

Otimização da escala mensal de motoristas de ônibus urbano utilizando a heurística *Variable Neighborhood Search*

Gustavo Peixoto Silva¹, Raphael Felipe de Carvalho Prates¹

Abstract: One of the last stage from public transportation planning concerns to defining the scale of urban bus drivers for short term period, called Crew Rostering Problem (CRT). This problem aims to generate sequences of daily shifts, including weekdays, Saturdays and Sundays, respecting labor laws and operational constraints. Moreover, a good crew roster should provide a better division of workload among the crews and still to reduce the overtime costs paid by the company. The model proposed in this paper is able to generate solutions satisfying a fixed scheme called 5/1 of day-off, beyond the labor laws and operational constraints imposed by the company. The Variable Neighborhood Search metaheuristic (VNS) was implemented using different neighborhood structures, varying the number of modifications performed on the current solution. The implementation was tested with data from a midsize company and the results show significant improvements compared to the solution adopted by the company.

Keywords: Crew rostering problem. Mass transit. Metaheuristic.

Resumo: Uma das últimas etapas no planejamento do transporte público consiste em definir a escala dos motoristas dos ônibus urbanos para um determinado período, denominada *Problema de Rodízio de Tripulações*. Esta etapa tem como objetivo gerar sequências de jornadas diárias, compreendendo os dias úteis, sábados e domingos, que respeitem as restrições legais e operacionais. Além disso, uma boa escala deve proporcionar uma melhor divisão da carga de trabalho entre as tripulações e ainda reduzir os custos com as horas extras pagas pela empresa. O modelo proposto neste trabalho gera soluções que respeitam o padrão de folga fixa do tipo 5/1 além das restrições legais e operacionais impostas pela empresa. A metaheurística *Variable Neighborhood Search* foi implementada utilizando diferentes estruturas de vizinhança e variando o número de modificações na solução. A implementação foi testada com dados de uma empresa de médio porte e os resultados mostraram melhorias significativas em relação à solução adotada pela empresa.

Palavras-chave: Rodízio das tripulações. Transporte público. Metaheurística.

1 INTRODUÇÃO

O problema de rodízio de tripulações (PRT) consiste em definir uma escala mensal de trabalho constituída pela sequência de jornadas diárias, de forma que se consiga balancear as horas extras e ociosas dentro de um dado horizonte de planejamento, assim como minimizar o total de tripulações empregadas (motoristas e cobradores). As restrições legais e operacionais devem ser respeitadas, para que o modelo construa soluções viáveis e permita a sua aplicação no cotidiano das empresas que atuam no sistema de transporte coletivo.

Um modelo normalmente empregado pelas empresas, para definir a escala mensal das tripulações, é o da escala fixa. Este é um modelo ineficiente uma vez que as tripulações repetem as mesmas jornadas ao longo do horizonte de plane-

jamento acumulando as horas extras ou ociosas que ocorrem diariamente. Além dos custos, surgem ainda alguns rodízios preferenciais, tais como os matutinos e aqueles com acúmulo de horas extras, o que pode gerar um ambiente desigual de trabalho.

Devido ao sistema de banco de horas de trabalho, previsto em lei e adotado por algumas empresas, torna possível compensar, ao longo do horizonte de planejamento, jornadas com horas extras com jornadas que apresentam horas ociosas. Este procedimento justifica o estudo de métodos de alocação das jornadas que buscam reduzir os custos e criar escalas de trabalho mais equilibradas e justas para as tripulações (Toffolo *et al.*, 2005). Dessa forma, o PRT pode ser visto como uma flexibilização do modelo de escala fixa, permitindo o rodízio das jornadas diárias e a utilização do sistema de banco de horas.

Neste estudo é proposto um modelo heurístico para resolver o PRT considerando o padrão de folga fixa do tipo 5/1 no qual as tripulações trabalham cinco dias consecutivos e folgam no

¹ Universidade Federal de Ouro Preto, Instituto de Ciências Exatas e Biológicas, Ouro Preto, MG, Brasil.

Manuscrito recebido em 20/05/2013 e aprovado para publicação em 10/03/2014. Este artigo é parte de TRANSPORTES v. 22, n. 1, 2014. ISSN: 2237-1346 (online).

DOI: <http://dx.doi.org/10.14295/transportes.v22i1.698>

sexto dia. Embora esta estratégia possa não ser tão econômica quanto aquelas que não apresentam um padrão fixo para as folgas (Mayrink e Silva, 2013), ele é largamente adotado pelas empresas devido à sua simplicidade de implementação. O que se espera com a realização de novos estudos sobre o rodízio de tripulações é minimizar os custos totais e ainda verificar a consequência de pequenas variações nas regras de alocação das jornadas em relação à escala das tripulações.

Este trabalho está dividido da seguinte forma: na Seção 2 é feita uma revisão das principais publicações que abordam o problema e na Seção 3 são apresentadas as características do problema abordado. A Seção 4 apresenta a metodologia adotada para a resolução do problema e na Seção 5 são mostrados os resultados obtidos ao resolver um caso real. Na Seção 6 são feitas as considerações finais do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Um dos trabalhos pioneiros na abordagem do problema de rodízio de tripulações do sistema de transporte público é o de Carraresi e Gallo (1984), no qual o problema foi formulado utilizando um modelo de designação multi-nível com gargalo. Para encontrar m sequências de jornadas tal que a carga máxima de trabalho seja minimizada, é definida uma rede n -níveis $G = (N, A)$, onde n é o número de dias no período e m é o número de jornadas por dia. Os autores assumem que o mesmo conjunto de jornadas deve se repetir a cada dia. Um nó na rede, denotado por $\langle i, k \rangle$ representa a jornada i executada no dia (nível) k . E um arco do nó $\langle i, k \rangle$ para o nó $\langle j, k+1 \rangle$ é incluído na rede se as jornadas i e j puderem ser executadas pela mesma tripulação, em dias consecutivos. Denotam-se tais arcos por (i, j, k) com fluxo x_{ij}^k assumindo valores zero ou um. Um peso w_i é associado a cada jornada i , representando o custo para a tripulação que a executar. Assim, o problema se resume em encontrar, na rede G , m caminhos disjuntos com origem no primeiro conjunto de nós $\{\langle 1, 1 \rangle, \langle 2, 1 \rangle, \dots, \langle m, 1 \rangle\}$ e destino no último conjunto de nós $\{\langle 1, n \rangle, \langle 2, n \rangle, \dots, \langle m, n \rangle\}$, tal que o caminho mais longo tenha um comprimento mínimo. Para $m = 2$ tem-se o problema padrão de designação com gargalo. Para $m > 2$, ele é denominado problema de designação multi-nível com gargalo. Os autores mostraram ainda

que o PRT é um problema NP-completo e apresentaram uma heurística que resolve iterativamente vários problemas de assinalamento com gargalo (*bottleneck assignment problem*) sobre os nós nos níveis k e $k+1$.

Bianco *et al.* (1992) formularam o PRT por meio de um modelo de programação linear inteira e descreveram uma heurística que utiliza o limitante inferior, proveniente da relaxação da integralidade do modelo, para reduzir as dimensões do problema. A cada iteração o algoritmo resolve um problema de designação multi-nível com gargalo, para o qual é proposto um novo procedimento que produz soluções que convergem assintoticamente para a solução ótima.

Yunes *et al.* (2005) abordou o PRT utilizando técnicas de programação matemática e de programação por restrições. Para o PRT é apresentado um algoritmo híbrido de geração de colunas combinando às duas técnicas mencionadas. Foram utilizadas duas metodologias para resolver esse problema. Na primeira abordagem, o problema é modelado como um problema de programação linear inteira, e resolvido com a técnica de *branch and bound*. A segunda abordagem se baseia em programação por restrições. A primeira abordagem só é aplicável a problemas de pequeno porte, e ainda se mostra menos eficaz do que a técnica programação por restrições.

No trabalho de Caprara *et al.* (2003) foram apresentados modelos matemáticos e algoritmos para resolver uma família de problemas de programação de pessoal (*staff scheduling*) que surgem em casos reais. Nestes problemas são fornecidas as jornadas diárias a serem executadas, assim como a duração de cada jornada e as necessidades de ocorrência dos períodos de folga para cada funcionário, durante o período de planejamento. O objetivo dos modelos consistiu em minimizar o número de funcionários necessários para executar todas as tarefas. Os autores decompõem o problema em duas etapas. Na primeira etapa é determinado o número mínimo de funcionários e seus respectivos padrões de folgas, ou seja, os dias em que cada funcionário deve trabalhar e os dias em que eles devem folgar. Nesta etapa ainda não fica definida a jornada que cada funcionário deverá realizar. O problema é formulado como um Problema de Programação Inteira e resolvido com a técnica de *branch-and-bound*, que pode ser combinada com a técnica de geração de colunas. Na segunda etapa são definidas as jorna-

das de trabalho de cada funcionário, garantindo a viabilidade da solução enquanto faz o balanceamento do tempo total de trabalho realizado pelos funcionários durante o período. Esta etapa é resolvida por meio de uma sequência de Problemas de Transporte, um para cada dia do horizonte de planejamento. Os experimentos computacionais tratam da alocação de atendentes de uma central de chamadas de emergência (*emergency call center*) e mostram que a melhor abordagem se baseia no modelo cujas variáveis estão associadas com padrões viáveis e são gerados por meio de programação dinâmica ou resolvendo um outro problema de programação inteira.

O trabalho de Ernst *et al.* (2004a) apresenta uma revisão bibliográfica com quase 200 referências sobre problemas de programação e rodízio de funcionários (*staff scheduling and rostering*), assim como diferentes métodos de resolução dos mesmos. A revisão aborda uma variedade de áreas de aplicação e técnicas de resolução comuns aos problemas. O trabalho também traz um esquema de classificação para descrever problemas de rodízio de funcionários. Uma revisão mais completa ainda pode ser encontrada em Ernst *et al.* (2004b), pois trata-se de uma revisão bibliográfica comentada que agrupa, de forma compreensiva, um conjunto de cerca de 700 referências da área. A revisão está focada principalmente nos algoritmos apresentados na literatura para gerar rodízios e programações de pessoal. Mas também cobre áreas correlatas como planejamento da força de trabalho e estimativa do número de funcionários necessários para executar um determinado conjunto de tarefas. As referências são classificadas de acordo com o tipo de problema abordado, a área de aplicação e o método de resolução empregado. Ao final, um breve resumo é apresentado para cada artigo citado.

Toffolo *et al.* (2005) formularam o problema de rodízio de tripulações empregando duas metaheurísticas: *Simulated Annealing* e *Iterated Local Search (ILS)*, sendo adotado como procedimento de busca local o método randômico de descida. Uma estrutura de vizinhança é definida pela troca das jornadas entre duas tripulações. Desta forma, para um referido dia k tenta-se trocar as jornadas entre duas tripulações i e j . Uma função de avaliação é construída levando em consideração restrições essenciais, tais como obrigações legais referentes às folgas das tripulações, e restrições não essenciais, situações que apesar de

indesejáveis podem ocorrer em determinadas circunstâncias. As restrições não essenciais são introduzidas para evitar que determinadas tripulações façam muitas horas extras no período, em detrimento de outras. Esta é uma situação indesejável, pois normalmente os tripulantes preferem fazer horas extras para terem suas remunerações aumentadas. Os autores mostram que para a resolução do PRT o método heurístico *ILS* foi mais eficiente do que o *Simulated Annealing*, encontrando soluções com menor custo em um menor tempo de processamento. Ambos os métodos são capazes de encontrar soluções viáveis com tempo de processamento muito reduzido.

Em Moz *et al.* (2009) é apresentado um modelo bi-objetivo para tratar o PRT de motoristas de ônibus urbano, assumindo um contexto de rodízio não cíclico e apropriado para lidar com as diversas necessidades inerentes aos motoristas, como por exemplo, o absentéismo. Assim, duas heurísticas evolucionárias foram descritas: uma que segue uma estratégia utópica (Pato e Moz, 2008), e a outra que é uma versão adaptada do *SPEA2 (Strength Pareto Evolutionary Algorithm)*. Elas se diferem em relação à estratégia utilizada para se aproximar da fronteira de Pareto. O horizonte de planejamento considerado neste estudo é de vinte e oito dias, ou seja, quatro semanas exatas, começando cada lista de tarefas numa segunda-feira. Uma sequência de tarefas pode ser avaliada de acordo com a equidade na distribuição de domingos/fins de semana para os motoristas, equidade na distribuição de tarefas em atraso, entre outras. Como objetivo do problema, os autores consideram a minimização do total de horas extras realizadas pelas tripulações (objetivo 1), e também a minimização do número de tripulações contratadas (objetivo 2). O trabalho mostra que ambas as metodologias são adequadas para resolver o problema. Entretanto, a segunda se mostra mais eficiente. Em um tempo de processamento razoável, elas foram capazes de gerar uma série de soluções que satisfazem todas as restrições do problema. Adicionalmente, entre as soluções foram identificadas aquelas potencialmente eficientes com respeito aos dois objetivos, sendo um de interesse da empresa e o outro de interesse dos motoristas. Segundo os autores, ambas as heurísticas apresentam vantagens e desvantagens. Desta forma, eles sugerem que as heurísticas devam ser utilizadas de forma complementar. Por outro lado, com pouco esforço, as heurísticas

podem ser adaptadas a uma grande variedade de problemas de rodízio.

Recentemente, Mayrink e Silva (2013) abordaram o problema de rodízio de tripulações em duas etapas, utilizando dois modelos distintos de otimização. Na primeira etapa é desenvolvido um modelo de designação baseado no trabalho de Carraresi e Gallo (1984). Assim, cada nó representa uma jornada diária, e os arcos unem as jornadas de um dia para o dia seguinte que podem ser executadas pela mesma tripulação. A cada arco é atribuído um custo em função do acúmulo ou compensação de horas extras com horas ociosas. O modelo de designação é resolvido de forma iterativa empregando o algoritmo *Out-of-kilter* (Ahuja *et al.*, 1993). O modelo tem como objetivo minimizar os custos atribuídos às sequências de jornadas construídas iterativamente. A cada iteração, as jornadas do próximo dia são acrescentadas ao rodízio até formarem o rodízio semanal. Posteriormente, utilizando o mesmo método, o rodízio do período é construído utilizando os rodízios semanais como dados de entrada.

Ao final da primeira etapa, é gerado um rodízio que pode conter soluções inviáveis no período, uma vez que as restrições referentes às folgas, tais como a exigência de uma folga a cada seis dias de trabalho, normalmente não são contempladas para todas as tripulações. Logo, é necessária a adição de tripulações do tipo *folguistas*, adequando o modelo às restrições das folgas trabalhistas. A formulação deste problema se dá por meio de um modelo de programação linear inteira. O objetivo deste modelo é minimizar a quantidade de tripulações do tipo *folguistas*, sujeito às restrições de folga das tripulações. Neste caso as folgas não têm um padrão fixo, mas simplesmente garantem que as tripulações não trabalhem mais do que seis dias consecutivos e que tem pelo menos uma folga no domingo por mês. A combinação dos dois modelos foi testada com dados reais de uma empresa de transporte coletivo da região metropolitana de Belo Horizonte e os resultados obtidos se mostraram muito superiores à solução empregada pela empresa.

Mesquita *et al.* (2011) apresentaram um método que obtém soluções viáveis para um modelo de programação multiobjetivo binária do problema de programação integrada entre veículos, tripulações e o rodízio das tripulações. Este método é um algoritmo sequencial embutido em um ambiente de programação por metas. Inicial-

mente é resolvido o problema de programação dos veículos para todos os dias no horizonte de planejamento. Posteriormente, usando essas soluções como ponto de partida e cobrindo o horizonte de planejamento, o PRT é abordado em um contexto de programação não cíclica, por um modelo binário multiobjetivo que considera tanto os interesses da empresa quanto os dos motoristas. Este problema também é resolvido por uma aproximação de programação por metas que levam a soluções mensais quase ótimas. A fim de avaliar a qualidade das diferentes soluções para o rodízio das tripulações, foram realizadas medidas de perspectivas diferentes do custo.

Em Nurmi *et al.* (2012) é apresentada uma forma de planejar os dias de folga das tripulações em uma base anual e as jornadas em uma base mensal para uma empresa finlandesa de transporte urbano. Os dias de folga e as jornadas de trabalho são planejadas e alocadas usando um algoritmo que inclui características de heurísticas populacionais, assim como as heurísticas não populacionais *Simulated Annealing*, *Busca Tabu* e *Cadeias de Ejeção*. As restrições do problema são classificadas nos seguintes tipos: cobertura, regulamentação, operacionais, preferências operacionais e pessoais. O algoritmo para o método de solução do PRT é baseado em um método de busca local em população, na técnica de cadeias de ejeção e procedimentos do tipo *Lin-Kernighan* (Lin e Kernighan, 1973). A principal heurística é a busca gulosa de subida da encosta (*hill-climbing*) e os autores citam também que a solução funciona melhor com dados de entrada randômicos.

3 PROBLEMA DE RODÍZIO DE TRIPULAÇÕES

As jornadas diárias, a serem executadas por cada tripulação, são constituídas por uma sequência de viagens que serão realizadas por uma mesma tripulação ao longo do dia. As jornadas podem ser agrupadas em jornadas dos dias úteis, jornadas dos sábados e jornadas dos domingos/feriados. Esta classificação se deve ao fato dos quadros de horários serem iguais nos dias úteis e variarem nos sábados e domingos/feriados de acordo com a demanda.

As jornadas são divididas ainda em relação ao seu horário de início. Normalmente o dia é dividido em quatro *turnos*, cada um com duração de seis horas. O primeiro turno inicia às 04:00 e ter-

mina às 09:59. De maneira análoga são definidos os demais turnos. Outra característica de interesse diz respeito à existência de um intervalo de interrupção ao longo da jornada diária. As jornadas do tipo *dupla pegada* apresentam uma interrupção de pelo menos duas horas e estão relacionadas aos veículos que operam apenas nos horários do dia com maiores demandas. Caso contrário, a jornada é denominada como *pegada simples*. Neste trabalho, é considerada a carga diária de trabalho de seis horas e quarenta minutos. Jornadas com duração superior apresentam horas extras e com duração inferior horas ociosas ou ociosidade. Certas tripulações, denominadas *folguistas*, são alocadas à medida que surge a necessidade de se cobrir as folgas das demais tripulações. Estas são tripulações iguais às demais, exceto pelo fato de não pertencerem a um determinado turno.

Diversas restrições operacionais e legais devem ser consideradas no PRT, sendo estas denominadas restrições do problema. Estas restrições podem variar de região para região em função dos acordos coletivos firmados entre os sindicatos patronais e dos empregados do setor. Assim, as restrições respeitadas neste trabalho são:

- i) o horizonte de planejamento sempre inicia em uma segunda-feira;
- ii) o horizonte de planejamento é constituído de 7 semanas;
- iii) o tempo de descanso mínimo entre uma jornada e a próxima é de onze horas;
- iv) nos dias úteis as tripulações devem executar jornadas pertencentes ao mesmo turno;
- v) as tripulações fazem somente *pegada simples*, *dupla pegada* ou jornadas do tipo *no-turno* ao longo da semana;
- vi) as tripulações que fizerem jornadas do tipo *dupla pegada* devem folgar aos domingos;
- vii) nenhuma tripulação pode trabalhar mais de seis dias consecutivos sem folga;
- viii) todas as tripulações têm direito a uma folga no domingo a cada 7 semanas; e
- ix) as folgas são consideradas fixas ao longo do horizonte de planejamento, sendo utilizado o padrão 5/1, ou seja, trabalham 5 dias consecutivos e folgam 1 dia.

Para a elaboração do rodízio de tripulações foi utilizado como dados de entrada um conjunto de jornadas diárias fornecidas por uma empresa do transporte coletivo urbano de Belo Horizonte-MG. Os dados de entrada correspondem a um horizonte de planejamento constituído de 7 semanas. As jornadas a serem executadas nos dias úteis se repetem de segunda a sexta. As jornadas

dos sábados e domingos são diferentes devido à variação na demanda de passageiros, surgindo folgas naturais nestes dias. Estas folgas não são, no entanto, suficientes para atender à legislação, sendo necessária a alocação de tripulações *folguistas* para suprir as demais folgas.

4 MÉTODO DE RESOLUÇÃO DO PROBLEMA

Neste trabalho o PRT foi resolvido empregando a heurística *Variable Neighborhood Search* (VNS) (Mladenovic e Hansen, 1997) clássica tendo como método de busca local o *Variable Neighborhood Descent* (Mladenovic e Hansen, 1997). Diferentes estruturas de vizinhança foram implementadas para determinar aquela que produz a melhor solução para o problema. Para construir a solução inicial do VNS, foi considerado que as tripulações já se encontravam em operação. Portanto, no início do período, cada tripulação já conta com um número de dias trabalhados no período anterior. Como a legislação não permite que os funcionários trabalhem mais do que seis dias consecutivos, para cada tripulação, foi gerado um número aleatório entre zero e seis que corresponde ao número de dias trabalhados consecutivamente no mês anterior. Desta forma, aquele tripulante que trabalhou, por exemplo, quatro dias consecutivos no período anterior (número gerado aleatoriamente) deve trabalhar no primeiro dia e folgar no segundo dia do período em questão, cumprindo assim o padrão adotado que é do tipo 5/1.

Na heurística utilizada neste trabalho, são exploradas diferentes estruturas de vizinhança, sendo necessário definir, para cada tipo de problema, as diversas estruturas a serem utilizadas. Assim, foram implementadas e testadas três diferentes estruturas de vizinhança, variando também o número de movimentos permitidos. Dessa forma, foi possível chegar aos parâmetros que melhor exploraram o espaço de soluções para o PRT.

4.1 Metaheurística *Variable Neighborhood Search* - VNS

A ideia do VNS é simples e consiste em realizar mudanças sistemáticas de estruturas de vizinhança dentro de um processo de exploração do espaço de soluções. À medida que as estruturas de vizinhança são incrementadas, o caminho de busca é modificado, explorando soluções mais

distantes da solução corrente. A busca em torno de uma nova solução se dá quando um movimento de melhora é realizado. Quando isso ocorre, o processo de busca retorna à primeira vizinhança. Na sua versão original, o VNS faz o uso do método VND para realizar o processo de busca local.

A implementação do VNS parte da solução inicial s_0 que é adotada como a solução corrente s e a cada iteração uma nova solução s' é criada através de mudanças aleatórias na solução corrente dentro de uma estrutura de vizinhança $N^{(k)}(s)$. Estas mudanças dependem do valor do parâmetro k , que determina qual das vizinhanças será empregada. No problema abordado neste trabalho, a variável k determina a quantidade de jornadas trocadas dentro de um mesmo dia ou em dias diferentes dependendo da estrutura de vizinhança

considerada. Assim, começando em $N^{(l)}(s)$, troca-se aleatoriamente uma jornada de duas tripulações em um determinado dia, gerando a nova solução s' . A partir de s' aplica-se um procedimento de busca local encontrando uma solução s'' . Se s'' for melhor do que a solução corrente, é feita a atualização, retornando à primeira estrutura de vizinhança $N^{(l)}(s)$. Caso contrário, o procedimento se repete para a estrutura de vizinhança $N^{(k+1)}(s)$. O valor de k é incrementado até o número máximo de estruturas de vizinhança utilizadas. Após atingir seu valor máximo, ele retorna para a vizinhança $N^{(l)}(s)$ e o processo continua até que a condição de parada seja alcançada. Nesta implementação foi utilizado o tempo de processamento como condição de parada. O Algoritmo 1 sintetiza a metaheurística VNS.

Procedimento VNS(solução s)

Início:

1. Seja um conjunto de $kmax$ estruturas de vizinhanças;
2. Seja $f(s)$ a função de avaliação de uma solução s dada pela expressão (1);
3. Encontre uma solução inicial s e determine uma condição de parada;
4. Enquanto a condição de parada não for satisfeita faça:
5. $k \leftarrow 1$;
6. Enquanto $k \leq kmax$ faça:
7. Gere um vizinho qualquer s' de s utilizando a k -ésima vizinhança;
8. Encontre o melhor vizinho s'' de s' aplicando o procedimento VND(s',k);
9. Se $f(s'')$ for melhor do que $f(s)$ // $f(.)$ dada pela expressão (1)
10. Então $s \leftarrow s''$ e $k \leftarrow 1$;
11. Senão
12. $k \leftarrow k + 1$;
13. Fim_enquanto;
14. Fim_enquanto;
15. Retorna a solução s ;

Fim

Algoritmo 1 - Pseudo-código da Metaheurística VNS

4.2 Variable Neighborhood Descent – VND

O VND é um método de refinamento que se baseia em mudanças sistemáticas de estruturas de vizinhança. Este método tem como dados de entrada uma solução inicial, uma função de avaliação e um índice para as estruturas de vizinhança que são utilizadas pelo método de busca do melhor vizinho. Caso a solução resultante da busca melhore a função de avaliação com relação à so-

lução corrente, ela é adotada como solução corrente e o procedimento retorna à primeira estrutura de vizinhança $N^{(l)}(s)$. Caso contrário, o processo se repete para a próxima estrutura de vizinhança $N^{(k+1)}(s)$. O método continua até que todas as estruturas de vizinhança tenham sido exploradas, retornando a melhor solução encontrada, conforme apresentado no Algoritmo 2.

Procedimento VND(solução s , vizinhança k)**Início**

1. Seja k uma estrutura de vizinhança dada;
2. $r \leftarrow 1$;
3. Enquanto $r \leq k$ faça:
 4. Encontre o melhor vizinho s' de s na vizinhança r ;
 5. Se $f(s')$ for melhor do que $f(s)$
 6. Então $s \leftarrow s'$ e $k \leftarrow 1$;
 7. Senão
 8. $r \leftarrow r + 1$;
9. Fim_enquanto;
10. Retorna a solução s ;

Fim**Algoritmo 2** - Pseudo-código do Procedimento VND**4.3 Função de avaliação**

O custo de uma solução para o PRT pode ser calculado em função do total de horas extras e ociosas acumuladas no período. Ao computar o “saldo” de uma dada tripulação no período, este pode resultar no acúmulo de horas extras, horas ociosas ou ser zerado. Estes valores são multiplicados pelos seus respectivos pesos w_1 e w_2 , utilizados para relacionar a expressão com os custos reais da empresa. O custo $f(s)$ de uma solução s é dado pelo somatório das horas extras e as ociosidades ponderadas de cada tripulação. Neste trabalho foram utilizados pesos unitários, atribuindo a mesma importância para ambos os fatores. A expressão (1) fornece a função de custo utilizada na implementação da metaheurística.

$$f(s) = \sum_{i \in \text{tripulações de } s} w_1 \times \text{hora_extra}_i + w_2 \times \text{ociosidade}_i \quad (1)$$

4.4 Solução inicial

A solução inicial, assim como todas as soluções geradas pela metaheurística, considera o padrão de folgas 5/1. Como a escala é gerada para um dado período, no início desse período, cada tripulação já conta com um número de dias trabalhados anteriormente sem folgar. Como estes dados não são fornecidos pela empresa, o número de dias trabalhados anteriormente é gerado aleatoriamente para cada tripulação e armazenado em um vetor, denominado $\text{dias_trab}[]$, o qual se torna um dado de entrada do problema. A partir do número de dias trabalhados anteriormente é feita

a alocação de todas as folgas da tripulação para todo o período, uma vez que as folgas atendem um padrão fixo. Portanto, o número mínimo de folguistas depende do vetor $\text{dias_trab}[]$. Assim, foram gerados diferentes vetores de dias trabalhados anteriormente e foi armazenado aquele que levou ao menor número de folguistas na escala.

Uma solução do PRT pode ser representada por uma matriz, sendo que as linhas correspondem aos m rodízios, cada rodízio representando uma sequência de jornadas a serem executadas por uma dada tripulação ao longo do período. Desta forma, neste trabalho, os termos rodízio e tripulação são sinônimos, e uma solução para o PRT é um rodízio completo, ou uma escala de todas as tripulações para o período. As colunas correspondem aos n dias do horizonte de planejamento e o valor na posição da matriz representa a jornada a ser executada por aquele rodízio no respectivo dia. Ou seja, uma solução pode ser dada por $R_{m \times n} = \text{rodízio}[t, d]$, onde $\text{rodízio}[t, d] = j$, sendo que $t \in \text{rodízios da solução}$, $d \in \text{dias do período}$ e $j \in \text{jornadas a serem executadas no dia } d$.

O procedimento de construção da solução inicial é dividido em duas etapas. Inicialmente, a partir do vetor $\text{dias_trab}[]$, é feita uma atribuição sequencial das jornadas, sendo que o rodízio i recebe a jornada i no primeiro dia, isto é, $\text{rdz}[i, 1] = i$, $\forall i \in \{\text{jornadas do dia 1}\}$. Caso o rodízio i esteja de folga no dia 1, então é acrescido mais um rodízio à escala (mais uma linha na matriz) que representa uma tripulação do tipo folguista para cobrir a folga mencionada. Esta tripulação passa a fazer parte da solução e pode ser utilizada para cobrir folgas nos outros dias do período. Uma vez definida a escala até o dia d , é realizada a alocação do dia $d+1$, que segue um procedimento guloso. Entretanto, para garantir a viabilidade da solução, os rodízios são colocados em ordem crescente pelo número de possibilidades de combinação da jornada realizada no dia d com as jornadas a serem realizadas no dia $d+1$. Assim, inicialmente são definidas as jornadas do dia $d+1$ para os rodízios com menores possibilidades de combinação, deixando para o final do processo a definição daqueles rodízios com maiores possibilidades de combinação. O custo de atribuição do rodízio i à jornada j do dia $d+1$ é dado pela expressão $c_{ij} = |cr_i + c_j|$, onde cr_i é a soma das horas extras (positivas) com as horas ociosas (negativas) de todas

as jornadas atribuídas ao rodízio i até o dia d e c_j é o custo da jornada j no dia $d+1$, dado pelas horas extras ou ociosas da referida jornada. Finalmente, percorrendo os rodízios reordenados, é feita a atribuição de menor custo dentre todas as possibilidades. Este procedimento se repete até o final do período. Caso o rodízio i tenha completa-

do 5 dias de trabalho no dia d , então não é atribuída jornada a este rodízio no dia $d+1$. Se for necessário, um novo rodízio do tipo folguista é acrescentado à solução e este pode ser utilizado nos demais dias do período. O pseudo-código do procedimento de construção da solução inicial é apresentado no Algoritmo 3 a seguir.

Procedimento Solução Inicial(vetor dias_trab[1..n], matriz rdz[1..n, 1..m])

Início:

1. Para $t = 1$ até m faça:
2. Se dias_trab[t] = 5
3. Então rdz[t , 1] := 0;
4. acrescente uma tripulação à escala e atribua a jornada t a ela no primeiro dia
5. do período;
6. Senão rdz[t , 1] := t ;
7. Fim_para;
8. Atualizar m ;
9. Para $d = 2$ até n faça:
10. Ordenar as jornadas de rdz[t , $d-1$] pela dificuldade de alocação destas com as
11. jornadas do dia d ;
12. Para $t = 1$ até m faça:
13. Se não estiver prevista folga para o rodízio t no dia d
14. Então $i :=$ rdz[t , $d-1$]; // armazena a jornada realizada pelo rodízio t em $d-1$
15. $S :=$ jornadas do dia d que podem ser alocadas à tripulação que executou
16. a jornada i no dia $d-1$;
17. rdz[t , d] := j tal que c_{ij} seja mínimo $\forall j \in S$; // c_{ij} descrito anteriormente
18. Fim_se;
19. Se houver jornada do dia d que não foi coberta com as tripulações em serviço
20. Então acrescentar mais tripulações à escala e alocar as jornadas restantes;
21. Fim_para;
22. Fim_para;
23. Retorne a solução rdz[[]];

Fim

Algoritmo 3 - Procedimento para a construção da solução inicial do PRT

4.5 Estruturas de vizinhança

Uma das etapas fundamentais do VNS é a determinação das estruturas de vizinhança. As estruturas de vizinhança devem ser escolhidas para explorar, de forma eficiente, o espaço de soluções, permitindo que regiões cada vez mais distantes da solução corrente sejam exploradas. A estrutura de vizinhança determina a estratégia de busca no espaço de soluções. Quanto maior a vizinhança, melhor será a qualidade da solução

ótima local. No entanto, vizinhanças maiores requerem um tempo de pesquisa elevado. Portanto, uma vizinhança maior, não implica em uma heurística melhor, exceto se a vizinhança for explorada de forma eficiente. Neste trabalho foram exploradas três diferentes estratégias de modificação da solução corrente. As estruturas de vizinhança estão associadas à quantidade de modificações realizadas a partir de cada estratégia.

4.5.1 Primeira estrutura de vizinhança

A primeira estrutura de vizinhança consiste na escolha aleatória de duas tripulações diferentes i e j , além de um dia d_1 . A jornada da tripulação i é trocada com a jornada da tripulação j no dia d_1 . Um novo dia d_2 diferente do anterior é sorteado, permanecendo fixas as duas tripulações i e j . As jornadas das duas tripulações são trocadas para este novo dia. E assim são realizadas k trocas de jornadas entre as mesmas tripulações. Este procedimento fixa duas tripulações e realiza as trocas em k dias diferentes, caracterizando k estruturas de vizinhança. Uma vez realizado um *movimento*, ou seja, um conjunto de trocas, é analisado o custo final da nova solução com base na função de custo (1). Caso as trocas representem uma diminuição do custo elas são efetuadas, caso contrário, são sorteadas novas tripulações e reinicia-se o procedimento.

4.5.2 Segunda estrutura de vizinhança

A segunda estrutura de vizinhança consiste no sorteio de um dia d , permanecendo este fixo e os pares de tripulações são escolhidos aleatoriamente. Os pares de jornadas candidatas à troca são obtidos até atingir o número total de k trocas. Caso este conjunto de trocas represente uma melhora na função de custo, elas são realizadas, caso contrário, o processo é reiniciado com o sorteio de um outro dia.

4.5.3 Terceira estrutura de vizinhança

Uma terceira estrutura de vizinhança foi implementada com a união das duas estruturas de vizinhança descritas anteriormente. Neste caso foi utilizada a primeira estrutura de vizinhança para gerar as perturbações da solução corrente descrita na linha 7 do Algoritmo 1. A segunda estrutura de vizinhança foi utilizada para realizar a busca local do VND na linha 4 do Algoritmo 2.

Empregando as três estruturas de vizinhança, foram obtidas três versões da metaheurística VNS para a resolução do PRT. Estas versões fo-

ram utilizadas para resolver um problema real, cujos resultados são apresentados a seguir.

5 RESULTADOS OBTIDOS

As versões do VNS foram implementadas na linguagem C/C++ em um computador pessoal com processador Intel Core i3 com 2 GB de memória RAM. O algoritmo foi desenvolvido com a opção de resolver o problema utilizando três estruturas de vizinhança diferentes e um número máximo de trocas calibrado empiricamente.

Foram conduzidos testes iniciais para calibrar o parâmetro k de cada uma das estruturas de vizinhança. O número de execuções realizadas para cada valor de k foi igual a 5, sendo que cada execução teve um tempo de duração de uma hora. Dessa forma foi possível obter, para cada estrutura de vizinhança, a melhor solução, a solução média e o desvio padrão para o valor da função de avaliação. Com os parâmetros que produziram as melhores soluções foram realizados outros dez testes, os quais constituíram as diferentes soluções finais para o PRT.

5.1 Dados de entrada

Os dados de entrada são as jornadas previstas para serem executadas nos dias úteis, nos sábados e nos domingos do período de planejamento. O período estudado é típico sem a ocorrência de feriados. A Tabela 1 apresenta um resumo dos dados de entrada, sendo que em cada dia útil são realizadas 104 jornadas de trabalho, das quais 4 são do tipo dupla pegada e 13 do tipo noturno. Analisando as 104 jornadas, são contabilizadas 62 horas e 46 minutos de horas extras e 78 horas e 36 minutos de horas ociosas. O objetivo do modelo de otimização consiste em compensar as horas extras com as horas ociosas intercalando as jornadas de trabalho ao longo do período de planejamento.

Tabela 1 - Caracterização dos dados de entrada

Tipo de Dia	Motoristas	Dupla-pegada	Noturno	Horas-extras	Ociosidade
Dia útil	104	4	13	62:46	78:36
Sábado	70	11	0	45:37	26:54
Domingo	53	9	0	27:41	16:01

Foram considerados apenas os motoristas, pois estes são responsáveis por retirar o veículo da garagem no início do dia e retornar com o veículo à garagem no final da operação.

5.2 Resultados obtidos utilizando a primeira estrutura de vizinhança

Os testes foram iniciados utilizando a primeira estrutura de vizinhança. Nesta estrutura de vizinhança são fixadas duas tripulações escolhidas aleatoriamente, e realizadas as trocas das jornadas em k dias diferentes. Para escolher o valor de k foram realizadas cinco execuções da metaheurística VNS com diferentes valores para

k . A Tabela 2 apresenta os resultados para k assumindo os valores iguais a 5, 10, 15, 20, 30 e 40. Estes valores foram escolhidos de forma empírica, tendo em vista o efeito da modificação na solução em função do parâmetro k . Como o período de planejamento foi de 7 semanas, ou seja, 49 dias, k foi limitado a 40 para que o processo de perturbação ainda mantivesse características da solução corrente. Na Tabela 2 são exibidas a melhor solução e a média dentre todas as soluções obtidas. A linha %DP apresenta a porcentagem do desvio padrão em relação à média do valor da função de avaliação.

Tabela 2 - Resultados considerando a primeira estrutura de vizinhança

k	5	10	15	20	30	40
Melhor solução	100.733,00	101.435,00	98.719,00	97.589,00	96.690,00	97.325,00
Média	100.979,40	101.829,20	99.373,20	98.429,40	96.956,40	98.184,60
%DP	0,25	0,63	0,34	0,80	0,69	0,75

Tendo em vista os resultados apresentados na Tabela 2, foi escolhido o valor para o parâmetro $k = 30$. Para este valor de k foram realizar 10 execuções do VNS, cujos resultados da melhor solução e da solução média estão detalhados na Tabela 3.

A metaheurística se mostra robusta uma vez que o desvio padrão está na ordem de 1% da média. Portanto as soluções são muito próximas umas das outras, apesar do fator de aleatoriedade presente no VNS.

Tabela 3 - Resultados considerando a primeira estrutura de vizinhança e $k = 30$

Primeira Estrutura de Vizinhança com $k = 30$			
	Horas Extras	Horas Ociosas	Custo Total
Melhor	632:11	967:39	95.990
Média	645:21	985:52	97.873
%DP			1,22

5.3 Resultados obtidos utilizando a segunda estrutura de vizinhança

Na segunda estrutura de vizinhança é escolhido um dia aleatoriamente e são sorteados k pares de jornadas a serem trocas entre si. Para esta estrutura o valor de k também variou entre 5 e 30

e os resultados são apresentados na Tabela 4. Neste caso a melhor solução foi obtida para $k = 20$, valor escolhido para calibrar o algoritmo e realizar mais um conjunto de testes.

Tabela 4 - Resultados considerando a segunda estrutura de vizinhança

k	5	10	15	20	30
Melhor solução	96.129,00	96.266,00	96.115,00	95.958,00	97.043,00
Média	97.343,20	97.575,80	96.973,00	97.129,00	98.459,38
%DP	1,20	0,83	0,70	0,91	0,84

Os resultados devido à utilização desta segunda estrutura, apresentados na Tabela 5, são

muito similares aos resultados obtidos com a primeira estrutura. Porém, neste caso a porcentagem

do desvio padrão em relação à média é ligeiramente menor. Portanto o VNS se mostra ainda mais robusto do que no caso anterior.

Tabela 5 - Resultados considerando a segunda estrutura de vizinhança e $k = 20$

Segunda Estrutura de Vizinhança com $k = 20$			
	Horas Extras	Horas Ociosas	Total
Melhor	628:03	969:32	95.855,00
Média	645:08	985:42	97.850,00
%DP			0,83

5.4 Resultados obtidos utilizando a terceira estrutura de vizinhança

Neste caso a primeira estrutura de vizinhança foi utilizada para gerar a solução vizinha à solução corrente e a segunda estrutura de vizinhança foi utilizada para realizar a busca local no VND. A melhor solução apresentada na Tabela 6 foi obtida para o valor de $k = 30$. Assim, novos testes foram conduzidos tendo este valor para o parâmetro k do método.

Os resultados obtidos com esta estrutura de vizinhança foram os melhores dentre as três estru-

turas implementadas neste estudo. Na Tabela 7 há um detalhamento da melhor solução e da solução média das 10 execuções. O desvio padrão foi superior aos casos anteriores, entretanto a melhor solução obtida com a terceira estrutura levou a uma redução de aproximadamente 22% das horas extras das melhores soluções obtidas com as duas estruturas de vizinhança anteriores. Considerando as horas extras, houve uma redução de aproximadamente 15%.

Tabela 6 - Resultados considerando a terceira estrutura de vizinhança

K	5	10	15	20	30	40
Melhor solução	97.508,00	96.536,00	93.823,00	85.333,00	84.377,00	89.172,00
Média	101.091,40	98.794,00	97.755,40	95.173,80	91.524,20	96.504,88
%DP	1,90	1,70	2,31	6,77	7,50	7,75

Tabela 7 - Resultados considerando a terceira estrutura de vizinhança e $k = 30$

Terceira Estrutura de Vizinhança $k=30$			
	Horas Extras	Horas Ociosas	Custo Total
Melhor	490:40	832:09	79.369,00
Média	598:56	932:41	91.897,00
%DP			7,82

A terceira estrutura pode não ser tão robusta quanto às estruturas anteriores, uma vez que o seu desvio padrão é de 7,82%. Entretanto esta combinação foi a que produziu os melhores resultados. O valor do desvio padrão elevado pode estar associado ao fato desta versão do VNS explorar um espaço mais amplo de soluções, o que leva a uma maior variabilidade nos ótimos locais encontrados. Na Tabela 6 pode ser observado que o valor do desvio padrão cresce juntamente com o crescimento do valor do parâmetro k .

Este, por sua vez, está diretamente associado à perturbação da solução corrente no procedimento de geração do próximo vizinho a ser pesquisado. Desta maneira, diferentes regiões foram

exploradas, sendo obtidos valores mais diversificados para a função objetivo. Portanto, pode-se supor que foi a variabilidade na solução vizinha que propiciou a obtenção dos melhores resultados para o problema, o que tem como contrapartida os valores elevados para o desvio padrão das soluções.

5.5 Comparação das soluções heurísticas com a solução da empresa

Os resultados obtidos neste trabalho foram comparados com a solução adotada pela empresa no período estudado. Na Tabela 8 são apresentadas as melhores soluções obtidas por cada versão do VNS. Os resultados mostram que a aplicação

do VNS permitiu uma redução significativa no custo da escala dos motoristas. A redução nas horas extras varia entre 43% e 56% em relação à solução da empresa. Para as horas ociosas, a redução variou entre 43% e 51% da solução da empre-

sa. O número de motoristas foi constante para as diferentes versões e houve uma redução de 50% em relação ao total de motoristas utilizado pela empresa.

Tabela 8 - Comparação dos resultados obtidos com os dados da empresa

Solução	Motoristas	Horas-extras	Ociosidade
Empresa	244	1.106:39	1.695:52
VNS + E1	122	632:11	967:39
VNS + E2	122	628:03	969:32
VNS + E3	122	490:40	832:09

Apesar das reduções obtidas pelo VNS em suas diferentes versões, é possível observar que os dados da empresa contam com motoristas que trabalharam apenas alguns dias no período considerado. Isso pode ser explicado, segundo a empresa, devido a período de férias, demissões e afastamentos por motivo de saúde. Mas a empresa assegura que este número varia entre 10% e 20% dos motoristas. Assim, fica claro que a utilização de métodos de otimização na elaboração da escala dos motoristas do setor do transporte público pode levar à redução dos custos da empresa e uma melhor divisão da carga de trabalho das tripulações.

6 CONCLUSÕES

Este trabalho apresentou uma metaheurística VNS para resolver o Problema de Rodízio de Tripulações do Sistema de Transporte Público. A metaheurística foi modelada para resolver um caso prático da realidade brasileira, considerando suas restrições legais e operacionais. Neste modelo foi adotado o padrão de folga fixa denominado 5/1, no qual o motorista trabalha cinco dias consecutivos e folga no sexto dia. Este padrão é adotado por cumprir a restrição trabalhista de que as tripulações devem folgar pelo menos um domingo em um período de 7 semanas.

A metaheurística VNS foi implementada em três versões considerando diferentes estruturas de vizinhança. Na primeira estrutura de vizinhança são fixadas duas tripulações escolhidas aleatoriamente, realizando a troca de suas jornadas em k dias diferentes. Na segunda estrutura é fixado um dia, variando os k pares de jornadas a serem trocas. E a terceira vizinhança, que é uma combinação das anteriores, utiliza a primeira estrutura para gerar um vizinho da solução corrente e faz a

busca local na segunda vizinhança, tendo em vista a metaheurística implementada.

O algoritmo foi testado com os dados de uma empresa brasileira e mostraram que as soluções produzidas nas três versões apresentam reduções que variam entre 43% e 56% das horas extras contidas na solução da empresa. O número de motoristas foi reduzido em 50% e é igual para todas as versões. A versão mais eficiente foi aquela que combinou a primeira e a segunda estrutura de vizinhança. Isso pode ser explicado pelo fato de que a combinação permite explorar melhor o espaço de soluções.

A continuidade deste trabalho pode se dar em diferentes direções. Um primeiro aprimoramento deve levar em conta o vetor de dias trabalhados antes do início do período, vetor este escolhido aleatoriamente. A definição deste vetor é fundamental para minimizar o número total de tripulações no padrão de folgas do tipo 5/1. Uma outra possível continuidade do trabalho deve considerar o padrão de folgas variáveis, que pode levar a uma redução no número de tripulações necessárias.

AGRADECIMENTOS

Os autores agradecem ao CNPq, à FAPEMIG e à UFOP pelo apoio recebido durante o desenvolvimento deste trabalho.

REFERÊNCIAS

- Ahuja, R. K.; Magnanti, T. L. e Orlim, J. B. (1993) *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey.
- Bianco, L.; Bielli, M.; Mingozzi, A.; Ricciardelli, S. e Spadoni, M. (1992) A heuristic procedure for the crew rostering problem. *European Journal of Operations Research*, v. 58, n. 2, p. 272-283. DOI: [10.1016/0377-2217\(92\)90213-S](https://doi.org/10.1016/0377-2217(92)90213-S).

- Caprara, A.; Monaci, M. e Toth, P. (2003). Models and algorithms for a staff scheduling problem. *Mathematical Programming*, 98(1-3), 445–476. DOI: [10.1007/s10107-003-0413-7](https://doi.org/10.1007/s10107-003-0413-7)
- Carrara, P. e Gallo, G. (1984) A multi-level bottleneck assignment approach to the bus drivers rostering problem. *European Journal of Operational Research*, v. 16, n. 2, p. 163–173. DOI: [10.1016/0377-2217\(84\)90071-7](https://doi.org/10.1016/0377-2217(84)90071-7).
- Ernst, A. T.; Jiang, H.; Krishnamoorthy, M. e Sier, D. (2004a) Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, v. 153, p. 3-27. DOI: [10.1016/S0377-2217\(03\)00095-X](https://doi.org/10.1016/S0377-2217(03)00095-X).
- Ernst, A. T.; Jiang, H.; Krishnamoorthy, M.; Owens, B. e Sier, D. (2004b) An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, v. 127, p. 21-144. DOI: [10.1023/B:ANOR.0000019087.46656.e2](https://doi.org/10.1023/B:ANOR.0000019087.46656.e2).
- Lin, S. e Kernighan, B. W. (1973) An effective heuristic for the traveling salesman problem. *Operations Research* 21, 498–516, 1973. DOI: [10.1287/opre.21.2.498](https://doi.org/10.1287/opre.21.2.498).
- Mayrink, V. T. M. e Silva, G. P. (2013) Optimization of crew rostering assignment for public transport systems. *Journal of Transport Literature*, v. 7, p. 192-213. DOI: [10.1590/S2238-10312013000300009](https://doi.org/10.1590/S2238-10312013000300009).
- Mesquita, M.; Moz, M.; Paias, A.; Paixão, J.; Pato, M. e Respício, A. (2011) A new model for the integrated vehicle-crew-rostering problem and a computational study on rosters. *Journal of Scheduling*, v. 14, p.319–334. DOI: [10.1007/s10951-010-0195-8](https://doi.org/10.1007/s10951-010-0195-8).
- Mladenović, N. e Hansen, P. (1997) Variable Neighborhood Search. *Computers and Operations Research* v. 24, n 11, p. 1097-1100. DOI: [10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2).
- Moz, M.; Respício, A. e Pato, M. V. (2009). Bi-objective evolutionary heuristics for bus driver rostering. *Public Transport*, v. 1, n. 3, p. 189–210. DOI: [10.1007/s12469-009-0013-x](https://doi.org/10.1007/s12469-009-0013-x).
- Nurmi, K.; Kyngas, J. e Post, G. (2012). Driver rostering for bus transit companies. *Engineering Letters*, v. 19, n. 2, p. 125–132.
- Pato M. e Moz, M. (2008) Solving a bi-objective nurse rostering problem by using a utopic Pareto genetic heuristic. *Journal of Heuristics*, v. 14, p. 359–374. DOI: [10.1007/s10732-007-9040-4](https://doi.org/10.1007/s10732-007-9040-4).
- Toffolo, T. A.; Souza, M. J. F. e Silva, G. P. (2005) Resolução do Problema de Rodízio de Tripulações de Ônibus Urbano via *Simulated Annealing* e *Iterated Local Search*. *Anais do XIX Congresso de Pesquisa e Ensino em Transportes*, ANPET, v. 2, p. 657-668.
- Yunes, T. H.; Moura, A. V. e Souza, C. C. (2005). Hybrid Column Generation Approaches for Urban Transit Crew Management Problems. *Transportation Science*, v. 39, n. 2, p. 273-288. DOI: [10.1287/trsc.1030.0078](https://doi.org/10.1287/trsc.1030.0078).