coleta de lixo, conforme Ball (1988), Dror e Ball (1987), Carter *et al.* (1996), Golden e Wasil (1987), Beltrami e Bodin (1974) e Russell e Igo (1979). A primeira aplicação reportada para o PRVP é devida a Beltrami e Bodin (1974) que abordam o problema prático de coleta de lixo em Nova York. Algoritmos mais elaborados foram posteriormente desenvolvidos por Christofides e Beasley (1984), Tan e Beasley (1984), Russell e Gribbin (1991), Gaudioso e Palleta (1992) e Chao *et al.* (1995). Todos esses autores utilizaram programação linear inteira para atribuir uma combinação de visita para cada cliente. Eles então, resolveram o PRV para cada dia do período utilizando um algoritmo modificado de Clarke e Wright (1964). Melhoramentos locais foram obtidos usando a abordagem *record-to-record* de Dueck (1993) e o algoritmo de troca 2-opt (Lin, 1965). Finalmente, um processo de reinicialização foi utilizado para diversificar a busca. O melhor método reportado foi proposto por Cordeau *et al.* (1997) que utilizaram uma heurística de busca tabu capaz de resolver três tipos de problemas de roteamento: o Problema do Caixeiro Viajante Periódico, o Problema de Roteamento de Veículos Multi-Depósitos e o Problema de Roteamento de Veículos Periódico.

Diferentemente das heurísticas já apresentadas, ao invés de utilizar uma versão modificada do algoritmo de Clarke e Wright (1964), utilizou-se neste trabalho o algoritmo HTA para resolver o PRV para cada dia do período. Além disto, para controlar o processo de busca utilizou-se uma idéia similar aos procedimentos do algoritmo HTA, no qual faz-se uma análise dos custos do roteamento referente a todo o período.

Esta abordagem levou a soluções de alta qualidade, sendo comparável com cinco heurísticas competitivas encontradas na literatura sobre o tema. Em alguns casos, os resultados mostraram-se compatíveis com a melhor heurística existente, encontrando soluções ligeiramente inferiores.

2. APLICAÇÃO DO ALGORITMO HTA AO PRVP

O algoritmo HTA, desenvolvido por Pura e França (2001), é composto de mecanismos de busca tabu que estão internamente adaptados aos seus movimentos de troca (veja Dror e Levy, 1986) e aos movimentos 2-opt. Estes mecanismos proíbem estrategicamente o movimento de algumas arestas, permitindo maior variabilidade nas soluções. Escolheu-se o algoritmo HTA para o roteamento porque apresenta ganhos não desprezíveis quanto à qualidade da solução. No HTA, para diversificar a busca por soluções em uma certa região do espaço de solução, são aplicadas perturbações na trajetória da busca. Estas perturbações são aplicadas depois de uma análise e identificação dos padrões da trajetória. Tais análises são realizadas ao longo de todo o processo de busca em intervalos específicos, com as perturbações sendo aplicadas por um certo número de iterações.

O algoritmo desenvolvido, *Period Vehicles Routing Problem - Controlled Search* (PVRP-CS), possui as seguintes fases:

1. *Construção da solução inicial.*
2. *Realização de movimentos de melhoria.* Realizam-se movimentos de troca de clientes entre dias, para minimizar o custo da solução inicial. Após a aplicação de cada movimento, o algoritmo HTA é aplicado para formar as rotas dos dias envolvidos.
3. *Intensificação e diversificação.* Com base no padrão de trajetória, obtido através do acompanhamento dos valores das soluções intermediárias, permite-se uma maior intensificação ou diversificação da busca por novas soluções.
4. *Critério de parada.* Atingindo o número máximo de reinicializações, pare. Caso contrário, volte ao passo 2.

2.1. Construção da Solução Inicial

Foi elaborado o seguinte procedimento:

1. Inicialmente, atribuem-se todos os clientes ao mesmo dia. Nesta etapa ainda não é especificada uma combinação de visita para cada cliente.
2. Aplica-se o algoritmo HTA, obtendo-se um conjunto de rotas como se todos os clientes fossem atendidos no mesmo dia.
3. Atribui-se, então, cada uma das rotas formadas aos T dias do período.
4. Aplica-se o algoritmo HTA a cada dia t para gerar a solução inicial.

2.1.1. Atribuição das Rotas aos T dias do período.

Para exemplificar a atribuição das rotas iniciais aos T dias do período considere a figura abaixo.

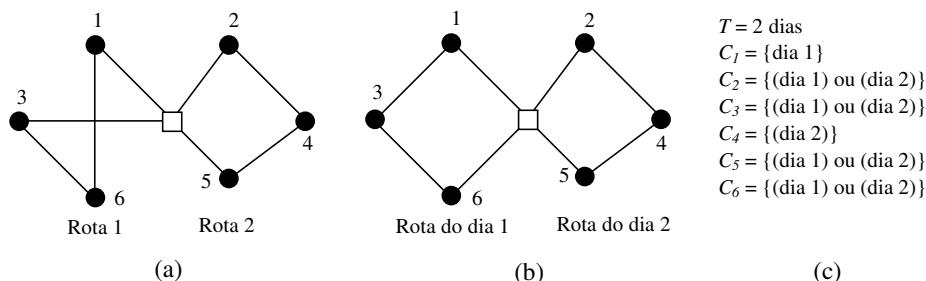


Figura 1: (a) Rotas geradas ao aplicar o algoritmo HTA, (b) Clientes atribuídos aos dias e posterior roteamento e (c) Possíveis combinações de visitas para os clientes no período.

Depois da aplicação do HTA (procedimento 2), obtém-se um roteamento com as formas apresentadas na Figura 1 (a). Para atribuir as rotas aos dias, atribui-se cada cliente da rota 1 ao dia 1 (se for possível) e cada cliente da rota 2 ao dia 2 (se for possível). Com isso, especifica-se o dia em que o cliente será visitado, atribuindo-lhe a respectiva combinação. Portanto, seguem-se os seguintes passos:

- I. Analisa-se se os clientes 1, 3 e 6 da rota 1 (Figura 1 (a)), podem ser atendidos no dia 1. Para isso:

- Verifica-se na combinação C_1 (Figura 1 (c)) se o cliente 1 pode ser atendido no dia 1. Se for possível escolhe-se a respectiva combinação, caso contrário outra combinação deve ser escolhida aleatoriamente.
- Do mesmo modo, verifica-se na combinação C_3 se o cliente 3 pode ser atendido no dia 1. Se for possível escolhe-se a respectiva combinação, caso contrário outra combinação deve ser escolhida aleatoriamente.
- Verifica-se na combinação C_6 se o cliente 6 pode ser atendido no dia 1. Se for possível escolhe-se a respectiva combinação, caso contrário outra combinação deve ser escolhida aleatoriamente.

II. Analisa-se de forma análoga se os clientes 2, 4 e 5 da rota 2 (Figura 1 (a)), podem ser atendidos no dia 2. Caso o número de rotas geradas na Figura 1 (a) for maior que o tamanho do período (no exemplo $T = 2$), atribua a próxima rota ao dia 1 novamente e repita o processo do passo I.

Ao final deste processo, têm-se os clientes 1, 3 e 6 atribuídos ao dia 1 e os clientes 2, 4 e 5 atribuídos ao dia 2. Aplica-se o algoritmo HTA a cada dia t gerando a solução inicial (como exemplo, veja Figura 1 (b)).

2.2. Movimentos de Melhoria

Para melhorar a solução gerada anteriormente, são aplicados movimentos de troca e inserção de nós em rotas e movimentos chamados de *clean-up*.

2.2.1. Troca e Inserção

Neste tipo de movimento trocam-se as combinações de dias de visitas atribuídas aos clientes. Esta heurística, baseada na heurística de Christofides e Beasley (1984), foi utilizada porque realiza tanto movimentos de troca de clientes entre dias, quanto movimentos de inserção de clientes em rotas.

Nesta heurística, pequenos subconjuntos de clientes são escolhidos e todas as possíveis combinações de dias de visitas para cada subconjunto são enumeradas. Forma-se assim uma família U de subconjuntos de clientes. Cada subconjunto de clientes da família U é a união de dois outros conjuntos de clientes (M_1 e M_2). M_1 é um conjunto contendo T_1 clientes que aparecem consecutivamente na rota de algum dia t , e M_2 é um conjunto contendo T_2 clientes próximos geograficamente do conjunto M_1 , sendo necessariamente os clientes do conjunto M_2 pertencentes a uma rota de outro dia. A união do conjunto M_1 com o conjunto M_2 ($M_1 \cup M_2$) constitui o primeiro subconjunto de clientes da família. Uma família U completa é gerada considerando todos os dias T e todos os clientes i da rota do dia t . Cada subconjunto de clientes $M_1 \cup M_2$ formado, constitui uma iteração neste algoritmo.

2.2.2. Movimento *clean-up*

Na aplicação parcial dos movimentos *clean-up* (Chao *et al.*, 1995) consideram-se mudanças de combinações de visitas dos clientes que requerem somente uma visita no período, ou seja, clientes com frequência de visita igual a um.

Primeiramente, estes clientes são armazenados em uma lista C . Toma-se o primeiro cliente de C e calcula-se o custo de atribuí-lo ao dia d . Se o custo for menor que o da solução corrente, o cliente é atribuído ao dia d , caso contrário, este é mantido no mesmo dia. O procedimento é aplicado para todos os clientes da lista C . Após a efetivação de um movimento, o algoritmo HTA é aplicado às rotas dos dias envolvidos.

2.3. Intensificação e Diversificação

Na tentativa de melhorar a solução e o tempo computacional do algoritmo foi utilizada uma idéia similar ao algoritmo HTA integrando estratégias de intensificação e diversificação. Assim, o algoritmo compõe-se de uma lista cíclica LC , que guarda o histórico dos custos gerados nas últimas it iterações. Estes custos são analisados aproximando-os por uma reta através do Método dos Mínimos Quadrados (Ruggiero e Lopes 1988). Através dessa reta classifica-se a trajetória das soluções em: *trajetória estagnada*, *trajetória ascendente* e *trajetória descendente*. Dependendo do padrão da trajetória, tomam-se algumas providências para as iterações seguintes: informa-se o momento da próxima avaliação (Av) e dependendo da trajetória, aplica-se o movimento *clean-up* como forma de intensificar a busca, ou se permite que o algoritmo aceite apenas soluções ruins, diversificando a busca. Para isto foi estabelecido um parâmetro x , de tolerância à piora da solução.

2.3.1. Análise das Soluções

Para cada elemento de U (M_1 U M_2) são geradas todas as possíveis combinações de visitas entre clientes. Para cada uma das possíveis combinações de visitas entre estes clientes obtém-se um custo C , que é o valor da função objetivo, ou seja, o custo total do roteamento no período. Por exemplo, suponha que tanto o conjunto M_1 quanto o conjunto M_2 sejam formados por apenas um cliente e que as possíveis combinações de visitas para estes clientes sejam:

$C_{M1} = \{(\text{dia 1}) \text{ ou } (\text{dia 2})\}$, possíveis combinações de visitas para o cliente do conjunto M_1 .

$C_{M2} = \{(\text{dia 1}) \text{ ou } (\text{dia 2})\}$, possíveis combinações de visitas para o cliente do conjunto M_2 .

Para cada combinação de elementos de C_{M1} e C_{M2} obtém-se um custo C_i , conforme mostrado na Tabela 1.

Tabela 1: Possíveis de trocas combinações de visitas entre os clientes do conjunto M_1 e M_2

C_{M1}	C_{M2}	Custo total do roteamento no período
Dia 1	Dia 1	C_1
Dia 1	Dia 2	C_2
Dia 2	Dia 1	C_3
Dia 2	Dia 2	C_4

Cada custo C_i será comparado com um custo intermediário C_{interm} , inicialmente atualizado com o valor da solução inicial. A medida em que o processo de busca se desenvolve, o valor de C_{interm} é atualizado da seguinte forma:

- Se a trajetória for estagnada o algoritmo permite que o custo intermediário C_{interm} seja atualizado por um C_i que seja $x\%$ pior que o custo intermediário corrente.

- Caso a trajetória for ascendente ou descendente o custo intermediário C_{interm} é atualizado por um C_i que seja melhor que o custo intermediário corrente.

Se algum destes custos C_i for melhor que C^* , o custo da melhor solução factível, atualiza-se a melhor solução. Ao final destas possíveis trocas C_{interm} é armazenado na lista LC . A lista LC será automaticamente preenchida até o momento da avaliação (Av), quando uma análise da trajetória dos custos será realizada. Caso não esteja no momento da avaliação, outro elemento de U é escolhido para a realização de novas trocas, constituindo a iteração $it + 1$ do processo.

Com o objetivo de permitir uma maior flexibilidade na busca por soluções, o procedimento acima permite soluções intermediárias infactíveis. Essa infactibilidade vem de uma relaxação na carga máxima do dia, permitindo-se uma carga de até $w = 20\%$ acima da carga máxima do dia. Essa carga é calculada multiplicando-se a capacidade máxima do veículo pelo número máximo de veículos por dia do período.

2.3.2. Momento da Avaliação (Av) e Classificação do Padrão da Trajetória

Nesse momento, identifica-se o padrão da trajetória utilizando-se o método de interpolação linear conhecido como Método dos Mínimos Quadrados, que ajusta a reta que mais se aproxima dos custos armazenados na lista LC .

O valor do coeficiente angular θ dessa reta, identificará a trajetória. Por exemplo:

Suponha que os custos armazenados em LC , apresentem sua distribuição conforme a Figura 2 (a).

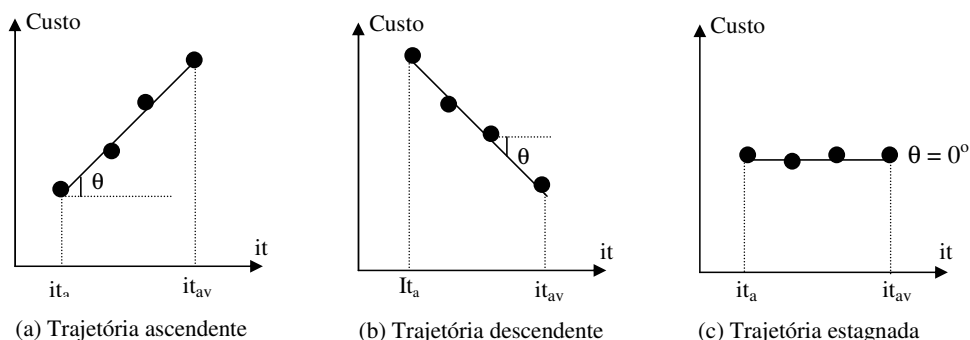


Figura 2: Representação das trajetórias ascendentes, descendentes e estagnadas

Como $\theta > 0,0001$, a trajetória é classificada como *ascendente* e tomam-se as seguintes providências:

- Muda-se o momento da próxima avaliação para o instante atual mais Av iterações;
- Aplica-se o movimento *clean-up* na tentativa de melhorar a solução;
- Atribui-se zero a x (parâmetro de tolerância), ou seja, aceita-se apenas soluções de melhora.

Supondo agora que os custos armazenados em LC apresentem a distribuição conforme a da Figura 2 (b), com $\theta < -0,0001$. Neste caso a trajetória é classificada como *descendente* e tomam-se as seguintes providências:

- Muda-se o momento da próxima avaliação para o instante atual mais Av iterações.
- Aplica-se o movimento *clean-up* na tentativa de melhorar ainda mais a solução, constituindo uma etapa de intensificação.
- Atribui-se zero a x , ou seja, aceita-se apenas soluções de melhoria.

Finalmente, supondo que os custos armazenados em LC apresentem a distribuição conforme a da Figura 2 (c), com $(0,0001 < \theta < -0,0001)$. Neste caso a trajetória é classificada como *estagnada* e, então:

- Muda-se o momento da próxima avaliação para o instante atual mais Av iterações.
- Permite-se que nas próximas it iterações a solução intermediária seja piorada em até $x\%$, constituindo uma etapa de diversificação.

3. RESULTADOS COMPUTACIONAIS

O algoritmo foi implementado em Pascal, compilado em Free Pascal i386 v 1.0.6 e os experimentos realizados em um computador AMD Athlon (TM) XP 2200+. Foram tratados 10 problemas descritos em Christofides e Beasley (1984), cujos tamanhos variam entre 50 a 100 nós (além do depósito), com períodos variando de 2 a 10 dias.

Nos experimentos realizados, ao se usar o algoritmo HTA foi usada uma variação paramétrica de 10 faixas de período tabu e 5 faixas de horizonte de avaliação. Cada faixa tem tamanho igual a 4. A Tabela 2 apresenta os problemas utilizados com os respectivos horizontes de avaliações e faixas de período tabu para a obtenção dos melhores resultados.

Para os problemas com 50 clientes, os melhores resultados foram obtidos na faixa de horizonte de avaliação $[h_{min}, h_{max}] = [4, 8]$. Para problemas com 75 clientes, destaca-se a faixa $[h_{min}, h_{max}] = [10, 15]$. Para os problemas com 100 clientes, não parece haver uma faixa de horizonte superior às demais. Verificou-se que períodos tabu pequenos (menores que 14) foram suficientes para obtenção das melhores soluções.

Tabela 2: Resultados obtidos para os horizontes de avaliações e períodos tabu

Problemas	$[h_{min}, h_{max}]$	$[t_{min}, t_{max}]$
50a	[4, 8]	[4, 8]
50b	[4, 8]	[6, 10]
50c	[4, 8]	[4, 8]
75a	[10, 15]	[6, 10]
75b	[4, 8]	[8, 12]
75c	[10, 15]	[6, 10]
100a	[3, 7]	[10, 14]
100b	[6, 10]	[0, 4]
100c	[10, 15]	[10, 14]
100d	[7, 11]	[8, 12]

Os momentos de avaliação utilizados no decorrer de todo o período são dependentes das trajetórias que controlam o valor da função objetivo, intensificando ou diversificando a busca. Após alguns testes, foram estabelecidos de forma arbitrária, um valor de $Av = 47$ para as trajetórias ascendentes e descendentes e $Av = 4$ para trajetória estagnada. Da mesma forma, permitiu-se uma piora na solução de $x = 2\%$, quando a trajetória estiver estagnada.

Com relação aos movimentos de troca, utilizou-se apenas um cliente no conjunto M_1 e um cliente no conjunto M_2 . Também estipulou-se um acréscimo na demanda do dia de $w = 20\%$, permitindo uma maior variabilidade nas trocas. Como critério de parada utilizaram-se duas reinicializações, sendo satisfatório para a obtenção das soluções.

4. COMPARAÇÃO DOS RESULTADOS COM HEURÍSTICAS DA LITERATURA

Nesta seção é feita uma comparação dos resultados obtidos com cinco heurísticas competitivas encontradas na literatura. A Tabela 3 apresenta os problemas tratados e os custos obtidos pelos autores.

Tabela 3: Comparação dos resultados computacionais

Problemas	C / G / L	C / G / W	C / B	T / B	R / G	PVRP-CS
50a	524.61	524.61	547.4	-	537.3	524.61
50b	1322.87	1322.87	1443.1	1481.3	1355.4	1335.59
50c	524.61	524.61	547.5	-	-	524.61
75a	835.43	840.2	843.9	-	867.8	835.32
75b	2027.99	2046.2	2187.3	2192.5	2141.3	2068.2
75c	836.37	847.2	938.2	-	-	836.36
100a	826.14	831.1	839.2	-	833.6	826.63
100b	2034.15	2043.0	2153.3	2281.8	2108.3	2061.25
100c	826.14	828.3	875.0	-	-	826.63
100d	1595.84	1611.9	1675.0	1833.7	1638.5	1652.88

Em negrito: Melhor solução obtida.

A Tabela 3 refere-se aos seguintes autores com os respectivos computadores utilizados:

- C / G / L: algoritmo de Cordeau *et al.* (1997) - Estação SUN SparcStation 10.
- C / G / W: algoritmo de Chao *et al.* (1995) - Estação SUN 4/ 370.
- C / B: algoritmo de Christofides e Beasley (1984) - Computador CDC 7600.
- T / B: algoritmo de Tan e Beasley (1984) - Computador CDC 7600.
- R / G: algoritmo de Russell e Gribbin (1991) - Computador IBM 20MHZ.
- PVRP-CS: versão proposta deste algoritmo para a resolução do Problema de Roteamento de Veículos Periódico com Busca Controlada.

Quanto à qualidade das soluções, os resultados apresentam-se superiores ao algoritmo de C / B, T / B e R / G. Com relação a C / G / W as soluções apresentam-se de igual tamanho. Comparado com o melhor resultado da literatura (C / G / L), observa-se que foram obtidos excelentes resultados nos problemas 50a, 50c, 75a, 75c, 100a e 100c. Além disso, verifica-se que o algoritmo apresentou bons resultados para os outros problemas. Vale destacar que, no problema 75a foi possível obter uma solução com um custo ligeiramente inferior ao da melhor solução já reportada.

A Tabela 4 apresenta uma comparação percentual com as heurísticas dos respectivos autores mencionados acima:

Tabela 4: Comparação percentual do PVRP-CS com as demais heurísticas

Problemas	C / G / L (%)	C / G / W (%)	C / B (%)	T / B (%)	R / G (%)
50a	0	0	4,16	-	2,36
50b	-0,96	-0,96	7,45	9,84	1,46
50c	0	0	4,18	-	-
75a	0,01	0,58	1,02	-	3,74
75b	-1,98	-1,08	5,45	5,67	3,41
75c	0	1,28	10,85	-	-
100a	-0,06	0,54	1,5	-	0,83
100b	-1,33	-0,94	4,27	9,67	2,23
100c	-0,06	0,2	5,53	-	-
100d	-3,57	-2,54	1,26	9,86	-0,83

Em negrito: refere-se o quanto PVRP-CS foi melhor ou igual a outra heurística

Os custos percentuais (cp) foram obtidos através da Equação 1:

$$cp = 100 \times \left(\frac{(\text{custo .da.outra .heurística}) - (\text{custo .do.PVRP} - CS)}{(\text{custo .da.outra .heurística})} \right) \quad (1)$$

Portanto, quando $cp > 0$ indica o quanto PVRP-CS é melhor que a outra heurística, quando $cp = 0$ indica que PVRP-CS obteve o mesmo resultado da outra heurística e quando $cp < 0$ significa que a outra heurística obteve um resultado melhor que o da PVRP-CS. Observando a tabela, verifica-se que PVRP-CS obteve êxito em 30 dos 41 resultados apresentados, mostrando a eficiência da heurística.

5. CONCLUSÕES

Neste trabalho, foi proposto uma heurística para a resolução do Problema de Roteamento de Veículos Periódico, utilizando o algoritmo HTA para o roteamento. Testes computacionais envolvendo dez problemas foram realizados e os resultados comparados com cinco heurísticas competitivas encontradas na literatura sobre o assunto.

A heurística desenvolvida (PVRP-CS), mostrou-se muito eficaz, quanto a qualidade da solução, em relação às heurísticas existentes, colocando-a entre as duas melhores implementações de que se tem conhecimento. É importante ressaltar que todos os algoritmos envolvidos nestas análises são bastante eficientes. Por esta razão, mesmo pequenas melhorias são dignas de nota. Os resultados indicam que o algoritmo com mecanismos adaptativos de controle dos padrões das trajetórias é capaz de produzir soluções de alta qualidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- Ball, M. (1988) Allocation/Routing: Models and Algorithms. *Vehicles Routing: Methods and Studies* (Golden B., Assad A., Eds.), North-Holland, Amsterdam.
- Beltrami, E. e Bodin, L. (1974) Networks and Vehicle Routing for Municipal Waste Collection. *Networks*, v. 4, p. 65-94.
- Bodin, L.; Golden, B.; Assad, A. e Ball, M. (1983) Routing and Scheduling of Vehicle and Crews: The State of the Art. *Comput. Oper. Res.*, v. 10, n. 2, p. 63-211, Pergamon Press Oxford.

- Bott, K. e Ballou, R. (1996) Research Perspectives in Vehicle Routing and Scheduling. *Transportation Research*, 20 A, p. 239-243.
- Carter, M. W.; Farvolden, J. M.; Laporte, G. e Xu, J. (1996) Solving an Integrated Logistics Problem Arising in Grocery Distribution. *INFOR*, v. 34, p. 290-306.
- Chao, M.; Golden, L. B. e Wasil, E. (1995) An Improved Heuristic for the Period Vehicle Routing Problem. *Networks*, v. 26, p. 25-44.
- Christofides, N. e Beasley, J. E. (1984) The Period Routing Problem. *Networks*, v. 14, p. 237-256.
- Christofides, N.; Mingozzi, A. e Toth P. (1979) The Vehicle Routing. *Combinatorial Optimization*, p. 315-338.
- Clarke, G. e Wright, J. (1964) Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Ops. Res.*, v. 12, p. 568-581.
- Cordeau, J. F.; Gendreau, M. e Laporte, G. (1997) A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems. Center for research on transportation, Technical Report, *Networks*, v. 30, p. 105-119, Montreal, Canada.
- Czech, Z. J. e Czarnas, P. (2002) Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windons. 10th Euramicro Workshop on Parallel, Distributed and Networks based Processing, Canary – Islands – Spain, p. 376-383.
- Dror, M. e Levy, L. (1986) A Vehicle Routing Improvement Algorithm Comparison of a Greedy and a Matching Implementation for Inventory Routing. *Computers and Operations Research*, v. 13, n. 1, p. 33-45.
- Dror, M. e Ball, M. (1987) Inventory/Routing: Reduction from an Annual to a Short-Period Problem. *Naval Res. Log. Q.*, v. 34, p. 891-905.
- Dueck, G. (1993) New Optimization Heuristics: The Great Deluge Algorithm and the Record-to Record Travel. *J. Comput. Phys.*, v. 104, p. 86-92.
- Ficher, M. L. (1995) Vehicle Routing. *Network Routing, Handbooks in Operations Research and Management Science* (Ball, M. O., Magnanti T. L., Monna C. L. and Nemgauser G. L., Eds.), North-Holland, Amsterdam, v. 8, p. 1-33.
- Galvão, A. N. (2001) *Logística e Gerenciamento da Cadeia de Distribuição*. Campus, Rio de Janeiro.
- Gaudio, M. e Paletta, G. (1992) A Heuristic for the Periodic Vehicle Routing Problem. *Trans. Sci.*, v. 26, p. 86-92.
- Gendreau, M. e Braysy, O. (2001) Genetic Algorithms for the Vehicle Routing Problem with Time Windons. *Internal Report STF 42 A 01021, SINTEF Aplied mathematics, departament of optimization, Oslo, Norway*.
- Golden, B. L. e Wasil, E. A. (1987) Computerized Vehicle Routing in the Soft Drink Industry. *Operations Res.*, v. 35, p. 6-17.
- Laporte, G. (1992) The Vehicle Routing Problem: An Overview for Exact and Approximate Algorithms. *Eur. J. Oper. Res.*, v. 59, p. 345-358.
- Laporte, G. e Osman, I. H. (1995) Routing Problems: A Bibliography. *Ann. Oper. Res.*, v. 61, p. 227-262.
- Lin, S. (1965) Computer Solutions of the Traveling Salesman Problem. *Bell Syst. Tech. J.*, v. 44, p. 2245-2269.
- Osman, I. H. (1993) Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem. *Annals of Operations Research*, v. 41, p. 421-451.
- Pureza, V. (1991) *Problema de Roteamento de Veículos via Metaheurística Tabu*, Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP.
- Pureza, V. e França, P. (2001) Uma Abordagem Adaptativa de Busca Tabu Aplicada ao Problema de Roteamento de Veículos. *Revista do Transporte Público*, v. 9, fac. 2, p. 28-47, S. P., Brasil.
- Ruggiero, M. A. G. e Lopes, V. L. R. (1988) *Cálculo numérico: aspectos teóricos e computacionais*, McGraw-Hill, São Paulo.
- Russell, R. A. e Gribbin, D. (1991) A Multiphase Approach to the Period Routing Problem. *Networks*, v. 21, p. 747-745.
- Russell, R. A. e Igo, W. (1979) An Assignment Routing Problem. *Networks*, v. 9, p. 1-17.
- Sipper, D. e Bulfin, J. R. (1997) *Planning Control and Integration*. McGraw-Hill International Edition.
- Taillard, E. (1993) Parallel Iterative Search Methods for Vehicle Routing Problems. *Networks*, v. 23, n. 8.
- Tan, C. C. e Beasley, J. E. (1984) A Heuristic Algorithm for the Period Vehicle Routing Problem. *Omega*, v. 12, n. 5, p. 497-504.
- Watson-Gandy, C. D. e Foulds, L. R. (1981) The Vehicle Scheduling Problem: A Survey. *New Zealand Ops. Res.*, v. 9, p. 73-91.

Renato Hallal (renato_hallal2000@yahoo.com.br)

Vitória M. Pureza (vpureza@power.ufscar.br)

Rua Visconde de Inhaúma, 452 – Centro – São Carlos, SP, Brasil

CEP: 13560-190

Fone: (16) 3371-3950 ou (17) 9791-6468