

O USO DA HEURÍSTICA *ADAPTIVE LARGE NEIGHBORHOOD SEARCH* PARA RESOLVER O PROBLEMA DE PROGRAMAÇÃO DE TRIPULAÇÕES

Leandro do Carmo Martins

Universidade Federal de Ouro Preto – Departamento de Computação
Campus Morro do Cruzeiro, Ouro Preto, MG – Brasil - 35400-000
leandro.cm@live.com

Gustavo Peixoto Silva

Universidade Federal de Ouro Preto – Departamento de Computação
Campus Morro do Cruzeiro, Ouro Preto, MG – Brasil - 35400-000
gustavo@iceb.ufop.br

RESUMO

Este trabalho apresenta uma adaptação da heurística *Adaptive Large Neighborhood Search* (ALNS) para resolver o Problema de Programação de Tripulações (PPT) de Ônibus Urbano. O PPT consiste em minimizar o total de tripulações que conduzirão uma frota em operação diária. A solução é um conjunto de jornadas executadas pelas tripulações ao longo do dia. As jornadas devem satisfazer leis trabalhistas, acordos sindicais e as regras operacionais da empresa. O PPT é NP-difícil e normalmente é resolvido por metaheurísticas. Assim, a ALNS tem como objetivo minimizar custos fixos e variáveis do PPT, satisfazendo todas as suas restrições. Os custos fixos são representados pelo número de jornadas e os custos variáveis pelo total de horas extras. A implementação foi testada com dados reais de uma empresa que opera em Belo Horizonte, MG, e gerou soluções superiores às adotadas pela empresa e que se mostraram competitivas em relação a outros métodos da literatura.

PALAVRAS-CHAVE. Problema de Programação de Tripulações, Transporte Público, *Adaptive Large Neighborhood Search*.

Área principal: Logística e Transportes, Metaheurísticas

ABSTRACT

This work presents an adaptation of *Adaptive Large Neighborhood Search* (ALNS) to solve the Crew Scheduling Problem (CSP) for urban buses. The CSP consists of determining the number of crews to drive the fleet in daily operation. The solution is a set of work shifts performed by each crew member. The shifts must meet several constraints due to labor laws, union agreements and the operational rules of the company. This problem is NP-hard and it is normally solved through metaheuristics. Therefore, the ALNS has as objective to minimize fixed and variable costs from the CSP, satisfying all the constraints. The fixed costs are represented by the number of shifts and the variable costs by the overtime amount. The implementation was tested with real data from a company operating in Belo Horizonte city and generated better solutions than those adopted by the company and was competitive compared to other literature methods.

KEYWORDS. Crew Scheduling Problem, Public Transport, *Adaptive Large Neighborhood Search*.

Main area: Logistic & Transport, Metaheuristics

1. Introdução

O planejamento do Sistema de Transporte Público Urbano é um processo complexo e dividido em etapas, que envolve tanto as empresas de transporte urbano quanto o poder público. Este planejamento é composto por cinco etapas: *i*) a definição das rotas dos ônibus, *ii*) a criação dos quadros de horários das linhas para todos os dias da semana, *iii*) a programação dos veículos que realizarão as viagens, *iv*) a programação diária das tripulações que conduzirão a frota em operação, e *v*) o rodízio das tripulações ao longo de um dado período de planejamento (Bodin *et al.*, 1983).

Dentre as etapas do planejamento, as empresas devem realizar a programação dos veículos, a programação das tripulações e o rodízio das tripulações. Assim, as empresas de transporte público devem definir o tamanho de suas frotas, o número de tripulações (dupla motorista e cobrador) e os períodos de trabalho e de folga destas tripulações de forma que todas as viagens sejam executadas dentro da legalidade e com o menor custo possível. Esta é uma tarefa de grande complexidade para a qual podem ser utilizados métodos de otimização na sua resolução (Wren, 2004).

A realização das viagens previstas leva à definição de uma escala diária de trabalho para os motoristas e cobradores, denominada *Problema de Programação de Tripulações* (PPT). O PPT consiste em determinar o número mínimo de tripulações e atribuir a elas as viagens que devem ser executadas em um determinado dia de trabalho de tal forma que cada viagem seja atribuída à jornada de uma tripulação e que não haja sobreposição entre as viagens realizadas pelas tripulações. O objetivo é minimizar o número de tripulações e o total de horas extras contidas na escala diária (Bodin *et al.*, 1983).

O PPT tem como dados de entrada a programação dos veículos, previamente definida, para um determinado tipo de dia, que pode ser um dia útil, um sábado ou um domingo/feriado. A programação dos veículos define os *blocos dos veículos*, ou seja, o conjunto das viagens a serem executadas por cada veículo da frota. Os blocos são divididos em *tarefas*, que são um conjunto de viagens consecutivas que devem ser executadas pela mesma tripulação, visto que não há condições suficientes, entre essas viagens, para que haja a troca da tripulação. As condições necessárias para a troca de duas tripulações definem uma *oportunidade de troca* (OT), que consiste em um intervalo de tempo mínimo entre duas viagens (*tempo de troca*), o qual deve ocorrer em um local apropriado para a execução da troca (*ponto de troca*) (Silva *et al.*, 2006). Essas tarefas são caracterizadas por um local inicial e final, e um horário de início e de término, que representam onde e quando a tarefa deve ser inicializada e finalizada. Além disso, as tarefas são caracterizadas por um veículo de operação e sua duração é dada pela diferença entre seu horário final e inicial. Um conjunto de tarefas executadas por uma tripulação é denominado *jornada*. Desta forma, uma tripulação está associada unicamente a uma jornada e vice-versa, portanto são considerados sinônimos neste trabalho.

As jornadas podem ser classificadas em *simples* ou *dupla pegada*. Uma jornada do tipo dupla pegada é aquela em que o trabalho é dividido em duas etapas, com intervalo superior a duas horas entre a primeira e a segunda etapa de trabalho. Cada etapa é denominada de *pegada*. Isso ocorre, geralmente, devido aos picos de demanda de viagens, comuns nos dias úteis. O intervalo superior a duas horas não é considerado como hora trabalhada e não faz parte da remuneração. Uma jornada simples não apresenta uma interrupção desta magnitude e sua duração é dada pela diferença entre seu horário de término e de início. Por outro lado, a duração de uma jornada de dupla pegada é dada pela diferença entre seu horário de término e de início, subtraindo ainda o intervalo superior a duas horas que caracteriza o regime. Independente do tipo de jornada, as horas extras correspondem ao tempo trabalhado que excede o tempo de duração normal de uma jornada diária.

A construção das jornadas deve satisfazer um conjunto de restrições legais e operacionais, o que torna o PPT um problema de difícil resolução. Fischetti *et al.* (1987) mostram que este problema é do tipo *NP-Completo*. Na resolução do PPT, para o caso estudado, foram consideradas as seguintes restrições: 1) as jornadas tem uma remuneração fixa por seis horas e quarenta minutos de trabalho, que define a duração normal de trabalho; 2) as jornadas não podem ter mais que duas horas extras de

trabalho por dia; 3) o intervalo de tempo entre o fim de uma jornada e o seu início no dia seguinte deve ser de pelo menos onze horas; 4) a troca de veículos, feito por uma tripulação durante a operação, pode acontecer somente em locais pré-determinados; 5) o número de troca de veículos, realizadas em uma jornada, é limitado e dado a priori.

Neste trabalho será apresentada a resolução do PPT por meio da heurística *Adaptive Large Neighborhood Search* (ALNS). A ALNS é caracterizada pela destruição e reparo da solução corrente, através de várias heurísticas de remoção e inserção que são utilizadas na mesma busca. A escolha das heurísticas utilizadas a cada iteração é realizada através de um peso ajustado dinamicamente que se baseia na eficiência de cada método ao longo da busca.

A implementação desenvolvida foi testada com dados reais de uma empresa de médio porte que atua na cidade de Belo Horizonte, MG, e as soluções alcançadas superaram as soluções adotadas na prática pela empresa e mostraram-se competitivas em relação a outros métodos da literatura.

Este trabalho está dividido da seguinte forma: na Seção 2 é apresentada uma breve revisão bibliográfica sobre o PPT. Na Seção 3 é descrito como a ALNS foi implementada para resolver o PPT. Na Seção 4 são apresentados os resultados obtidos nos experimentos computacionais e a discussão destes resultados. Finalmente, na Seção 5, são apresentadas as conclusões do trabalho.

2. Revisão Bibliográfica

A abordagem exata mais explorada é aquela que formula o PPT como um problema de cobertura ou de particionamento de conjuntos (*set covering* ou *set partitioning model*) e utiliza a técnica de geração de colunas para resolvê-lo (Smith e Wren, 1988, Desrochers e Soumis, 1989, Fores *et al.*, 1999). A variedade de trabalhos deriva das diferentes possibilidades de se encontrar uma solução inteira a partir da solução do Problema de Programação Linear. As técnicas mais exploradas são o *branch-and-bound*, *branch-and-price* e a relaxação *lagrangeana*, as quais permitem formas alternativas de implementação, tendo em vista as características do problema (Barnhart *et al.*, 1998, Friberg e Haase, 1999).

Lourenço *et al.* (2001) formulam o PPT como um problema de cobertura, considerando várias funções objetivo. Os autores apresentam metaheurísticas multi-objetivas para resolver o PPT visto que na prática vários objetivos devem ser considerados, sendo que muitos deles são conflitantes. As metaheurísticas propostas são baseadas na Busca Tabu e nos Algoritmos Genéticos, e ambos têm como procedimento interno o GRASP, uma metaheurística mono-objetivo. Para todos os problemas considerados, a Busca Tabu multi-objetivos e os Algoritmos Genéticos (AG) multi-objetivos geraram bons resultados dentro de um tempo razoável, competitivos com um método de programação linear que obtém a solução ótima para a maioria dos casos, exceto para problemas muito grandes.

Chen e Shen (2013) apresentam uma nova estratégia de geração de colunas para tratar o PPT. Nos algoritmos tradicionais, as colunas a serem adicionadas ao problema em futuras iterações do problema principal são determinadas pela resolução de subproblemas, tais como o problema de caminho mínimo com recursos limitados, geralmente muito demorados. Por essa razão, a ideia principal da abordagem é que um conjunto razoavelmente grande de jornadas potenciais, chamado *shift-pool*, é pré-compilado utilizando características específicas do problema. Dentre as características analisadas, consideram-se fatores que aumentam o tempo de trabalho e reduzam o tempo ocioso. Durante o processo de geração de colunas, o subproblema é resolvido até que não existam mais colunas com reduções de custo na *shift-pool*. Esta ideia permite reduzir o tempo de processamento para resolver o PPT. Nos experimentos são consideradas instâncias de até 701 viagens, 55 blocos de veículos, e duas garagens. Os resultados mostram que o algoritmo proposto supera a geração de colunas tradicional.

Souza (2014) apresenta um modelo híbrido de Programação Linear Inteira em conjunto com a metaheurística *Late Acceptance Hill Climbing* (LAHC) para resolver o PPT. O problema é

modelado como um problema de cobertura, mas ao contrário dos modelos clássicos de particionamento ou de cobertura, onde jornadas promissoras devem ser geradas e informadas *a priori*, o modelo é composto por uma série de restrições que permite a geração de soluções viáveis e a resolução do PPT a partir das tarefas a serem realizadas, obtidas diretamente dos dados de entrada. A LAHC é utilizada com o objetivo de resolver problemas de maiores dimensões. Nestes casos, partindo de uma solução viável gerada pela metaheurística, os componentes da solução são particionados em conjuntos menores, ou seja, subproblemas de dimensões menores que podem ser melhorados pelo método exato. O modelo foi testado com problemas baseados em dados reais e se mostraram eficientes na geração das soluções, mas em um tempo elevado de processamento.

Silva e Silva (2015) aplicam a metaheurística *Guided Local Search* (GLS) para resolver o PPT. A GLS é caracterizada pela construção de uma sequência de soluções, realizando uma busca local na solução corrente, e se diferencia das demais metaheurísticas por penalizar as características indesejáveis na solução corrente como forma de escapar de ótimos locais. Desta forma, ao encontrar um ótimo local, a metaheurística seleciona alguns atributos da solução encontrada e os penalizam, fazendo que a solução deixe de ser um ótimo local e que a busca possa se mover para outro ponto de ótimo local. Esse processo se repete até que algum critério de parada seja satisfeito. A GLS foi combinada com o método de busca local *Variable Neighborhood Descent*, produzindo resultados similares aos apresentados na literatura, com possibilidades de melhoria.

Song *et al.* (2015) tratam o PPT com um AG aperfeiçoado com recombinação de genes. A recombinação de genes é responsável pela reconstrução de jornadas, sendo capaz de reduzir significativamente o número de jornadas na solução. Os autores utilizam listas tabu e não tabu para a construção das soluções iniciais, que representam se as tarefas já foram ou não alocadas em alguma jornada. O método padrão da roleta foi adotado para selecionar os indivíduos promissores para o *crossover*, caracterizado por dois pontos de corte. Como a operação de *crossover* pode resultar em cromossomos com tarefas repetidas, as mesmas devem ser removidas. Assim, as tarefas duplicadas de uma jornada são removidas e reinseridas no final do novo cromossomo, construindo uma nova jornada. O processo de mutação consiste na reconstrução de n jornadas escolhidas aleatoriamente, ou seja, são divididas em segmentos que são reconstruídos e substituídos. Os autores mostraram que o método proposto foi capaz de reduzir mais de 30 jornadas quando tem-se um grande número de tarefas, caracterizando-se como um algoritmo de performance computacional muito boa.

A heurística *Adaptive Large Neighborhood Search* (ALNS) foi proposta por Ropke e Pisinger (2006a) para resolver o Problema de Coleta e Entrega com Janelas de Tempo. A ALNS é uma extensão da *Large Neighborhood Search* (LNS), sendo que a LNS parte de uma solução inicial viável qualquer e tem como mecanismo de busca a remoção e a inserção de elementos da solução, que definem a robustez e o desempenho da heurística. No ALNS, as heurísticas de remoção e inserção do LNS são escolhidas a partir das estatísticas recolhidas durante o processo de busca. São apresentadas três heurísticas de remoção: *Shaw*, *Random* e *Worst* e duas heurísticas de inserção: *Greedy* e *Regret*. Em cada iteração, uma heurística de remoção e uma de inserção é escolhida independentemente e o par é aplicado. Todas heurísticas são utilizadas na busca e a seleção do par é feita tendo em vista os pesos individuais de cada heurística, os quais são ajustados automaticamente utilizando estatísticas das iterações anteriores. Os pesos medem o sucesso da heurística e são atualizados a cada iteração. Quando maior a pontuação, mais bem sucedida é a heurística. A metodologia proposta foi capaz de melhorar as melhores soluções conhecidas na literatura para mais de 50% dos problemas, indicando a vantagem de utilizar várias heurísticas de remoção e inserção concorrentemente ao invés de apenas um par de heurísticas.

Os mesmos autores integraram, posteriormente, novas heurísticas de remoção ao ALNS. Três novas heurísticas de remoção: *Cluster*, *Neighbor Graph* e *Request Graph* foram propostas e o problema passou a cobrir uma classe de Problemas de Roteamento de Veículos (Ropke e Pisinger, 2006b). Posteriormente, abrangendo diferentes variações do Problema de Roteamento de Veículos,

os autores propuseram três novas heurísticas de remoção: *Time-oriented*, *Node-pair removal* e *Historical request-pair* (Pisinger e Ropke, 2007). Semelhante ao PPT, como o problema de coleta e entrega com janela de tempo é um problema *NP*-difícil, levanta-se a hipótese se a ALNS pode ser eficiente na sua resolução. Desta forma, este trabalho apresenta as adaptações realizadas na ALNS para resolver o PPT, assim como os resultados obtidos nesta primeira etapa do trabalho.

3. Adaptive Large Neighborhood Search Aplicada ao PPT

A ALNS inicia sua busca a partir de uma solução inicial viável e opera com diferentes métodos que destroem e reparam a solução corrente. Enquanto um par dessas heurísticas de remoção e inserção é adotado durante toda a busca na LNS, na ALNS várias heurísticas de remoção e inserção são utilizadas na mesma busca. Um peso é inicialmente atribuído a cada heurística com o intuito de se encontrar o par mais eficiente para o problema. Os pesos são ajustados dinamicamente, baseados na eficiência de cada método ao longo da busca.

3.1. Representação da Solução

Para o PPT, uma solução é representada por um conjunto de jornadas, e cada jornada contém a sequência das tarefas a serem realizadas pela tripulação responsável. Esta representação possibilita aplicar os mecanismos de remoção e reinserção inerentes à ALNS.

3.2. Solução Inicial

A ALNS inicia sua busca a partir de uma solução inicial viável. Para o PPT, a solução inicial foi gerada de maneira gulosa. Dado um conjunto de tarefas a ser realizado em um determinado dia de operação, o objetivo é agrupá-las respeitando todas as restrições impostas ao problema sem que haja sobreposição de tarefas. Um sequenciamento válido de tarefas constitui em uma jornada e cada tarefa deve estar alocada a uma e somente uma jornada.

A construção da solução inicia com a ordenação das tarefas pelo horário de início crescente. Em caso de empate, verifica-se o horário de término. A partir da primeira tarefa t_i , contida no conjunto ordenado das tarefas, inicia-se então a alocação das tarefas. A tarefa é incluída na primeira jornada. A próxima tarefa t_j , a ser incluída, é aquela que ainda não foi alocada, que leva ao menor acréscimo na função objetivo e respeita as seguintes restrições: *i*) pertence ao mesmo veículo de operação de t_i ; *ii*) o horário de início de t_j é maior ou igual ao horário de término de t_i ; e *iii*) o ponto de início de t_j é igual ao ponto de término de t_i . A última tarefa alocada passa a ser a nova tarefa t_i , e o processo se repete enquanto o tempo máximo permitido para uma jornada não for atingido. Caso o tempo máximo permitido para uma jornada exceda o tempo normal, esta alocação resulta em horas extras para a jornada. E caso o tempo máximo permitido para uma jornada seja atingido, uma nova jornada é inicializada e o processo se repete até que todas as tarefas tenham sido alocadas.

3.3. Função Objetivo

Considere $j_i, i = 1, \dots, n$, as jornadas de uma solução s , a serem realizadas durante um determinado dia. Então, o custo diário de execução das jornadas é calculado pela expressão:

$$Custo(s) = \sum_{i=1}^n CustoFixo + w_1 \times HE(j_i) + w_2 \times DP(j_i) \quad (1)$$

O *CustoFixo* está associado à manutenção de uma dupla no quadro de funcionários da empresa, $HE(j_i)$ é o tempo total (em minutos) de horas extras da jornada j_i e $DP(j_i) = 1$ se j_i for uma jornada do tipo dupla pegada, caso contrário $DP(j_i) = 0$. As constantes w_1 e w_2 são ajustadas para adequar as soluções obtidas aos interesses da empresa.

O custo de uma solução do PPT, dado pela equação (1) foi avaliado considerando os seguintes valores para os parâmetros: $CustoFixo = 10.000$, $w_1 = 4$ e $w_2 = 5.000$, de modo que o método desenvolvido possa ser comparado com outros autores que utilizaram a mesma base de

dados e ponderação dos parâmetros da Função Objetivo (FO). Dessa forma, o custo da tripulação tem um valor elevado para que a redução no número de duplas seja o principal critério de otimização, seguindo-se pela redução das duplas pegadas e finalmente, das horas extras. O valor da FO para a solução adotada pela empresa foi calculado aplicando os mesmos coeficientes definidos para o modelo.

3.4. Heurísticas Destrutivas

Foram implementadas quatro heurísticas destrutivas, sendo três delas propostas por Ropke e Pisinger (2006a) e Shaw (1997), e uma delas foi proposta neste trabalho. No PPT, os elementos a serem removidos e reinseridos na solução são as tarefas executadas pelas tripulações. Assim, uma heurística destrutiva deve remover uma quantidade pré-definida q de tarefas de uma dada solução.

3.4.1. Remoção Randômica

A Remoção Randômica é caracterizada pela remoção de q tarefas escolhidas aleatoriamente. Inicialmente é sorteada uma jornada e dela é removida uma tarefa, também escolhida aleatoriamente. O procedimento é encerrado quando q tarefas tiverem sido removidas da solução.

3.4.2. Remoção de Pior Posição

Na Remoção de Pior Posição, a cada iteração é removida a tarefa que produz a maior diferença entre o valor da FO com a tarefa e o valor da FO sem a tarefa na solução. O custo de uma tarefa i em uma solução s é dado por $custo_pior(i, s) = f(s) - f_{-i}(s)$, onde $f_{-i}(s)$ é o custo da solução s sem a tarefa i . Esta remoção tem como objetivo realocar tarefas que causam grande impacto na FO, ou seja, aquelas que maximizam a função $custo_pior(i, s)$.

3.4.3. Remoção Shaw

A Remoção *Shaw* tem como ideia remover tarefas similares, o que permite que sejam reinseridas facilmente, podendo gerar soluções melhores. A semelhança de duas tarefas é computada através de uma medida de *similaridade*. Para o PPT, duas tarefas são similares se elas iniciarem e finalizarem no mesmo local, pois como as tarefas se “encaixam” pelos pontos iniciais e finais de operação, é impossível realizar a troca de tarefas que com locais distintos de início e término. A similaridade entre duas tarefas i e j para o PPT é dada por:

$$similaridade(i, j) = \alpha(|st_i - st_j| + |et_i - et_j|) + \beta(|(et_i - st_i) - (et_j - st_j)|) \quad (2)$$

Em (2), st_i corresponde à hora inicial e et_i à hora final da tarefa i . A primeira parcela de (2) considera a proximidade do horário do início e do final das tarefas, enquanto a segunda parcela considera a proximidade do tempo de duração das duas tarefas. Desta forma, quanto menor o valor de (2), maior a similaridade entre o par de tarefas i e j . Os pesos α e β controlam a proporção das características das tarefas na medida.

O procedimento inicia escolhendo aleatoriamente uma tarefa a ser removida. As demais $q-1$ tarefas são escolhidas de acordo com a similaridade em relação a uma tarefa removida anteriormente. Determinadas tarefas podem não alcançar $q-1$ tarefas similares, quando não existirem $q-1$ tarefas que iniciam e terminam no mesmo local que a tarefa considerada. Neste caso, a quantidade q de tarefas removida passa a ser a quantidade possível de remoção.

3.4.4. Remoção Média

Visto que um dos objetivos ao resolver o PPT é minimizar o número de horas extras, é interessante que as jornadas tenham a duração aproximadamente igual. Assim, é proposto neste trabalho o método de Remoção Média. Nesta remoção, são retiradas tarefas das jornadas cujos tempos de duração excedem a duração média. São permitidas remoções da primeira tarefa ou da última tarefa da jornada. A escolha é aleatória a cada remoção. Não é interessante remover uma

tarefa contida no interior da jornada, visto que ela não diminui a duração da jornada e pode aumentar sua ociosidade interna.

3.5. Heurísticas Construtivas

Para reparar as soluções parcialmente destruídas, duas heurísticas construtivas foram empregadas no PPT. As heurísticas foram adaptadas daquelas propostas por Ropke e Pisinger (2006a) e Potvin e Rousseau (1993). No contexto do PPT, *inserir uma tarefa em uma posição* é o mesmo que inserir uma tarefa em uma jornada, visto que não é possível aloca-la em mais de uma posição na jornada, devido a restrição temporal, que coloca as tarefas em ordem cronológica. Quando a inserção de novas tarefas nas jornadas existentes não for possível, é criada uma nova jornada a partir da tarefa a ser inserida.

Durante a reconstrução das jornadas, permite-se que sejam realizadas trocas de veículos. Isso possibilita ampliar o espaço de busca. Uma quantidade máxima de trocas de veículos em uma jornada é definida a *priori*. Para o caso estudado, permite-se, no máximo, 2 trocas de veículo por jornada.

3.5.1. Inserção Gulosa

O método de Inserção Gulosa insere uma tarefa em sua posição mais barata na solução. Para cada tarefa i , é verificada, dentre as possibilidades de inserção, aquela que resulta no menor custo. Esta posição é denominada *posição de custo mínimo*, c_i . Por fim, é escolhida a tarefa i dentre as tarefas não alocadas com o menor valor de c_i e é realizada a inserção em sua posição de custo mínimo. O processo continua até que todas as tarefas tenham sido inseridas.

3.5.2. Inserção de Arrependimento

As heurísticas de arrependimento se baseiam na heurística Gulosa e integram uma espécie de avaliação do tipo “olhar à frente” ao selecionar uma tarefa para inserção. Considere $f_{\Delta}(i, k)$ como a mudança no valor da FO causada pela inserção da tarefa i em sua k -ésima posição válida mais barata, então o valor da inserção 2-arrependimento é dado por $c_i^* = f_{\Delta}(i, 2) - f_{\Delta}(i, 1)$. Em cada iteração, é inserida a tarefa i que maximize c_i^* . Desta forma, é escolhida para a inserção a tarefa que produzir o maior impacto na FO caso ela não seja alocada na sua melhor posição e tiver que ir para a sua segunda melhor posição.

3.6. Escolha das Heurísticas de Remoção e Inserção

Definidas as heurísticas que destroem e reconstróem uma solução, o próximo passo consiste em escolher um par de heurísticas para modificar a solução corrente. Um único par de heurísticas poderia ser utilizado durante toda a busca. No entanto, é interessante que todas as heurísticas sejam utilizadas, visto que cada uma delas pode ser mais adequada em um determinado momento da busca. Acredita-se que esta alternância resulta numa aplicação mais robusta (Ropke e Pisinger, 2006a).

A escolha das heurísticas que formam o par é independente, e a seleção de cada uma, durante a busca, é feita pelo princípio de seleção da roleta. A cada heurística é atribuído um peso, e o sorteio é dado em função de sua probabilidade de escolha. Considerando um conjunto de k heurísticas, com pesos w_i , $i \in \{1, 2, \dots, k\}$, uma heurística j é selecionada com probabilidade:

$$\frac{w_j}{\sum_{i=1}^k w_i} \quad (3)$$

Suponha um gráfico de pizza onde a cada heurística é atribuído um espaço proporcional ao seu peso. Uma roleta exterior é colocada sobre o gráfico e uma heurística é selecionada. Quanto maior o peso da heurística, maior seu espaço no gráfico e, portanto, maior sua chance de ser escolhida. O peso está diretamente relacionado com a eficiência da heurística em melhorar a solução.

3.7. Ajuste Adaptativo dos Pesos

Os pesos das heurísticas representam a eficiência de cada uma durante o processo de busca provido pela heurística ALNS. Os pesos são ajustados automaticamente usando estatísticas das iterações anteriores. Uma *pontuação* é mantida para cada heurística, que mede seu desempenho em um determinado número de iterações. Quanto maior a pontuação da heurística, maior sua eficiência. A busca é dividida em *segmentos*, que por sua vez, são divididos em iterações ou tempo de processamento. Assim, definem-se os critérios de parada do algoritmo. No início de cada segmento, a pontuação de cada heurística é definida como zero, e é aumentada durante a busca em três situações diferentes. No primeiro caso, aumenta-se σ_1 na pontuação quando a heurística é capaz de encontrar uma nova solução global s^* . Aumenta-se em σ_2 na pontuação quando a heurística é capaz de encontrar uma solução s' não encontrada anteriormente e que seja melhor do que a solução atual s . No último caso, aumenta-se em σ_3 na pontuação quando a heurística é capaz de encontrar uma solução s' não encontrada anteriormente e que seja pior que a solução atual s , porém ainda aceita pelo critério de aceitação, a ser descrito. Uma vez que a situação caracterizada pelo acréscimo de σ_1 é mais atraente que a situação caracterizada pelo acréscimo de σ_2 que, por sua vez, é mais atraente que a situação caracterizada pelo acréscimo de σ_3 , é compreensível que $\sigma_1 > \sigma_2 > \sigma_3$. Apesar de serem preferíveis as heurísticas que melhoram a solução, é também interessante utilizar heurísticas que possam diversificar a busca, como no último caso, o que encoraja as heurísticas a explorarem novas regiões do espaço de busca.

Em cada iteração da ALNS são aplicadas uma heurística de remoção e uma heurística de inserção. Como ambas são responsáveis pela geração de uma nova solução, a pontuação das mesmas é atualizada na mesma quantidade, pois não é possível garantir qual das duas foi o fator de sucesso da operação. No fim de cada segmento, os pesos das heurísticas são atualizados levando em consideração a pontuação obtida no segmento corrente.

Assim como as pontuações são inicializadas com o valor zero para cada heurística, os respectivos pesos são inicializados com valor 1. No final de cada segmento, novos pesos são calculados utilizando as pontuações salvas. As pontuações, por sua vez, são atualizadas no fim de cada iteração, e são zeradas ao iniciar um novo segmento. Sendo $w_{i,j}$ o peso da heurística i utilizado no segmento j , o peso da heurística i a ser utilizado no segmento $j+1$ é calculado como:

$$w_{i,j+1} = w_{i,j}(1 - r) + r \frac{\pi_i}{\theta_i} \quad (4)$$

Em (4), π_i é a pontuação obtida pela heurística i durante seu último segmento, e θ_i é a quantidade de vezes que a heurística i foi utilizada durante seu último segmento. O *fator de reação* r controla o quanto os próximos pesos serão influenciados pelos pesos do segmento anterior. Se $r = 0$, os pesos do segmento anterior são repetidos no próximo segmento. Se $r = 1$, os pesos do segmento anterior são totalmente descartados. Um valor entre 0 e 1 para o fator de reação resulta em novos pesos que consideram o desempenho das heurísticas no segmento anterior.

3.8. Critério de Aceitação

Assim como em Ropke e Pisinger (2006a), foi utilizado o critério de aceitação da metaheurística *Simulated Annealing*. Dessa forma, uma solução s' será aceita se for melhor do que s . Se s' for pior do que s , s' será aceita com probabilidade $e^{-(f(s')-f(s))/T}$, onde $T > 0$ representa a temperatura atual. A temperatura T começa em t_0 e diminui a cada iteração através da expressão $T = T * c$, onde $0 < c < 1$ é a *taxa de resfriamento*. Como uma boa escolha de t_0 deve estar relacionada ao problema, foi utilizado um método para a obtenção da temperatura inicial que leva em consideração as características de cada instância, descrito em Talbi (2009). Neste trabalho a temperatura inicial parte de um valor baixo e é aumentada a uma taxa de 10% até que pelo menos 70% das soluções vizinhas sejam aceitas.

3.9. Método para Minimizar o Número de Jornadas

O método utilizado para minimizar o número de jornadas no PPT é baseado no método proposto por Ropke e Pisinger (2006a). Ele parte de uma solução inicial viável s_i , com um determinado número de jornadas. Para cada jornada, todas as suas tarefas são removidas da solução e a heurística de Inserção Gulosa tenta reinseri-las nas demais jornadas existentes, criando uma nova solução s' . Se, ao final da reinserção, o número de jornadas de s' for menor do que em s_i , s' é aceita como s_i e a busca se reinicia a partir da primeira jornada da nova solução. Caso contrário, passa-se para a jornada seguinte. O procedimento é encerrado quando todas as jornadas foram consideradas no processo de destruição e reconstrução da solução, e o número de jornadas não foi reduzido.

O método de minimização de jornadas é aplicado na solução inicial, antes de realizar a busca ALNS. Assim, espera-se que a ALNS possa reduzir os custos variáveis da solução, visto que o número de jornadas já foi minimizado.

4. Experimentos Computacionais

Para testar a ALNS foram utilizados dados reais de uma empresa de transporte público que opera em Belo Horizonte. Os experimentos foram realizados em um computador com processador Intel(R) Core(TM) i7-4770, 8 GB de RAM, Windows 10, e o algoritmo foi desenvolvido em C++.

Os dados fornecidos pela empresa são: a quantidade de tarefas a ser executada em cada dia útil, no sábado e no domingo da semana estudada, com os seus respectivos horários de início e término, ponto inicial e final, e veículo de operação. Assim, foi possível calcular o tempo de duração, a quantidade de horas extras, a caracterização como jornada de dupla pegada e seus veículos de operação. Na Tabela 1 está contida, para cada tipo de dia, a quantidade de tarefas que devem ser executadas, dados de entrada que dão a dimensão de cada problema.

Tabela 1. Quantidade de tarefas a serem executadas nos dias úteis, sábados e domingos

	Seg	Ter	Qua	Qui	Sex	Sab	Dom
Total de Tarefas	705	674	814	872	787	644	345

4.1. Resolução do Problema

A heurística ALNS conta com uma série de parâmetros. Primeiro, deve-se definir a quantidade q de tarefas a serem removidas e reinseridas na solução. O critério de aceitação de uma solução é controlado por uma temperatura inicial t_0 , resfriada por uma taxa c . A heurística Remoção Shaw é controlada pelos parâmetros α , β e p_{shaw} , enquanto a heurística Remoção de Pior Posição é controlada apenas pelo parâmetro p_{pior} . Também devem ser definidos os parâmetros σ_1 , σ_2 , σ_3 e r , utilizados no ajuste adaptativo dos pesos. O critério de parada da ALNS é definido pela quantidade de segmentos e seu tamanho, dado pelo tempo de processamento.

Devido à quantidade e variedade de parâmetros, em um primeiro momento, alguns dos parâmetros citados foram definidos de acordo com Ropke e Pisinger (2006a). Os valores utilizados são representados pelo vetor $(q, c, \alpha, \beta, p_{shaw}, p_{pior}, \sigma_1, \sigma_2, \sigma_3, r) = (35, 0.99975, 1, 1, 6, 6, 20, 10, 5, 0.25)$. A partir deste vetor de parâmetros da ALNS, foram realizados testes computacionais com a finalidade de encontrar a melhor temperatura inicial t_0 para o problema.

A temperatura inicial foi o primeiro parâmetro a ser calibrado, uma vez que testes preliminares mostraram que a alteração de seu valor tem um grande impacto na solução obtida ao longo da busca. Um valor elevado para t_0 , conforme retornado pelo método descrito na seção 3.8, possibilita a aceitação de um grande número de soluções de piora, que por sua vez, geraram soluções ainda piores, e que eram aceitas como solução corrente. No entanto, ao invés de definir um baixo valor para t_0 e ignorar o uso do método de inicialização de t_0 , um novo valor foi redefinido baseado em seu valor original. Dessa forma, o valor de t_0 foi reduzido em 50% de seu valor a cada etapa dos

testes. Foram executados testes com o valor de t_0 em 100%, 50%, 25%, 12,50% e 6,25% do valor original de t_0 . Como as soluções melhoravam sistematicamente, ou seja, quanto menor a porcentagem de t_0 , melhores eram as soluções obtidas, foram considerados novos valores. Os novos valores foram 3,125%, 1,5625%, 0,78125% e 0,39125%. Nesta fase preliminar de testes, o problema da segunda-feira foi utilizado para representar os dias úteis, por ser um dia típico da semana. Assim, três problemas foram considerados: dia útil (segunda-feira), sábado e domingo.

Em cada teste foram realizados seis segmentos, cada segmento limitado a cinco minutos de execução. A partir de cinco testes computacionais, foram extraídas a média aritmética e o desvio padrão amostral do valor da FO. A Figura 1 mostra os valores médios da FO (no eixo ordenadas) obtidos para cada porcentagem de t_0 (eixo das abscissas) para os dias úteis, sábado e domingo, respectivamente.

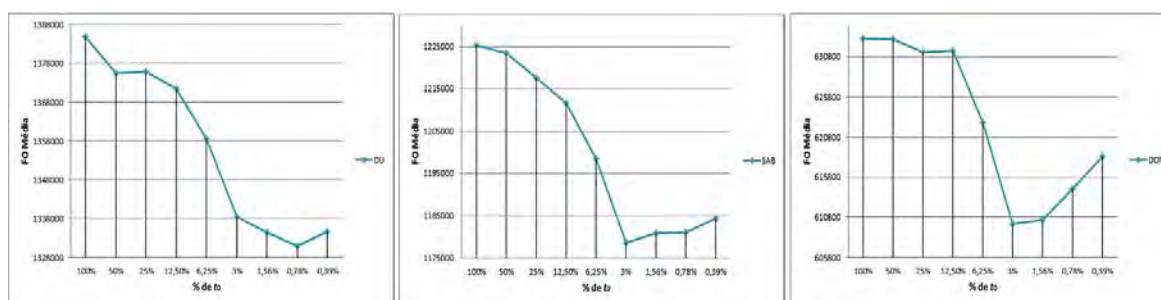


Figura 1. FO Média obtida para as porcentagens de t_0 no dia útil (DU), sábado (SAB) e domingo (DOM)

Analisando a Figura 1, nota-se a grande influência da porcentagem de t_0 ao longo da busca provida pela ALNS. Valores muito altos para a temperatura inicial levam a um ponto de partida extremamente ineficiente para a heurística. No entanto, valores muito baixos, observados na extremidade direita do gráfico, anulam a possibilidade de escape de ótimos locais da heurística, e as soluções tendem a piorar. Quanto menor o valor de t_0 , maior a resistência do método em aceitar uma solução vizinha de piora. Assim, de acordo com os resultados e análises apresentados, foram escolhidos os pesos de 0,78%, 3,125% e 3,125% do parâmetro temperatura inicial t_0 para os dias úteis, sábado e domingo, respectivamente. A partir destes valores, foram realizados novos testes de eficiência considerando todos os problemas, porém com um tempo de processamento maior.

Com a realização dos testes de eficiência, considerando o parâmetro t_0 calibrado, foram obtidos os resultados apresentados na Tabela 2. Para cada dia da semana, a Tabela 2 mostra o valor da FO Média, a porcentagem do desvio padrão, (DP%) e o valor da FO da melhor solução (Melhor FO) obtida dentre as 10 execuções de cada dia.

Tabela 2. FO média, DP (%) e o valor da FO da melhor solução obtida para os dias úteis, sábado e domingo com as respectivas porcentagens de t_0 selecionadas

Características	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
FO Média	1.329.9924	1.242.618	1.485.368	1.605.606	1.521.742	1.175.903	609.934
DP (%)	0,706	0,560	0,702	0,628	0,569	0,872	0,619
Melhor FO	1.315.880	1.231.528	1.473.800	1.584.964	1.510.448	1.157.464	605.040

Na Tabela 3, é apresentado o melhor resultado obtido para cada dia juntamente com a solução realizada pela empresa e pelo modelo proposto por Silva e Silva (2015). Nesta tabela, para cada modelo proposto, é apresentado o valor da função objetivo (FO), o número de tripulações (Jorn.), a quantidade de jornadas de dupla pagada (DP) e o total de horas extras (HE).

Como pode ser notado na Tabela 3, o modelo proposto superou a solução adotada pela empresa em todos os dias analisados. Comparando os resultados com o modelo proposto por Silva e

Silva (2015), a heurística ALNS obteve a mesma quantidade de jornadas para os domingos e a sexta-feira, utilizando uma jornada a menos no sábado. Para este dia, foi necessário realizar três duplas pegadas a mais e um número maior de horas extras. Comparando a qualidade das soluções da ALNS com as de Silva e Silva (2015) pode-se verificar que a ALNS conta com um número maior de horas extras em todos os problemas e de jornadas na maioria dos problemas. Estas características podem ser melhoradas com uma calibração adequada dos parâmetros da heurística estudada.

Tabela 3. Comparação dos resultados obtidos pelo ALNS com a solução obtida por Silva e Silva (2015) e com a solução adotada pela empresa, para cada dia considerado

	Dados	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
ALNS	FO	1.315.880	1.231.528	1.473.800	1.584.964	1.510.448	1.157.464	605.040
	Jorn.	123	119	140	152	142	109	57
	DP	13	4	10	8	13	10	5
	HE	87:00	89:42	99:10	104:01	106:02	72:46	41:50
Silva e Silva (2015)	FO	1.276.280	1.216.340	1.455.988	1.581.920	1.478.832	1.146.440	601.560
	Jorn.	120	116	139	150	142	110	57
	DP	12	8	10	13	8	7	5
	HE	67:50	68:05	66:37	70:30	78:28	47:40	27:20
Empresa	FO	1.390.804	1.335.620	1.540.460	1.668.856	1.580.964	1.253.100	686.468
	Jorn.	134	130	149	162	155	124	68
	DP	6	3	5	4	1	0	0
	HE	86:41	85:55	106:05	120:14	108:11	54:35	26:57

Na Tabela 4 são apresentadas as diferenças percentuais das FOs das soluções obtidas pela heurística ALNS e daquelas obtidas por Silva e Silva (2015). Tais porcentagens foram obtidas da seguinte forma: $(FO \text{ da ALNS} - FO \text{ de Silva e Silva (2015)}) / FO \text{ de Silva e Silva (2015)}$.

Tabela 4. Diferença (%) dos resultados, em relação a FO, obtidos pela ALNS e por Silva e Silva (2015)

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
Diferença (%)	3,103%	1,249%	1,223%	0,192%	2,138%	0,962%	0,578%

Analisando os valores da Tabela 4 pode-se observar que os resultados produzidos pela ALNS estão muito próximos dos melhores resultados obtidos anteriormente, com as diferenças das FOs variando entre 0,578% e 3,103%. Esta constatação leva a crer que ao final da calibração de todos os parâmetros da ALNS, é possível que sejam encontrados resultados melhores do que aqueles conhecidos para as instâncias testadas.

5. Conclusões

Este trabalho apresenta uma implementação da heurística *Adaptive Large Neighborhood Search* para resolver o Problema de Programação de Tripulações do Transporte Público. Nesta implementação, as heurísticas de remoção e inserção proposta por Ropke e Pisinger (2007) foram adaptadas para o PPT, e uma nova heurística de remoção foi proposta para o problema em estudo. A implementação está na fase inicial da definição dos seus parâmetros, sendo que apenas a temperatura inicial do critério de aceitação foi calibrada. Para os demais parâmetros, foram utilizados valores obtidos na literatura. Ainda assim, os resultados preliminares se mostraram competitivos com resultados apresentados na literatura. Com a calibração de todos os parâmetros do ALNS, especificamente para o problema, espera-se obter soluções melhores do que as melhores conhecidas para os problemas.

Uma característica adicional a ser considerada como trabalho futuro consiste em limitar a quantidade de jornadas de dupla pegada na solução, ao invés de considerá-las na FO. O peso para jornadas de dupla pegada, considerado na FO, seria desconsiderado e uma penalização deve ser acrescentada ao custo da solução se a quantidade máxima de duplas pegadas for ultrapassada.

Agradecimentos

Os autores agradecem à CAPES, ao CNPq, à FAPEMIG e à UFOP pelo apoio recebido no desenvolvimento deste trabalho.

Referências

- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. e Vance, P. H.** (1998), Column Generation for Solving Huge Integer Programs. *Transportation Science*, 23, 1–33.
- Bodin, L., Golden, B., Assad, A. e Ball, M.** (1983), Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research*, v 10, p. 63-211.
- Chen, S. e Shen, Y.** (2013), An improved column generation algorithm for crew scheduling problems. *Journal of Information and Computational Science*, v. 10, n. 1, p. 175–183.
- Desrochers, M. e Soumis, F.** (1989), A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science*, INFORMS, v. 23, n. 1, p. 1–13.
- Fischetti, M., Martello, S. e Toth, P.** (1987) The fixed job schedule problem with spread-time constraints. *Operations Research*, 35(6), p. 849–858.
- Fores, S., Proll, L. e Wren, A.** (1999) An improved ILP system for driver scheduling. N. H. M. Wilson (Ed), *Computer-Aided Transit Scheduling*, p. 43–61. Springer.
- Friberg, C. e Haase, K.** (1999) An exact branch and cut algorithm for the vehicle and crew scheduling problem. *Computer-Aided Transit Scheduling*, p. 63–80. Springer.
- Lourenço, H. R., Paixão, J. P. e Portugal, R.** (2001), Multiobjective metaheuristics for the bus driver scheduling problem. *Transportation science*, INFORMS, v. 35, n. 3, p. 331–343.
- Pisinger, D. e Ropke, S.** (2007), A general heuristic for vehicle routing problems. *Computers & operations research*, Elsevier, v. 34, n. 8, p. 2403–2435.
- Potvin, J.-Y. e Rousseau, J.-M.** (1993), A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, Elsevier, v. 66, n. 3, p. 331–340.
- Ropke, S. e Pisinger, D.** (2006a), An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, Informs, v. 40, n. 4, p. 455–472.
- Ropke, S. e Pisinger, D.** (2006b), An unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, Elsevier, v. 171, n. 3, p. 750–775.
- Shaw, P.** (1997), A new local search algorithm providing high quality solutions to vehicle routing problems. APES Group, Department of Computer Science, University of Strathclyde, Glasgow, Scotland, UK, Citeseer.
- Smith, B. M. e Wren, A.** (1988) A Bus Crew Scheduling System Using a Set Covering Formulation. *Transportation Research*, v. 22A, p. 97–108.
- Silva, G. P., Souza, M. J. F. e von Atzingen, J.** (2006), Métodos exatos para resolver o problema de programação da tripulação. *Transportes*, v. 14, n. 1, p. 25-32.
- Silva, T. A. e Silva, G. P.** (2015), O uso da metaheurística *Guided Local Search* para resolver o Problema de Escala de Motoristas de Ônibus Urbano. *Transportes*, v. 23, n. 2, p. 1-12.
- Song, C., Guan, W., Ma, J. e Liu, T.** (2015), Improved genetic algorithm with gene recombination for bus crew-scheduling problem. *Mathematical Problems in Engineering*, 2015, 1. Hindawi Publishing Corporation.
- Souza, D. S.** (2014), Uma Abordagem Híbrida para Resolver o Problema da Escala de Motoristas de Ônibus Urbano. Dissertação de Mestrado – Universidade Federal de Ouro Preto.
- Talbi, E. G.** (2009) *Metaheuristics: from design to implementation*, John Wiley & Sons.
- Wren, A.** (2004) Scheduling vehicles and their drivers – Forty years’ experience. Technical Report, School of Computing Studies, Leeds University, England.