

MÉTODOS DE PESQUISA EM VIZINHANÇA VARIÁVEL APLICADOS À RESOLUÇÃO DO PROBLEMA DE PROGRAMAÇÃO DIÁRIA DE TRIPULAÇÕES DE ÔNIBUS URBANO

Marcone Jamilson Freitas Souza

Gustavo Peixoto Silva

Sílvia Maria Santana Mapa

Universidade Federal de Ouro Preto

RESUMO

Este trabalho aborda o Problema de Programação Diária de Tripulações (PPT) no Sistema de Transporte Público. Tal problema consiste em atribuir um conjunto de tarefas aos tripulantes de uma dada empresa participante do sistema de forma que todas as viagens das linhas sob responsabilidade desta sejam executadas com o menor custo possível. A solução do PPT é um conjunto de jornadas diárias de trabalho de tripulantes. Neste trabalho, o PPT foi abordado utilizando três variantes do Método de Pesquisa em Vizinhança Variável. Esse método explora o espaço de soluções utilizando diferentes estruturas de vizinhança, as quais modificam as jornadas de trabalho através de operações realizadas com suas tarefas. Cada solução gerada é avaliada por uma função baseada em penalidades que visa atender a legislação trabalhista, as regras operacionais da empresa, assim como melhorar o aproveitamento da mão-de-obra operacional. Os algoritmos foram testados com dados reais de uma empresa que opera na cidade de Belo Horizonte.

ABSTRACT

This work deals with the Bus Crew Scheduling Problem (BCSP) related to a company that operates in the public transportation system. Such problem consists on assigning the set of all vehicle trips under responsibility of a company, to a set of drivers with the minimum operational cost. The solution of BCSP, called a schedule, is a set of driver duties. In this work the BCSP is solved through the Variable Neighborhood Search metaheuristic. This method explores the solutions space using different neighborhoods, which modify the driver duties through changes in the drivers' tasks. Each schedule obtained is evaluated by a penalty function that has as objective to satisfy the labor agreement rules, the operational rules of the company, and to optimize the use of the crew power work as well. The algorithms were tested with real data provided by a company operating in Belo Horizonte city.

1. INTRODUÇÃO

O planejamento de transportes de ônibus urbano é normalmente decomposto em vários subproblemas devido à sua complexidade: Tabela de horários, Programação de veículos, Programação Diária de Tripulações e Programação Mensal de Tripulações. A Figura 1 ilustra a interligação entre esses problemas.

No planejamento do serviço de transportes de uma cidade, o transporte público é dividido em um conjunto de linhas, cada qual normalmente identificada por um número, e que corresponde a um percurso a ser realizado por um ônibus de um ponto a outro da cidade. Para cada linha a respectiva frequência é baseada na demanda. Após isso, uma “Tabela de Horários” é construída, resultando em viagens, às quais são associados um local de início e outro de final e um horário de início e outro de final da viagem.

Conhecidas as viagens a serem executadas pela empresa concessionária do serviço de transporte público, o problema seguinte é o da “Programação dos Veículos”. Neste, o objetivo é fazer o dimensionamento da frota e a alocação dos veículos às viagens, de forma que todas as viagens programadas para as diversas linhas da empresa sejam executadas. Da programação dos veículos resulta um conjunto de viagens realizadas por cada veículo diariamente, desde a sua saída da garagem até o retorno à esta.

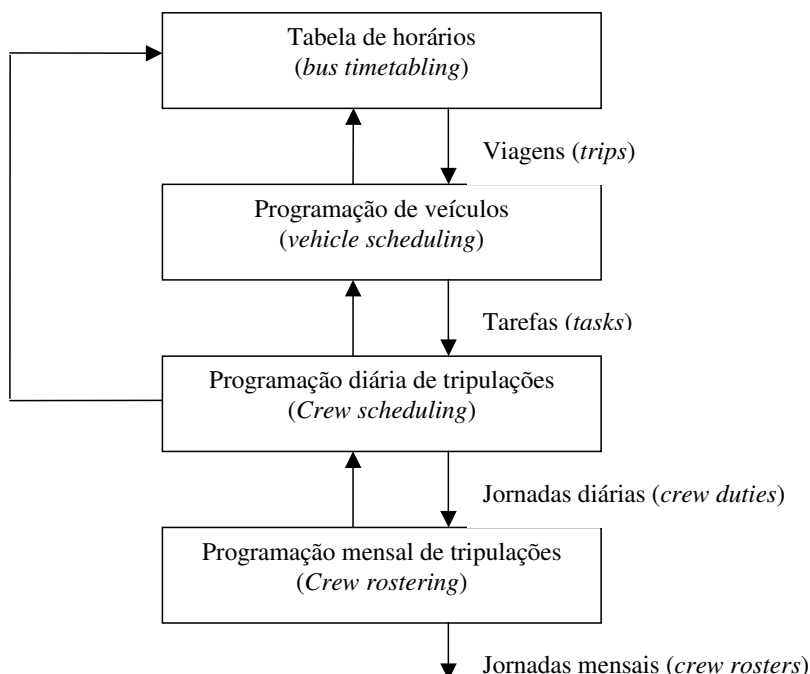


Figura 1: Planejamento de transporte

Para resolver o problema seguinte, o da ‘Programação diária de tripulações’, as viagens de cada veículo, resultantes do problema de programação de veículos, são agrupadas em tarefas. Uma tarefa é um conjunto de viagens a serem realizadas por uma mesma tripulação e que não contém oportunidades para haver troca de tripulações. A partir desse conjunto de tarefas de todos os veículos, são geradas as jornadas diárias das tripulações, resultantes da resolução do problema de programação diária de tripulações. Na geração das jornadas diárias das tripulações, uma série de regras trabalhistas e operacionais da empresa deve ser observada.

Uma vez geradas as jornadas diárias de dias úteis, sábados, domingos e feriados, estas devem ser, agora, combinadas, de forma a fazer a escala mensal das tripulações, problema conhecido na literatura inglesa como *Crew rostering*. Na elaboração da escala mensal, observa-se que uma parte da legislação trabalhista, dos acordos coletivos e das regras operacionais foi contemplada na confecção das jornadas diárias. Entretanto, outras tantas devem ser observadas, como por exemplo: (a) A cada 6 (seis) dias de trabalho o tripulante tem direito a uma folga; (b) Tripulantes que fazem dupla pegada têm folga obrigatoriamente aos sábados ou domingos. Em uma escala mensal procura-se, dentre outros objetivos: (a) minimizar, ou mesmo eliminar, as mudanças de horários de trabalho de uma tripulação; (b) equilibrar as horas mensais trabalhadas pelas tripulações; (c) reduzir as mudanças de linhas de uma tripulação ao longo do mês.

A resolução do PPT é de grande importância, uma vez que nos gastos totais de uma empresa,

a mão-de-obra operacional representa um dos maiores custos (Bouzada, 2002), motivo pelo qual esse tema tem sido largamente estudado. A abordagem mais explorada é aquela que formula o PPT como um problema de recobrimento ou de particionamento e utiliza a técnica de geração de colunas para resolvê-lo (Smith e Wren, 1988; Desrochers e Soumis, 1989; Fores et al., 1999; Barnhart et al., 1998; Friberg e Haase, 1999). A variedade de trabalhos deriva das diferentes maneiras de gerar as colunas e diferentes metodologias para resolver o problema, tais como: *branch-and-bound*, *branch-and-price* e a relaxação lagrangeana.

Dada a NP-completude do problema, os sistemas heurísticos foram os primeiros a serem utilizados na resolução do PPT, os quais consistiam inicialmente apenas na automação do trabalho antes realizado manualmente (Elias, 1964). A principal desvantagem de tais sistemas é a sua incapacidade de detectar possibilidades de otimização. Manington e Wren (1975) iniciaram a inclusão de procedimentos de otimização neste tipo de sistema, desenvolvendo heurísticas de otimização. Embora tais heurísticas não garantissem a obtenção do ótimo global, elas eram capazes de produzir soluções em baixos tempos computacionais, além de permitirem incluir com facilidade qualquer tipo de restrição. Entretanto, tais heurísticas geravam soluções de baixa qualidade e muitas das vezes infactíveis. Isto acontecia porque elas ficavam presas nos primeiros ótimos locais encontrados. Com o surgimento das metaheurísticas, tais como Algoritmos Genéticos (Goldberg, 1989), Busca Tabu (Glover e Laguna, 1997) e *Simulated Annealing* (Kirkpatrick et al., 1983), abriu-se um novo horizonte na resolução do PPT. Embora tais métodos também não garantam a obtenção do ótimo global, eles são providos de mecanismos para escapar de ótimos locais. Na literatura podemos destacar nessa linha os trabalhos de Clement e Wren (1995), Wren e Wren (1995) e Kwan et al. (1999) que utilizam Algoritmos Genéticos, enquanto Shen e Kwan (2001) utilizam Busca Tabu.

Atualmente, as empresas devem atender outros objetivos conflitantes com a redução dos custos, tais como manutenção da qualidade do serviço e manutenção de determinadas características das jornadas de trabalho. Para abordar tal problema, trabalhos recentes fazem uso de otimização multiobjetivo. Lourenço et al. (2001) utilizam as metaheurísticas Busca Tabu e Algoritmo Genético para selecionar soluções não dominadas, segundo o conceito multiobjetivo. As colunas de tais soluções, geradas ao longo do processo, são armazenadas para constituir o domínio sobre o qual é aplicado um método exato para o problema de particionamento, quando se tratar de instâncias de pequeno porte. Para instâncias mais elevadas, é aplicado um método GRASP (Feo e Resende, 1995) desenvolvido pelos autores para esse fim.

Uma revisão dos principais métodos de solução desenvolvidos para o PPT pode ser encontrada em Wren e Rousseau (1995), muitos dos quais publicados em uma série de conferências internacionais sobre o tema (vide, por exemplo, Daduna et al. (1995), Wilson (1999), Voß e Daduna (2001)). Trabalhos que abordam casos brasileiros podem ser vistos em Souza et al. (2004), Silva et al. (2002) e Siqueira (1999), por exemplo. No trabalho de Souza et al. (2004) três métodos são comparados a partir do uso de três estruturas diferentes de vizinhança: SA-RTL (baseado em *Simulated Annealing*), BT-RT (baseado em Busca Tabu) e VNS-RTL (baseado no Método de Pesquisa em Vizinhança Variável). Mostrou-se que o método VNS-RTL foi o mais eficiente, tanto em qualidade de solução final, quanto em rapidez em apresentar uma solução de boa qualidade.

Neste trabalho aborda-se o Problema de Programação Diária de Tripulações através de um modelo heurístico baseado no Método de Pesquisa em Vizinhança Variável. Esse método explora o espaço de soluções utilizando diferentes estruturas de vizinhança, as quais modificam as jornadas de trabalho através de operações realizadas com suas tarefas. Utilizam-se neste trabalho as estruturas de vizinhança propostas em Souza et al. (2004) para desenvolver três variantes desse método, cada qual com uma sequência diferente de exploração de vizinhanças.

Este trabalho está organizado como segue. A seção 2 descreve com mais detalhes o problema abordado. Na seção 3 mostra-se a forma de representação de uma solução do problema, as estruturas de vizinhança desenvolvidas, como uma solução é avaliada e como é gerada a solução inicial dos métodos. A seção 3.5 apresenta o Método de Pesquisa em Vizinhança Variável e as variantes consideradas para resolução do PPT. Os resultados computacionais obtidos são apresentados e analisados na seção 4, enquanto as conclusões se encontram na última seção.

2. DESCRIÇÃO DO PROBLEMA

Como dito anteriormente, no transporte público usualmente a programação da tripulação é feita após a programação dos veículos. Nesta, as viagens são reunidas em *blocos*. Um bloco representa a sequência de viagens que um determinado veículo tem que realizar em um dia, começando e terminando na garagem. Cada bloco mostra também as *Oportunidades de Troca (OT)*. Uma *OT* é um intervalo de tempo suficiente, em um ponto apropriado, para haver a troca das tripulações. A partir do bloco de um veículo são criadas as tarefas. Cada tarefa é um conjunto de viagens compreendidas entre duas *OT's*. Assim, durante sua realização não é possível que haja troca de tripulação.

A programação de uma tripulação é formada por um conjunto de tarefas, chamado de *jornada*. As jornadas são divididas em dois tipos: Pegada Simples ou Dupla Pegada. No primeiro tipo as tarefas são realizadas de uma única vez e os intervalos de tempo entre as tarefas são inferiores a duas horas. Caso ocorra um intervalo maior que duas horas, a jornada é classificada como de dupla pegada. Este intervalo não é contabilizado na remuneração da tripulação.

Ao se reunir as tarefas formando as jornadas, deve-se levar em conta inúmeras restrições operacionais e trabalhistas. As restrições consideradas neste trabalho referem-se àquelas praticadas por uma empresa do Sistema de Transporte Público do município de Belo Horizonte no ano 2002. Essas restrições podem ser classificadas em dois tipos: restrições essenciais (aquelas que obrigatoriamente têm que ser satisfeitas) e restrições não essenciais (aquelas cujo atendimento melhoram a qualidade das jornadas de trabalho mas que, se não satisfeitas, não geram jornadas inviáveis). As seguintes restrições são consideradas essenciais: (a) As trocas das tripulações só podem ocorrer nas *OT's*; (b) As trocas das tripulações só podem ocorrer entre grupos de linhas predeterminadas, ou seja, grupos de linhas com as mesmas características; (c) Uma tripulação que faz pegada simples tem direito a 30 minutos de descanso/alimentação durante sua jornada diária de trabalho, podendo este período ser fracionado em intervalos menores, desde que um deles seja maior ou igual a 15 minutos; (d) A jornada normal de trabalho diário é de 7:10 horas para as tripulações com pegada simples e

6:40 horas para aquelas com dupla pegada, acrescidas de até duas horas extras; (e) O tempo entre o final de uma jornada diária de trabalho e o seu início no dia seguinte deve ser de, no mínimo, 11 horas; (f) O número de jornadas do tipo dupla pegada deve estar limitado a um certo valor.

As restrições não essenciais consideradas foram as seguintes: (g) O tempo ocioso de uma tripulação deve ser o menor possível; (h) O número de horas extras deve ser minimizado; (i) O número de tripulações deve ser mínimo; (j) O número de vezes que uma tripulação troca de veículo deve ser reduzido; (k) O número de vezes que uma tripulação com dupla pegada finaliza a primeira parte da jornada em um ponto e inicia a segunda parte em um outro ponto deve ser reduzido.

3. MODELAGEM DO PROBLEMA

3.1. Representação

Uma solução do problema é representada por uma lista de jornadas diárias de trabalho, sendo cada jornada constituída por uma lista de tarefas.

3.2. Estruturas de Vizinhaça

Dada uma solução s , para atingir uma solução s' , onde s' é dito vizinho de s , são usados três tipos de movimentos: Realocação, Troca e *Link*, para definir, respectivamente, três estruturas diferentes de vizinhaça, a saber: $N^R(s)$, $N^T(s)$, $N^L(s)$, conforme ilustra a Figura a seguir.

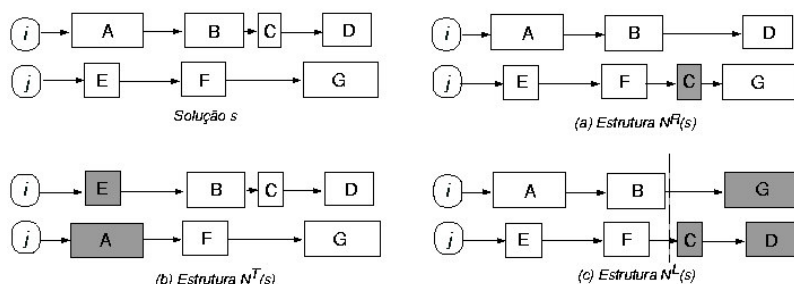


Figura 2: Estruturas de Vizinhaça

Nesta figura, A, ..., G representam tarefas a serem executadas nas jornadas i e j .

O movimento de realocação consiste em realocar uma tarefa de uma jornada para outra. Em (a), mostra-se que a tarefa C anteriormente pertencente à jornada i é realocada à jornada j . O conjunto de todos os vizinhos de s gerados através de movimentos de realocação define a estrutura de vizinhaça $N^R(s)$.

Já um movimento de troca consiste na permuta de tarefas entre duas jornadas distintas. Em (b) mostra-se que as tarefas A e E são trocadas entre as jornadas i e j . O conjunto de todos os vizinhos de uma escala s gerados a partir de movimentos de troca define a estrutura de vizinhaça $N^T(s)$.

Finalmente, o movimento *Link* consiste na troca de um conjunto de tarefas entre duas jornadas distintas i e j . Conforme ilustrado em (c), a partir de uma solução s define-se um ponto de corte em uma das jornadas, horário a partir do qual os blocos de tarefas serão trocados. O conjunto de todos os vizinhos de s gerados a partir de movimentos do tipo *Link* define a estrutura de vizinhança $N^L(s)$.

3.3. Função de Avaliação

Uma solução (ou escala) s é avaliada por uma função $f: s \in S \rightarrow \Re$, onde S é o conjunto de todas as possíveis soluções. A função de avaliação adotada baseia-se na penalização de cada um dos requisitos essenciais e não essenciais não atendidos. Desse modo, uma escala s é avaliada com base em duas componentes, uma de inviabilidade ($g(s)$), a qual mede o não atendimento aos requisitos essenciais, e outra de qualidade ($h(s)$), a qual mede o não atendimento aos requisitos considerados não essenciais. A soma dessas duas componentes constitui a função de avaliação de uma solução s , que deve ser minimizada, isto é:

$$f(s) = g(s) + h(s) \quad (1)$$

Deve ser observado que uma solução s é viável se, e somente se, $g(s) = 0$. Nas componentes da função f , os pesos atribuídos às diversas medidas refletem a importância relativa de cada uma delas e, sendo assim, deve-se tomar um valor bem mais elevado para os requisitos essenciais, de forma a privilegiar a eliminação das soluções inviáveis.

3.4. Geração de uma solução inicial

A solução inicial é gerada tomando-se como base a filosofia adotada pela empresa na construção manual de suas jornadas. Esta metodologia baseia-se na repartição dos blocos dos veículos. Cada bloco de um veículo contém um conjunto de tarefas a serem executadas pelo veículo no dia. Este bloco é dividido seguindo dois critérios: sua duração e o fato de o veículo retornar ou não à garagem mais de uma vez durante sua jornada diária. Caso o veículo não retorne à garagem mais de uma vez, a divisão do bloco é feita de acordo com a sua duração. Nesta situação, o bloco é dividido ao meio caso cada metade não ultrapasse 9:10 horas de trabalho, que é o tempo máximo de trabalho permitido por lei, incluindo as 2:00 horas extras. Caso contrário, o bloco é dividido em três partes, sendo as duas primeiras de no máximo 7:10 horas e a última com as tarefas remanescentes. Cada parte de um bloco é alocada a uma jornada diferente.

Caso o veículo retorne à garagem durante sua jornada, a repartição do bloco também se dá conforme sua duração, porém a divisão é feita no intervalo de tempo em que a diferença entre o final da última tarefa e o início da primeira tarefa, excluindo o tempo em que o veículo ficou na garagem, for inferior a 8:40 horas. O tempo em que o veículo permanece na garagem é excluído porque o tripulante que receber essa jornada fará dupla pegada e, portanto, não necessitará dos 30 minutos de intervalo para repouso e/ou alimentação.

Uma das vantagens de se gerar a solução inicial seguindo a filosofia da empresa é que esta não contém sobreposições de tarefas nem trocas de pontos ou linhas proibidas. Todavia, pode ocorrer excesso na duração das jornadas diárias de trabalho.

3.5. VNS aplicado ao PPT

O Método de Pesquisa em Vizinhos Variáveis, conhecido como VNS (*Variable Neighborhood Search*), proposto em Mladenovic e Hansen (1997), é um método de busca local que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. A ideia do método é a de explorar vizinhanças gradativamente mais “distantes” da solução corrente, focando a busca em torno de uma nova solução se e somente se um movimento de melhoria é realizado. O método VNS também inclui um procedimento de busca local a ser aplicado sobre o vizinho da solução corrente. O pseudocódigo desse algoritmo é apresentado pela Figura 2.

Procedimento VNS	
1	Seja s_0 uma solução inicial e r o número de estruturas de vizinhança;
2	$s \leftarrow s_0$; {Solução corrente}
3	<u>enquanto</u> (Critério de parada não satisfeito) <u>faça</u>
4	$k \leftarrow 1$; {Tipo de estrutura de vizinhança}
5	<u>enquanto</u> ($k \leq r$) <u>faça</u>
6	Gere um vizinho qualquer $s' \in N^{(k)}(s)$;
7	$s'' \leftarrow \text{BuscaLocal}(s')$;
8	<u>se</u> ($f(s'') < f(s)$)
9	<u>então</u> $s \leftarrow s''$; $k \leftarrow 1$;
10	<u>senão</u> $k \leftarrow k + 1$;
11	fim-se;
12	<u>fim-enquanto</u> ;
13	<u>fim-enquanto</u> ;
14	Retorne s ;
fim VNS;	

Figura 2: Método de Pesquisa em Vizinhos Variáveis

O procedimento de busca local implementado (linha 7 da Figura 2) foi o Método de Descida em Vizinhos Variáveis, conhecido como VND (*Variable Neighborhood Descent*), também proposto em Mladenovic e Hansen (1997). O método VND consiste nas linhas de 4 a 12 da Figura 2, excetuando-se a linha 7 e substituindo-se a linha 6 por “Encontre o melhor vizinho $s' \in N^{(k)}(s)$ ”. Desta forma, o método pára quando for atingida a última estrutura de vizinhança e nenhuma melhora na solução corrente for possível. Ao contrário do método VND, o método VNS tem condições de prosseguir a busca quando essa última situação ocorre, uma vez que retorna-se à primeira estrutura de vizinhança e seleciona-se um outro vizinho qualquer até que uma determinada condição de parada seja satisfeita.

Para resolver o PPT, foram desenvolvidos os métodos VNS-RTL, VNS-LRT, VNS-RLT, os quais utilizam as três estruturas de vizinhança, $N^R(s)$, $N^T(s)$ e $N^L(s)$, definidas na seção 3.2, em diferentes seqüências. Os três métodos utilizam o método VND e a mesma ordem de exploração de vizinhanças para fazer o refinamento das soluções. No método VNS-RTL, a seqüência de movimentos utilizados na exploração de uma solução s é Realocação, Troca e *Link*, nesta ordem. Já no método VNS-LRT, a seqüência de exploração é *Link*, Realocação e Troca. Finalmente, o método VNS-RLT considera a seqüência Realocação, *Link* e Troca.

Dada a elevada complexidade para avaliação de uma solução no movimento *Link*, a cada iteração do método VND ordenam-se, inicialmente, as jornadas de acordo com suas durações.

A seguir, é selecionado um conjunto das jornadas de maior duração (definido pelo parâmetro *MaiorJorn%* de todas as jornadas) e um conjunto das jornadas de menor duração (definido pelo parâmetro *MenorJorn%* de todas as jornadas), com $MenorJorn + MenorJorn \leq 100$. Então, a partir de um ponto de corte definido para cada uma das jornadas de maior duração, aplica-se o movimento *Link* considerando-se que um bloco de tarefas de uma dessas jornadas seja permutado com um bloco de tarefas iniciado a partir do mesmo ponto de corte em uma das jornadas de menor duração.

Para minimizar o esforço computacional na exploração de soluções nas três estruturas de vizinhança, evita-se a análise completa de uma dada vizinhança em cada iteração do método VND, interrompendo-se a busca na vizinhança corrente sempre que ocorrer uma solução de melhora.

4. RESULTADOS

Os algoritmos foram desenvolvidos na linguagem C++ usando o compilador Borland C++ Builder 6.0 e testados em um microcomputador com processador Pentium IV, 1.8 MHz, com 256 MB de memória RAM, sob sistema operacional Windows XP. O problema considerado para teste foi relativo a um dia útil de uma empresa de transporte público, situada em Belo Horizonte, responsável por 11 linhas de ônibus, com frota empenhada de 111 veículos.

Os pesos utilizados para penalizar os diversos requisitos da função de avaliação de uma solução foram os mesmos utilizados em Silva et al. (2002). O critério de parada adotado nos testes foi o tempo total de execução, restrito a 60 minutos.

O ponto de corte considerado para o movimento *Link* foi seis horas do início de uma jornada. Já os parâmetros *MaiorJorn* e *MenorJorn*, que definem o percentual de jornadas de maior duração e menor duração, respectivamente, foram ambos tomados como 10%.

Cada variante do método VNS foi executada 20 vezes, cada qual partindo de uma semente diferente de números aleatórios. A Tabela 1 apresenta, para cada algoritmo testado, o melhor valor obtido e o desvio do valor médio em relação ao melhor valor encontrado na instância considerada, isto é:

$$\text{desvio} = ((\text{Valor Médio} - \text{Melhor Valor}) / (\text{Melhor Valor})).$$

Tabela 1: Desempenho dos algoritmos

Método	VNS-LRT	VNS-RLT	VNS-RTL
Melhor valor	1.188.980	1.213.970	1.219.420
Desvio	2,2%	3,9%	4,9%

A Tabela 1 mostra que o método VNS-LRT obteve o menor valor para a função de avaliação. Além disso, esse método se mostrou mais robusto que os demais, apresentando pouca variabilidade em relação à solução final obtida.

A Tabela 2 apresenta algumas das características da solução em uso na empresa estudada, bem como as características das melhores soluções produzidas pelas três variantes do método VNS.

Tabela 2: Características das soluções geradas pelos algoritmos

	Empresa	VNS-LRT	VNS-RLT	VNS-RTL
#Jornadas	219	219	219	218
Hora-extra (hh:mm)	116:00	87:13	87:30	92:23
Trocas de veículos	0	28	32	31
Trocas de linhas	0	13	15	20

Pode-se observar que as soluções obtidas com todas as variantes do método VNS reduziram o total de horas extras diárias pagas pela empresa. Com o método VNS-RTL teve-se, adicionalmente, redução no número de jornadas e, portanto, de tripulações necessárias para a operação da frota. Tais reduções foram possíveis devido à flexibilização da operação, com a realização de trocas de veículos e de linhas.

A Figura 3 ilustra a evolução típica do valor da melhor solução nas três variantes do método VNS nos instantes iniciais do processo de busca. Pode-se observar que a variante VNS-LRT alcança soluções de boa qualidade mais rapidamente que as outras duas variantes. Esse comportamento era de se esperar, uma vez que ao se iniciar com o movimento *Link*, é possível retirar as inviabilidades da solução inicial mais rapidamente, pois o mesmo trabalha com blocos de tarefas ao invés de uma única tarefa ou de um par de tarefas, como nos movimentos Realocação e Troca.

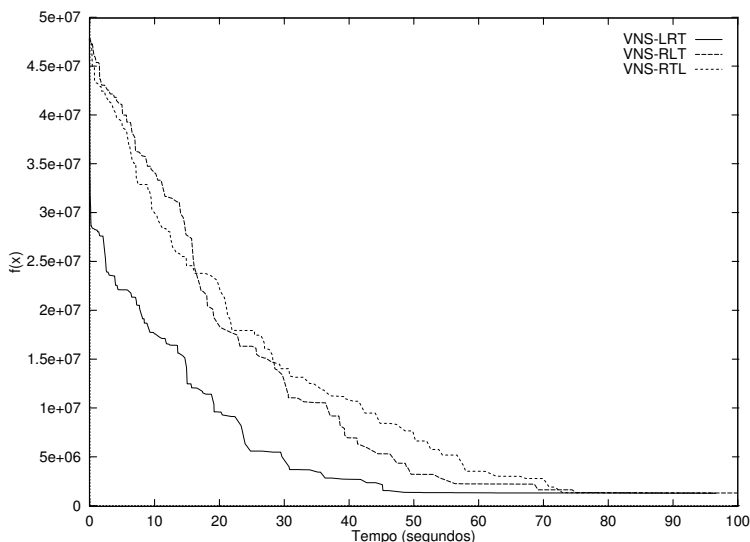


Figura 3: Evolução da melhor solução nos métodos VNS

5. CONCLUSÕES

Este trabalho apresenta três variantes do Método de Pesquisa em Vizinhança Variável aplicadas à resolução do problema de programação de tripulações de ônibus urbano considerando a realidade brasileira. A variante VNS-LRT, partindo de uma solução inicial seguindo a filosofia da empresa, mostrou ser a mais eficiente na abordagem do problema. Além de gerar soluções de boa qualidade mais rapidamente, mostrou ser um método robusto e o que produziu a solução final de melhor qualidade, com redução no número de horas-extras, um dos principais itens na planilha de custos com a mão-de-obra operacional.

Agradecimentos

Os autores agradecem ao CNPq e à Universidade Federal de Ouro Preto pelo apoio dado ao desenvolvimento deste trabalho, bem como à Borland Latin America pela cessão de uma licença de uso do software C++ Builder 6.0.

REFERÊNCIAS BIBLIOGRÁFICAS

- Barnhart, C.; E. L. Johnson; G. L. Nemhauser; M. P. Savelsbergh. e P. H. Vance (1998) Branch-and-Price: Column Generation for Solving Huge Integer Programs, *Operations Research*, v. 46, p. 316-329.
- Bouzada, C.F. (2002) Análise das Despesas Administrativas no Custo do Transporte Coletivo por Ônibus no Município de Belo Horizonte, *Dissertação de mestrado*, Fundação João Pinheiro, MG.
- Clement, R. e A. Wren (1995) Greedy Genetic Algorithms, Optimizing Mutantis and Bus Driver Scheduling. In: J.R. Daduna, I. Branco e J.M.P. Paixão (eds.) *Computer-Aided Transit Scheduling*, p. 213-235, Springer-Verlag.
- Daduna, J. R.; I. Branco e J. M. P. Paixão (1995) (eds), *Computer-Aided Transit Scheduling*, Proceedings of the 5th International Workshop on Computer-Aided Scheduling of Public Transport, Springer-Verlag.
- Desrochers, M. e F. Soumis (1989) A Column Generation Approach to the Urban Transit Crew Scheduling Problem, *Transportation Science*, v. 23, p. 1-13.
- Elias, S. E. G. (1964) The Use of Digital Computers in the Economic Scheduling for Both Man and Machine in Public Transport. *Technical Report 49*, Kansas State University Bulletin, Kansas, EUA.
- Feo, T.A. e M. G. C. Resende (1995) Greedy Randomized Adaptive Search Procedures, *Journal of Global Optimization*, v. 6, p. 109-133.
- Fores, S.; L. Proll e A. Wren (1999) An Improved ILP System for Driver Scheduling. In: Wilson, N. H. M. (ed.) *Computer-Aided Transit Scheduling*, p. 43-61, Springer-Verlag.
- Friberg, C. e K. Haase (1999) An Exact Branch and Cut Algorithm for the Vehicle and Crew Scheduling Problem. In: Wilson, N. H. M. (ed.) *Computer-Aided Transit Scheduling*, p. 63-80, Springer-Verlag.
- Glover, F. e M. Laguna (1997) *Tabu Search*, Kluwer Academic Publishers, Boston.
- Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Berkeley.
- Kirkpatrick, S.; D. C. Gellat e M. P. Vecchi (1983) Optimization by Simulated Annealing, *Science*, v. 220, p. 671-680.
- Kwan, A. S. K; R. K. Kwan e A. Wren (1999) Driver Scheduling Using Genetic Algorithms with Embedded Combinatorial Traits. In: Wilson, N. H. M. (ed.) *Computer-Aided Transit Scheduling*, p. 81-102, Springer-Verlag.
- Lourengo, H. R.; J. P. Paixão e R. Portugal (2001) Multiobjective Metaheuristics for the Bus-Driver Scheduling Problem, *Transportations Science*, v. 35, p. 331-343.
- Manington, B. e A. Wren (1975) A General Computer Method for Bus Crew Scheduling, *Workshop on Automated Techniques for Scheduling of Vehicle operators for Urban Public Transportation Services*, Chicago.
- Mladenovic, N. e P. Hansen (1997) Variable Neighborhood Search, *Computers and Operations Research*, v. 24, p. 1097-1100.
- Shen, Y. e R. S. K. Kwan (2001) Tabu Search for Driver Scheduling. In: S. Voß e J. Daduna (eds.) *Computer-Aided Scheduling of Public Transport*, p. 121-135, Springer-Verlag.
- Siqueira, P. H. (1999) Aplicação do Algoritmo do Matching no Problema da Construção de Escalas de Motoristas e Cobradores de Ônibus, *Dissertação de mestrado*, Setor de Tecnologia e de Ciências Exatas, UFFR, PR.
- Silva, G. P.; M. J. F. Souza e J. M. C. B. Alves (2002) Resolução do Problema de Programação Diária da

- Tripulação de Ônibus Urbano via *Simulated Annealing*, XVI Congresso de Pesquisa e Ensino em Transportes, Panorama Nacional de Pesquisa em Transportes, v. 2, p. 95-104, ANPET.
- Souza, M. J. F.; L. X. T. Cardoso; G. P. Silva; M. M. S. Rodrigues e S. M. S. Mapa (2004) Metaheurísticas aplicadas ao Problema de Programação de Tripulações no Sistema de Transporte Público, *Tendências em Matemática Aplicada e Computacional*, v. 5, 12 p., to appear.
- Smith, B. M. e A. Wren (1988) A Bus Crew Scheduling System Using a Set Covering Formulation, *Transportation Research*, v. 22a, p. 97-108.
- Voß, S. e J. R. Daduna (2001) (eds), "Computer-Aided Transit Scheduling", Proceedings of the 9th International Workshop on Computer-Aided Scheduling of Public Transport, Springer-Verlag.
- Wilson, N. H. M. (1999) (ed) "Computer-Aided Transit Scheduling", Proceedings of the 7th International Workshop on Computer-Aided Scheduling of Public Transport, Springer-Verlag.
- Wren, A. e J. M. Rousseau (1995) Bus Driver Scheduling - An Overview. In: J. R. Daduna, I. Branco e J. M. P. Paixão (eds.) *Computer-Aided Transit Scheduling*, p. 173-187, Springer-Verlag.
- Wren, A. e D. O. Wren (1995) A Genetic Algorithm for Public Transport Driver Scheduling, *Computer and Operations Research*, v. 22, p. 101-110.

Marcone Jamilson Freitas Souza (marcone@iceb.ufop.br)
Gustavo Peixoto Silva (gustavo@depro.em.ufop.br)
Sílvia Maria Santana Mapa (silvinhamapa@yahoo.com.br)
Departamento de Computação, Universidade Federal de Ouro Preto
Campus Universitário
35.400-000 Ouro Preto, MG, Brasil