

O uso da metaheurística *Guided Local Search* para resolver o problema de escala de motoristas de ônibus urbano

Tiago Alves Silva¹, Gustavo Peixoto Silva²

Resumo: Neste artigo é aplicada a metaheurística *Guided Local Search* (GLS) para resolver o Problema de Programação de Tripulações de Ônibus Urbano (PPT). O PPT consiste em encontrar um conjunto de jornadas a serem designadas aos motoristas que realizarão a operação diária com o menor custo. A GLS tem como princípio penalizar características indesejáveis presentes na solução corrente, com o objetivo de escapar de soluções ótimas locais. Como heurística de busca local, foi utilizada a heurística *Variable Neighborhood Descent*, que explora diferentes estruturas de vizinhança para encontrar um mínimo local. De acordo com pesquisa realizada pelos autores, esta abordagem é inédita para resolver o PPT. A implementação proposta foi testada com dados de problemas reais de uma empresa que opera em uma região metropolitana de Belo Horizonte. Os resultados obtidos são similares àqueles presentes na literatura, havendo possibilidades de melhorias, visto que a GLS pode ser explorada em diferentes aspectos.

Palavras-chave: Guided Local Search, Programação de Tripulações do Sistema de Transporte, Metaheurística.

Abstract: In this paper the metaheuristic *Guided Local Search* (GLS) is applied to solve the Crew Scheduling Problem (CSP) for public mass transport system. The CSP consists on finding a set of duties to be assigned to drivers in order that the daily service requirement be met at minimum cost. The GLS metaheuristic follows the basic principle of penalizing undesired characteristics that are present in the current solution and to append this penalization in the objective function with the aim of guiding the local search away from local minimum. As local search heuristic, it was employed *Variable Neighborhood Descent* technique, which explores different neighborhood structures to find a local optimum. According to a research conducted by the authors, this is a novel approach to solve the CSP. The proposed implementation was tested with data from real problems of a bus company operating in a metropolitan region of Belo Horizonte. The results are comparable with those reported in the literature, being subject to improvement, once the GLS can be exploited in different ways.

Keywords: Guided Local Search, Crew Scheduling Problem in mass transit, Metaheuristic.

1. INTRODUÇÃO

Em muitas cidades o ônibus é o principal, senão o único meio de transporte público de passageiros. Portanto, é necessário que as empresas do ramo tenham suas atividades muito bem planejadas para atender à demanda dos usuários do sistema de maneira econômica. Para tanto, devem ser utilizadas ferramentas computacionais que auxiliam o processo de decisão. Neste contexto é que se inserem os métodos de otimização, os quais podem levar à redução dos custos destas empresas. Entre os custos de maiores pesos se destacam os custos operacionais com os veículos e os seus motoristas e cobradores (Bouzada, 2003)

Geralmente, o planejamento de um sistema de transporte público é decomposto em etapas a fim de tornar o problema tratável do ponto de vista computacional. A etapa inicial é a elaboração das rotas e dos quadros de horários, de maneira que todas as regiões da cidade sejam atendidas e que o tempo de espera nos pontos e terminais atenda a um dado nível de qualidade estipulado no processo licitatório. A segunda etapa trata do problema da alocação de veículos que realizarão as viagens das tabelas de horários. Nessa etapa é determinada a quantidade necessária de veículos para atender as viagens de cada rota, assim como especificar

as viagens a serem realizadas por cada veículo durante a operação. A terceira etapa compreende o problema da alocação das tripulações, composta pelo motorista e o cobrador, tomando como base as viagens alocadas a cada veículo da frota em operação. Nessa etapa deve ser determinada a quantidade necessária de tripulantes para conduzir os veículos e quais as viagens que cada tripulação realizará. Desta maneira são definidas as jornadas diárias de trabalho que deverão ser executadas pelas tripulações da empresa. A etapa final do planejamento da operação do sistema diz respeito ao problema do rodízio das tripulações. Nesta etapa são geradas as escalas mensais das tripulações com o menor custo possível de maneira a cobrir todas as jornadas diárias, respeitando as regras trabalhistas para o período. Essas etapas geralmente são realizadas na ordem que foram expostas e a solução do problema de uma etapa é usada como entrada da etapa subsequente.

Dentro das etapas mencionadas anteriormente, o foco desse trabalho é o problema de alocação das tripulações, denominado na literatura como *Problema da Programação de Tripulações – PPT (crew scheduling problem)*. Este problema consiste em determinar o número mínimo de tripulações e especificar as suas viagens de tal forma a cobrir todas as viagens da frota em operação com o menor custo possível (Silva e Reis, 2014). Nesta etapa são definidas as jornadas de cada tripulação, portanto, uma tripulação está associada a uma jornada diária de trabalho e vice-versa. Para definir as jornadas diárias, devem ser respeitadas todas as leis trabalhistas e regras operacionais impostas pela empresa. Entre elas destacam-se a carga horária máxima de trabalho, o tempo de descanso entre as jornadas, o número máximo de tripulações que podem trabalhar em um determinado tipo

¹ **Tiago Alves Silva**, Graduado em Ciência da Computação pelo Departamento de Computação da Universidade Federal de Ouro Preto. (tiago.silva1230@gmail.com)

² **Gustavo Peixoto Silva**, Departamento de Computação da Universidade Federal de Ouro Preto. (gustavo@iceb.ufop.br)

Manuscrito recebido em 15/10/2014 e aprovado para publicação em 29/06/2014.
Este artigo é parte de TRANSPORTES v. 23, n. 2, 2015. ISSN: 2237-1346 (online). DOI: 10.14295/transportes.v23i2.856

de jornada, entre outras. As jornadas podem ser de dois tipos: *i) pegada simples*, quando o tempo entre as viagens é inferior a um dado valor, no caso de duas horas, e *ii) dupla pegada*, que são jornadas que apresentam um intervalo maior do que duas horas entre duas viagens. Estas jornadas existem para atender as demandas mais acentuadas que ocorrem especificamente nos horários de pico. Durante uma jornada de trabalho, os tripulantes pode realizar uma *troca de veículos*, que pode ser uma forma de reduzir o tempo ocioso das tripulações. Entretanto, o número de trocas realizado pelas tripulações normalmente é limitado para facilitar o controle sobre o desgaste dos veículos em operação. Uma jornada deve ser definida de tal forma que o tempo de descanso entre o seu final e seu início no dia seguinte seja maior ou igual ao tempo mínimo de descanso estipulado pela legislação, que no caso é de onze horas. Desta forma, uma mesma tripulação pode realizá-la em dias consecutivos sem infringir a regra do descanso em casa. Devido às restrições impostas ao problema, ele se enquadra no conjunto dos problemas NP-Difíceis, para os quais não existem algoritmos polinomiais que obtenham as soluções ótimas (Fischetti *et al.*, 1987). Portanto, é necessária a utilização de métodos heurísticos para resolver, de forma satisfatória, problemas de grande porte.

Neste trabalho, o PPT é resolvido por meio da metaheurística *Guided Local Search* – GLS (Voudouris e Tsang, 1999), que constrói uma sequência de soluções, realizando uma busca local na solução corrente. A GLS se diferencia das demais metaheurísticas por aplicar penalizações às características indesejáveis da solução corrente para escapar de ótimos locais. Assim, pode-se afirmar que a GLS “altera” a curva da função objetivo para que a busca local explore outras regiões deste espaço. Esta é a ideia básica da metaheurística para controlar um método de busca local com a finalidade de lhe dar mais eficiência e robustez (Voudouris e Tsang, 2003). Para aplicar a GLS, deve ser definido um conjunto de atributos ou características associadas à função objetivo. Quando a busca local se encontra em um mínimo local é selecionado o atributo de maior peso na solução ótima local e este atributo é penalizado. A escolha dos elementos a serem penalizados é um ponto importante na GLS e deve levar em conta o peso do elemento na composição do custo. Os elementos com maiores pesos na função objetivo devem ser escolhidos. A metaheurística busca a característica com o maior impacto na função objetivo. Para isso, a cada iteração, o algoritmo analisa todos os atributos e escolhe aquele de maior peso. Este procedimento acelera a saída do mínimo local e permite explorar outras regiões do espaço de soluções.

Para o PPT, tem-se como objetivo minimizar os custos referentes à remuneração das tripulações. Nesse caso, quando as penalidades são aplicadas durante a GLS, a função objetivo aumenta seu valor e um procedimento de busca local é aplicado por um determinado número de iterações. Portanto, quando são aplicadas sucessivas penalizações a uma solução de mínimo local, o valor objetivo desta solução aumenta até que esta deixe de ser um mínimo local. Com isso, a busca local avança no sentido de encontrar outro mínimo local no espaço de soluções viáveis. Após um determinado número de iterações os pesos das penalizações são zerados e inicia-se uma busca local considerando a função objetivo original. O algoritmo continua percorrendo o

espaço de busca até que um determinado critério de parada seja satisfeito.

A implementação desenvolvida neste trabalho utilizou como método de busca local a heurística de refinamento *Variable Neighborhood Descent* (VND), que percorre diferentes estruturas de vizinha para encontrar um mínimo local. A GLS foi testada com um conjunto de problemas reais de uma empresa de transporte público de médio porte e seus resultados foram comparados com resultados da literatura que utilizaram as metaheurísticas *Variable Neighborhood Search* e *Iterated Local Search* (Silva e Reis, 2014).

O trabalho está dividido da seguinte maneira: na Seção 2 é apresentada uma revisão bibliográfica que envolve tanto o problema estudado quanto a metaheurística implementada para resolvê-lo. Na Seção 3 são descritas as características do PPT resolvido e na Seção 4 são apresentadas as linhas gerais da metaheurística GLS, e na Seção 5 é mostrado como ela foi adaptada para resolver o PPT. Na Seção 6 são apresentados os testes de calibração da GLS, os resultados obtidos e a comparação com os resultados conhecidos. Finalmente, na Seção 7 são apresentadas as conclusões do trabalho.

2. REVISÃO DA LITERATURA

A revisão bibliográfica apresenta os principais trabalhos relacionados à resolução do PPT, o quais usam diferentes técnicas de otimização e estratégias para abordar o problema. Como neste trabalho é empregada a metaheurística GLS, para haver uma melhor compreensão desta, são apresentados alguns trabalhos que utilizaram esta metaheurística para solucionar problemas natureza diversa.

2.1. Métodos de resolução do PPT

O PPT é um problema clássico, sobre o qual existem diversos trabalhos que exploram diferentes técnicas de programação linear inteira para resolvê-lo. Entre estas técnicas se destacam o *Branch-and-Bound* (Fores *et al.*, 1999, Smith e Wren, 1988), o *Branch-and-Price* (Desrochers e Soumis, 1989, Barnhart *et al.*, 1998) e o *Branch-and-Cut* (Frigberg e Haase, 1999). Por outro lado, diversas metaheurísticas foram implementadas para resolver problemas de grande porte. Dentre essas pode-se destacar: Algoritmos Genéticos, GRASP, Busca Tabu, *Simulated Annealing*, entre outros (Shen e Kwan, 2001; Silva e Cunha, 2010; Lourenco *et al.*, 2001; Souza *et al.*, 2004). Com o emprego de metaheurísticas é possível obter resultados muito interessantes para o problema, embora a otimalidade das soluções não sejam garantidas.

O trabalho de Ernst *et al.* (2004) apresenta uma série de problemas relacionados à alocação de pessoal em diversas áreas, incluindo motoristas e cobradores de sistemas de transportes. Neste contexto, os autores descrevem problemas de alocação de funcionários em seus diferentes modos de transportes, ou seja, no aeroviário, ferroviário, transporte urbano e no transporte interurbano. No trabalho é apresentada uma revisão sobre as principais pesquisas tanto para o problema de programação da tripulação, como para o problema do rodízio das tripulações. É mencionado também que a complexidade dos problemas depende do tipo de transporte, da categoria da tripulação, dos tipos de frota, das regras e regulamentações, da regularidade das viagens e dos

custos a serem considerados. Segundo Ernst *et al.* (2004), uma das abordagens mais utilizadas para resolver os problemas de programação das tripulações é aquela que o decompõe nas seguintes etapas: geração das jornadas, otimização das jornadas e definição do rodízio da tripulação. Desta forma, a complexidade do problema é reduzida, permitindo sua resolução de forma satisfatória.

Lourenco *et al.* (2001) utilizam duas metaheurísticas para solucionar o PPT multiobjetivo: Busca Tabu e Algoritmo Genético. Isto se faz necessário uma vez que, na prática existem objetivos conflitantes a serem considerados, tais como o custo e a qualidade do serviço prestado. Na abordagem apresentada, para cada jornada é associada uma função de custo real dado pelas horas extras, horas de trabalho noturno e alimentação, e custos artificiais como troca de veículo e número de horas além da média. A qualidade do serviço pode ser medida por funções associadas a: *i*) o número de viagens descobertas, que em pequeno número é favorável à operação; *ii*) sobreposição de jornadas, que em pequeno número garante a realização de dadas viagens; *iii*) número total de jornadas, que deve ser minimizado devido aos custos fixos; *iv*) número total de jornadas do tipo dupla pegada, que deve ser mantido em um dado nível para não comprometer a programação mensal; e *v*) excesso de troca de veículos, que deve ser minimizado para não comprometer o controle do desgaste dos veículos. Na Busca Tabu, os autores consideram duas listas tabu, uma de inserção e outra de remoção de jornadas. A primeira lista contém as jornadas que foram removidas e não podem ser inseridas e a segunda lista contém as jornadas removidas que não podem ser incluídas na solução. A Busca Tabu é realizada para cada função objetivo e posteriormente é aplicada com uma função de soma ponderada. Um processo de intensificação otimizado é proposto onde a Busca Tabu é realizada com várias iterações utilizando somente a lista de inserção. Assim é gerado um conjunto de jornadas candidatas a compor a solução. Posteriormente é utilizada a metaheurística GRASP para selecionar as melhores jornadas. No Algoritmo Genético, a criação de novas populações se baseia na escolha aleatória de uma das funções objetivo para conseguir uma diversificação maior da população. É proposto um método de cruzamento entre duas populações utilizando o GRASP para tentar resolver este cruzamento como um subproblema e encontrar uma população dita perfeita. Os testes realizados com os dados de uma empresa de Portugal mostram que as soluções são superiores às aquelas adotadas na prática.

Um Algoritmo Genético Híbrido para resolver o problema de programação da tripulação foi proposto por Li e Kwan (2003). Esta abordagem foi testada tanto com problemas de ônibus urbano quanto de linhas de trem de empresas inglesas. Uma heurística gulosa é utilizada para construir uma escala, onde as jornadas são selecionadas sequencialmente de um conjunto grande de potenciais jornadas viáveis geradas previamente. As jornadas individuais e a escala devem ser avaliadas como um único processo. A teoria do conjunto *fuzzy* é aplicada sobre tais avaliações. Para jornadas individuais, a sua eficiência é avaliada por critérios identificados a partir do conhecimento prático do problema. O Algoritmo Genético é usado para gerar uma distribuição de pesos quase ótimos entre os critérios, e uma avaliação ponderada de um único valor é computada para cada jornada. A solução construída a partir da distribuição dos pesos gerados é avaliada pela função de aptidão do Algoritmo Genético. Para essa função, são utilizados os objetivos de

minimizar jornadas e minimizar o custo total formulado como uma meta *fuzzy*. Os resultados apresentados no trabalho são competitivos ao serem comparados com o estado da arte, apresentando um ganho em algumas instâncias testadas e reduzindo consideravelmente o tempo de processamento.

Uma utilização da metaheurística Busca Tabu para resolver o PPT da realidade brasileira é apresentada por Marinho *et al.* (2004). Nesta implementação foram utilizadas duas estruturas de vizinhança, uma de realocação e outra de troca de *tarefas* entre jornadas. Uma *tarefa* é uma sequência de viagens, de um mesmo veículo, que devem ser realizadas pela mesma tripulação por não haver tempo hábil para a sua troca. Esta mesma estrutura foi adotada no presente trabalho. A Lista Tabu tem como finalidade evitar que a heurística entre em um ciclo durante a busca do melhor vizinho. A lista é utilizada para armazenar as últimas soluções avaliadas, de modo a não permitir que o método volte para uma solução anteriormente avaliada. Três implementações da Busca Tabu foram abordadas no trabalho: a Busca Tabu com primeiro vizinho de melhora em uma porcentagem da vizinhança; a Busca Tabu com melhor vizinho em uma vizinhança variável e a Busca Tabu com primeiro vizinho de melhora em uma porcentagem da vizinhança com diversificação. Os testes realizados com os três tipos de busca obtiveram resultados relevantes ao serem comparados com outra heurística abordada na literatura e com dados de uma empresa brasileira.

O trabalho desenvolvido por Silva e Cunha (2010) apresenta o emprego da metaheurística GRASP com o uso da técnica de busca em vizinhança de grande porte *Very Large-scale Neighborhood Search* – VLNS (Ahuja *et al.*, 2000) para a solução do problema de programação de tripulações. Este método de busca local, inicialmente gera um grafo de melhoria a partir de uma solução pseudo-gulosa do GRASP. Enquanto houver *ciclos válidos* no grafo o algoritmo encontra um deles e atualiza o grafo gerando uma melhoria na solução corrente. Quando não houver mais ciclos válidos no grafo de melhoria, a solução corrente é um ótimo local e o processo é reiniciado. Um ciclo válido corresponde a um ciclo com custo negativo e representa uma realocação ou troca de tarefas entre jornadas que resulta na redução do custo da solução. A implementação foi testada com dados reais de uma empresa que opera na cidade de Belo Horizonte - MG e estes foram comparados com as soluções adotadas pela empresa, conferindo uma redução significativa nos custos operacionais.

Silva e Reis (2014) resolveram o PPT utilizando a metaheurística *Variable Neighborhood Search* - VNS (Mladenovic e Hansen, 1997). Aliados a ela foram usados dois métodos de busca local, as técnicas *Variable Neighborhood Descent* (VND) e *Very Large-scale Neighborhood Search* (VLNS). Para testar as implementações foi considerado que os tripulantes podem realizar no máximo uma troca de veículos e posteriormente duas trocas de veículos. Também foram considerados cenários onde as duplas pegadas são mais caras, com o intuito de diminuir sua quantidade na solução final. Os resultados obtidos mostram que as soluções apresentadas pela VNS com ambos os métodos de busca local são melhores que a adotada pela empresa. Com o uso da técnica VLNS, foram obtidos resultados melhores quando se considera a quantidade de duplas pegadas dentro dos limites da empresa. No entanto, com a técnica VND houve uma redução na quantidade de jornadas, mas

com um número maior de horas extras e duplas pegadas. Portanto, deve-se analisar qual o objetivo da empresa para então escolher o uso de uma ou outra técnica para gerar as escalas.

2.2. Algumas aplicações da metaheurística GLS

Tsang e Voudouris (1997) utilizaram a metaheurística GLS para resolver o problema da programação de funcionários para executar uma tarefa. Este problema consiste em alocar um número de engenheiros para realizar um conjunto de tarefas, minimizando o custo total de acordo com a função objetivo do problema. As tarefas são descritas por sua localização, duração e o momento do dia em que deve ser feito: de manhã, à tarde, o primeiro trabalho do dia, o último trabalho do dia ou se pode ser realizada a qualquer momento do dia. Cada engenheiro é descrito pela localização da sua base, por seus horários de início e fim de seu expediente, por seu limite de hora extra e por um fator de habilidade que indica quanto tempo o engenheiro leva para completar cada trabalho. A função objetivo busca minimizar o percurso total feito pelos engenheiros para completar as tarefas, o total de horas extras dos engenheiros e a quantidade de tarefas não realizadas. O atributo adotado nesse trabalho foi a ocorrência de tarefas não realizadas. Seu custo é dado pela duração da tarefa mais uma penalização pelo seu não cumprimento. Este se mostra um atributo ideal para a GLS, pois ele pode ou não ocorrer e é desejável que sua quantidade seja a menor possível. Segundo os autores, a eficiência da GLS é demonstrada pelo fato de ela supera todos os métodos aplicados ao problema até então, que incluem *Simulated Annealing*, Algoritmos Genéticos e Programação por Restrições.

De acordo com Balas e Vazacopoulos (1998), vários trabalhos que abordam o problema de *job shop scheduling* utilizam procedimentos de busca local que funcionam com troca de operadores combinados com as metaheurísticas *Simulated Annealing* e Busca Tabu. Neste trabalho, os autores apresentam um novo procedimento de busca em profundidade variável (*variable depth search*), o GLS, baseado em um esquema de troca usando um novo conceito de árvores de vizinhança (*neighborhood trees*). As propriedades estruturais da vizinhança são utilizadas para guiar a busca em direções promissoras. O trabalho mostra que este procedimento produz resultados tão competitivos quanto os procedimentos normalmente empregados na literatura, mesmo utilizado de forma isolada. Entretanto, um procedimento híbrido que embute o GLS em um procedimento do tipo *Shifting Bottleneck* (Adams *et al.*, 1988) e tira vantagem das diferenças entre as duas estruturas de vizinhança, se mostra particularmente eficiente. O procedimento híbrido foi testado extensivamente com a maioria dos problemas disponíveis à época, encontrando a solução ótima de problemas particularmente difíceis à época.

No trabalho de Voudouris e Tsang (1999) a metaheurística GLS é utilizada para resolver o problema do caixeiro viajante. No algoritmo implementado, denominado GLS+FLS+2Opt, a representação das soluções é feita por meio de um vetor contendo a ordem das cidades a serem visitadas. O método de construção da solução inicial considera a geração de uma permutação aleatória das cidades que devem participar da rota. O método de descida se baseia em movimentos *k-Opt*. Esse método seleciona *k* arestas da rota

corrente e as substitui por *k* novas arestas. O custo das soluções do caixeiro viajante é a soma dos pesos das arestas do caminho. Assim, ao penalizar uma aresta, a GLS aumenta seu custo até que ela saia da solução corrente e o procedimento encontre outro caminho. Essa implementação conseguiu resultados melhores do que os encontrados pelo algoritmo Lin-Kernighan (Lin e Kernighan, 1973), que é um algoritmo específico para tratar o problema do caixeiro viajante.

3. O PROBLEMA DE PROGRAMAÇÃO DE TRIPULAÇÕES

O PPT consiste em gerar jornadas de trabalho para os funcionários da empresa responsável pela realização das viagens dos veículos em operação, ou seja, gerar as jornadas para todas as tripulações: motorista e cobrador. Os dados de entrada do PPT são: *i*) a programação detalhada dos veículos que serão conduzidos, cujas viagens são agrupadas em tarefas; *ii*) as regras operacionais da empresa; e *iii*) as leis trabalhistas.

O PPT tem como dados de entrada a programação dos veículos em operação. A programação de cada veículo é composta pela sequência das viagens realizadas pelo veículo naquele dia. As viagens são agrupadas em *tarefas*, que são as menores porções de trabalho. Para que uma tripulação substitua outra, durante a operação de um veículo, deve existir uma *oportunidade de troca* - OT entre as tarefas. Uma OT ocorre quando o intervalo entre as duas tarefas consecutivas do veículo é maior do que certa quantidade de tempo, previamente definida, e o mesmo ocorre em um ponto apropriado, com a presença de um fiscal da empresa para vistoriar o veículo. Assim, todas as viagens realizadas pelos veículos da frota da empresa são transformadas em tarefas que devem ser alocadas às tripulações com o menor custo possível.

A remuneração fixa de uma tripulação se refere a um dado período normal de trabalho, no caso de seis horas e quarenta minutos. Jornadas com duração maior do que este período contém horas extras e jornadas com duração menor, contém ociosidade. Na resolução do PPT pretende-se alocar as tarefas às jornadas reduzindo o total de jornadas utilizadas, o total de horas extras e de horas ociosas contidas na escala.

Uma solução para o problema pode ser vista como um particionamento do conjunto de tarefas entre os tripulantes. Dessa forma, o conjunto de tarefas atribuído a cada tripulação constitui sua *jornada de trabalho*. O conjunto das jornadas da empresa constitui-se na *escala* diária de trabalho para as tripulações da empresa. Desta forma, uma tripulação corresponde a uma jornada e vice-versa.

1. As jornadas de trabalho devem respeitar uma série de acordos coletivos e de leis trabalhistas. As regras de trabalho diário consideradas nesse trabalho, e que são adotadas na prática são:
2. Cada tripulante pode fazer no máximo 2 horas extras por dia;
3. Toda jornada deve permitir que haja um tempo mínimo de descanso em casa de 11 horas;
4. Toda jornada de trabalho é remunerada por 6 horas e 40 minutos; e

5. Nas jornadas do tipo dupla pegada, o intervalo de mais de 2 horas que ocorre entre duas tarefas não é remunerado.

O custo de uma solução do PPT é composto pela combinação linear dos custos fixos e variáveis das suas jornadas. Os custos fixos são compostos pela remuneração paga aos tripulantes e os custos variáveis são representados pelos custos das horas extras trabalhadas, pelo tempo ocioso e pela quantidade de duplas pegadas presente na escala. A quantidade de jornadas do tipo dupla pegada deve ser controlada visto que suas tripulações têm direito a folgareem nos domingos. Desta forma, na prática, as jornadas deste tipo devem ser limitadas. A expressão a seguir representa a função objetivo de uma solução do PPT.

$$C = \sum_{i=1}^{tot_jorn} CF_i + w_1 \times hextra_i + w_2 \times hociosa_i + w_3 \times dupla_pegada_i \quad (1)$$

Onde: *tot_jorn* é o total de jornadas na solução, *CF_i* representa a remuneração fixa da tripulação *i*, *w₁* representa o peso de cada minuto de trabalho extra, *hextra_i* representa a quantidade de horas extras na jornada *i*, expressa em minutos, *w₂* representa o peso de cada minuto de tempo ocioso, *hociosa_i* representa o tempo ocioso, expresso em minutos, *w₃* representa o peso atribuído às duplas pegadas e *dupla_pegada_i* é igual a um se a jornada *i* for do tipo dupla pegada e zero caso contrário.

4. A METAHEURÍSTICA GUIDED LOCAL SEARCH - GLS

A metaheurística GLS parte de uma solução inicial, indicada na Figura 1 pelo ponto $x = a$, realiza uma busca local e chega a um ótimo local, o ponto $x = b$ na Figura 1. Quando o ótimo local é atingido, a GLS seleciona alguns atributos que estão presentes na solução encontrada e os penaliza. As penalizações transformam a função objetivo, cujo gráfico está representado na Figura 1 pelas linhas vermelhas. Novas penalizações fazem com que o ponto $x = b$ deixe de ser um ótimo local (linha vermelha superior) e a busca se mova para outro ponto de ótimo local. Este movimento da busca local é representado pela linha cinza, quando o ponto $x = b$ deixa de ser ótimo local e a busca encontra o ponto $x = c$ como novo ótimo local. Desta forma o algoritmo de busca local consegue escapar de um ponto ótimo local e encontrar um novo ótimo local. Esse processo se repete até que algum critério de parada seja satisfeito.

Com o intuito de acelerar a saída do ótimo local, a GLS busca penalizar os atributos de maior impacto na função objetivo. Quando a busca local chega a um ótimo local, a GLS analisa os atributos presentes naquela solução, seleciona o(s) mais significante(s) e aplica a penalização sobre ele(s). À medida que ocorre a penalização sobre um atributo, seu contador de penalização é incrementado. Isso é feito para que as penalizações não ocorram sempre sobre os mesmos atributos, como uma estratégia de diversificação das soluções.

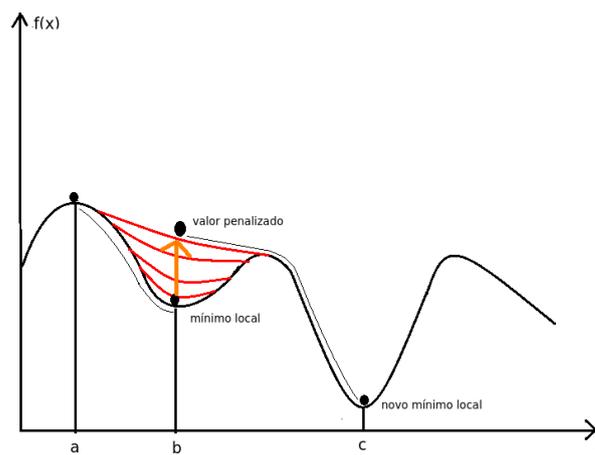


Figura 1. Deformação do espaço de soluções causado pela GLS

Como mencionado anteriormente, a GLS aumenta a função objetivo acrescentando penalidades a ela. Para aplicar GLS, é preciso definir os atributos a serem penalizados no problema. Por exemplo, no problema do caixeiro viajante, um atributo poderia ser "se a cidade B é visitada imediatamente após a cidade A, ou seja, se há a ligação direta entre as duas cidades na rota". A GLS associa um custo e uma penalidade para cada atributo. Os custos podem ser definidos simplesmente tomando os termos e seus coeficientes da função objetivo. Por exemplo, no problema do caixeiro viajante, o custo do atributo descrito anteriormente pode ser a distância entre as cidades A e B. As penalizações são inicializados com 0 e são aumentadas somente quando a busca local atinge um ótimo local. Assim, dada uma função objetivo *g* que associa cada solução candidata *s* a um valor numérico real, a GLS define uma função *h* que é usada na busca local, substituindo a função original *g*, dada por:

$$h(s) = g(s) + \lambda \times \sum p_i \times I_i(s) \quad (2)$$

Onde: *s* é uma solução candidata, λ é a um parâmetro da GLS, *i* percorre todos os atributos a serem penalizados, *p_i* é a penalidade para o atributo *i* (todo *p_i* é inicializado com 0) e *I_i* indica se a solução *s* apresenta o atributo *i* ou não, da seguinte maneira:

$$I_i(s) = \begin{cases} 1 & \text{se } s \text{ apresenta a característica } i; \\ 0 & \text{caso contrário} \end{cases} \quad (3)$$

A metaheurística GLS guia uma ou mais heurísticas de busca local, tendo como estratégia para escapar de ótimos locais aumentar o valor da função de custo, adicionando penalidades para os atributos selecionados. A diferença da GLS está principalmente na maneira como ela seleciona os atributos para penaliz-los. A intenção é penalizar "características desfavoráveis" ou "características de maior importância" quando a busca local atinge um ótimo local. A característica que tem um custo elevado afeta com maior intensidade o custo global. Outro fator que deve ser considerado é o valor atual da penalização do atributo. A utilidade de penalizar o atributo *i*, denominada *util_i*, no âmbito de um ótimo local *s'*, é definida da seguinte maneira:

$$util_i(s') = I_i(s') \times \frac{c_i}{1 + p_i} \quad (4)$$

Onde *c_i* é o custo e *p_i* é a o valor da penalização corrente do atributo *i*. Em outras palavras, se um atributo não fizer parte do ótimo local, indicado por *I_i*, então a

utilidade em penalizá-lo é zero. Quanto mais elevado for o custo c_i do atributo i , maior será a utilidade em penalizá-lo. Por outro lado, quanto maior o número de vezes que o atributo foi penalizado, menor será a utilidade de penalizá-lo novamente. Em uma solução ótima local, os atributos com os maiores valores para suas utilidades serão penalizados. Quando um atributo i é penalizado, seu valor de penalização p_i é sempre aumentado em uma unidade.

Considerando o custo e a penalização corrente para escolher os atributos a serem penalizados, a GLS focaliza o seu esforço de busca em regiões mais promissoras do espaço de busca, ou seja, áreas que contém soluções candidatas com boas características. Isto é, soluções com custos reduzidos. Por outro lado, as penalizações ajudam a prevenir que as heurísticas de busca local dirijam seus esforços a uma região qualquer do espaço.

O parâmetro λ é o único a ser calibrado na metaheurística GLS. Experimentos mostram que um bom valor para ele pode ser encontrado dividindo-se o valor da função objetivo do ótimo local pelo número de atributos presentes na função. Esse valor deve ser multiplicado por uma constante alfa previamente fixada (Voudouris e Tsang, 2003).

A metaheurística GLS não é sensível à solução inicial. Seu maior esforço se concentra em percorrer o espaço de soluções de maneira eficiente. Ela também não necessita de métodos de busca local complexos. A lista contendo os atributos presentes no ótimo local é representada no Algoritmo 1 por M . Geralmente, a partir do valor das variáveis de decisão é possível identificar os atributos a serem penalizados na solução. Quando isso não for possível, pode-se usar uma lista com os atributos presentes na solução que está sendo analisada. Essa lista deve ser atualizada a cada vizinho gerado. Com isso, ao encontrar um ótimo local, o processo de busca dos atributos se torna mais rápido. A utilidade destes atributos pode ser inserida em um vetor ordenado. Assim, ao selecionar os atributos a serem penalizados, basta tomar o(s) primeiro(s) índice(s) deste vetor. A escolha do atributo a ser considerado pela GLS varia de problema para problema. Sua escolha influencia fortemente o resultado final, visto que é com base em seus valores que as penalizações são aplicadas. O pseudocódigo da metaheurística é apresentado a seguir considera a função objetivo original g e a função h com as devidas penalizações, apresentada na expressão (2).

A metaheurística GLS pode ser combinada com Algoritmos Genéticos (AG) gerando o Algoritmo Genético Guiado (AGG). Nessa técnica, a GLS tem a função de prover maior robustez ao AG. Quando o AG não está mais produzindo progresso em suas iterações o GLS entra em cena modificando sua função de *fitness*. Assim a função de *fitness* modificada pode ser usada em novas gerações. A GLS também pode influenciar tanto o processo de mutação quanto o de cruzamento do AG com suas penalizações.

5. APLICAÇÃO DA METAHEURÍSTICA GLS AO PPT

Para aplicar o GLS, o PPT foi modelado como um problema de particionamento das tarefas entre os tripulantes gerando suas jornadas de trabalho. O particionamento é tal que as jornadas geradas devem respeitar todas as restrições do problema. O critério de parada adotado para a GLS foi o

tempo de processamento, no caso trinta minutos. Este valor foi definido após alguns testes computacionais que mostraram que para um tempo de processamento maior não se verificou uma melhoria na FO.

5.1. Solução Inicial

A solução inicial foi gerada pelo método guloso, tendo em vista adicionar à jornada corrente a tarefa ainda não alocada que levar ao menor acréscimo na função objetivo do problema. Isto é feito enquanto a jornada admitir a inclusão de uma nova tarefa. Caso contrário uma nova jornada é inicializada. O processo se repete até que todas as tarefas tenham sido atribuídas a uma jornada.

5.2. Estrutura de Vizinhaça

As melhorias na função objetivo do problema são feitas através da realocação e da troca de tarefas entre as jornadas. Com base nestes dois tipos de movimento é que foi definida a estrutura de vizinhaça do problema. Essa estrutura de vizinhaça define a quantidade de tarefas sequenciais que podem ser realocadas ou trocadas entre duas jornadas. Com base em experimentos realizados, o tamanho da estrutura de vizinhaça adotada foi quatro. Com isso, até quatro tarefas podem ser removidas de uma jornada e inseridas em outra. Esse movimento de tarefas entre jornadas pode ser caracterizado de dois tipos: realocação ou troca. A partir de duas jornadas i e j e retirando-se k tarefas consecutivas de i e as inserindo em j pode ocorrer as seguintes situações:

- as k tarefas podem ser inseridas em j sem a necessidade da retirada de nenhuma tarefa de j . Nesse caso é realizada uma realocação das k tarefas da jornada i para a jornada j ; ou
- as k tarefas podem ser inseridas em j somente se forem retiradas tarefas de j . Se as tarefas retiradas de j puderem ser inseridas em i , então é realizada uma troca. Se isso não for possível, o movimento é descartado.

Os movimentos são aceitos se, e somente se, as jornadas geradas respeitarem todas as restrições do problema.

5.3. Função Objetivo e Atributos

A Função Objetivo (FO) original do problema foi modificada inserindo-se as penalizações da metaheurística GLS. Com relação às penalizações impostas, a solução adotada foi utilizar múltiplas penalizações para os mínimos locais. Para isso os atributos escolhidos foram a quantidade de horas extras, o tempo ocioso e a quantidade de duplas pegadas. Dessa forma, a cada iteração da GLS a função objetivo recebe três penalizações, uma sobre cada atributo que apresentar o valor mais relevante. Essas penalizações não são necessariamente aplicadas à mesma jornada. São selecionadas, dentre todas as jornadas, aquelas que possuem o maior valor de cada atributo. Com isso, a função objetivo do problema fica da seguinte forma:

$$C = \sum_{i=1}^{tot-jorn} CF_i + (w_1 + \lambda \times p_{i1}) \times hextra_i + (w_2 + \lambda \times p_{i2}) \times hociosa_i + (w_3 + \lambda \times p_{i3}) \times dupla_pegada_i \quad (5)$$

Algoritmo *Guided Local Search***Função GLS** ($p, g, h, \lambda, [I_1, \dots, I_M], [c_1, \dots, c_M], M, Zera_Pen$)

```

 $k := 0;$ 
 $s_1 := Solucao\_Inicial();$ 
 $s' := s_1;$ 
para ( $i = 1$  até  $M$ ) faça
     $p_i := 0;$  //zera todas as penalidades
fim_para
enquanto (critério de parada não satisfeito) faça
     $s_{k+1} := BuscaLocal(s_k, h);$  //busca local considerando a função  $h$ 
    se ( $g(s_{k+1}) < g(s')$ ) então //armazena a melhor solução em  $s'$ 
         $s' := s_{k+1};$ 
    fim_se
    para ( $i = 0$  até  $M$ ) faça
         $util_i(s_{k+1}) := I_i(s_{k+1} \times c_i / (1 + p_i));$  //calcula a utilidade do atributo  $i$ 
    fim_para
    para ( $i = 1, \dots, M$ ) faça //penaliza os atributos
         $i = Max(util_i(s_{k+1})),$ 
         $p_i := p_i + 1;$ 
    fim_para
    se ( $k = Zera\_Pen$ ) então //a cada  $Zera\_Pen$  iterações
        para ( $i = 1$  até  $M$ ) faça //zera todas as penalidades
             $p_i := 0;$ 
        fim_para
         $k = 0;$ 
    fim_se
     $k = k + 1;$ 
fim_enquanto
retorna  $s'$  //retorna a melhor solução encontrada
fim_função

```

Algoritmo 1. Pseudocódigo da metaheurística GLS

A função objetivo (5) é uma adaptação da função (1) onde foram inseridas as penalizações referentes à GLS. Em cada atributo i , foi inserido o parâmetro λ que é o coeficiente de penalização, além de p_{i1} , p_{i2} e p_{i3} que representam a quantidades de penalizações que esse atributo obteve. Somando o produto de ambos ao peso originalmente aplicado aos atributos, é obtido um coeficiente aumentado para o atributo na função. Consequentemente, há um aumento no valor da função objetivo. Dessa forma ocorre uma modificação na função objetivo e consequentemente nas soluções de menores valores.

A modificação do coeficiente de um atributo de uma jornada, hora extra por exemplo, tem como finalidade substituir as tarefas que estão gerando tal custo excessivo. Mas para que o processo se complete, é necessário retornar ao peso original com uma dada periodicidade. Assim, faz-se necessária a utilização de técnicas de uma variação da GLS, a *Reset-GLS* (Voudouris, 1997). Essa variação da GLS inicializa novamente todas as penalizações impostas até então, após uma quantidade de iterações do algoritmo. A finalidade deste procedimento é devolver aos atributos seus custos originais para uma próxima busca local, evitando que a busca descarte boas soluções pelo fato delas estarem penalizadas. Essa medida faz com que o algoritmo sempre comece a penalizar a partir da função objetivo original e essas penalizações não influenciem uma futura busca local, que poderá levar a solução corrente a um novo ótimo.

5.4. Métodos de Busca Local

Neste trabalho foram testados, separadamente, os principais métodos de busca local descritos na literatura,

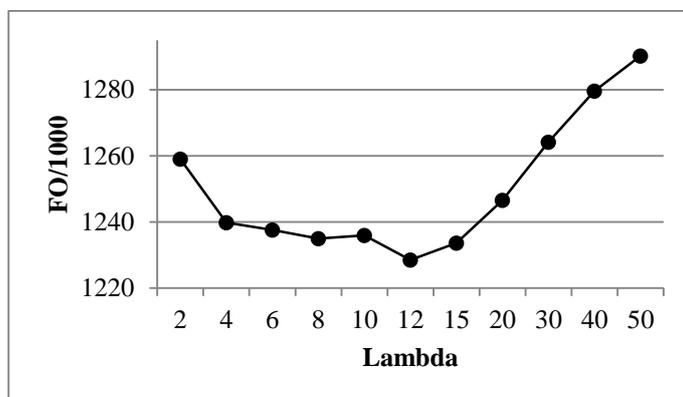
que são: *i*) método de descida com melhor vizinho de melhora (*best improvement*); *ii*) método de descida com o primeiro vizinho de melhora (*first improvement*), *iii*) método de descida com vizinhança variável (*variable neighborhood descent - VND*) combinado com o de melhor vizinho de melhora; e *iv*) método de descida com vizinhança variável combinado com o do primeiro vizinho de melhora. Estes métodos de busca local são largamente utilizados na literatura, foram testados de forma independente, e suas descrições podem ser encontradas em Talbi (2009). O método que produziu os melhores resultados em um teste preliminar foi adotado para os testes de eficiência da GLS.

6. EXPERIMENTOS COMPUTACIONAIS

Os experimentos foram divididos em duas categorias: calibração dos parâmetros e testes de comparação de desempenho. Todos os testes computacionais foram executados em um computador core i7, 3.4 GHz com 8GB de memória RAM. Foram utilizados os seguintes valores para os parâmetros do problema: *i*) custo da remuneração fixa da tripulação CF_i : 10.000, *ii*) custo da dupla pegada: 5.000 e 600 (testados separadamente), *iii*) custo da hora extra expressa em minutos: 4 e *iv*) custo da hora ociosa expressa em minutos: 4. Estes valores foram os mesmos obtidos a partir de testes empíricos realizados no trabalho de Silva e Reis (2014). A escolha também se deve à possibilidade de se realizar a comparação dos resultados obtidos. Foram realizadas 10 execuções do algoritmo para cada valor dos parâmetros considerados. Este número de execuções foi escolhido por não haver uma variação significativa no valor da FO.

Tabela 1. Valor da função objetivo para os diferentes métodos de descida

Valor de k	1	2	3	4
Melhor melhora	1.290.900	1.297.096	1.319.888	1.329.004
Primeira melhora	1.286.372	1.307.012	1.319.468	1.312.448
VND com melhor melhora	1.290.840	1.276.404	1.276.176	1.266.792
VND com primeira melhora	1.286.408	1.266.968	1.268.128	1.257.864

**Gráfico 1.** Variação do valor da função objetivo em função de lambda

Cada execução do algoritmo GLS teve uma duração de 30 minutos.

6.1. Calibração dos Parâmetros da GLS

A partir do processo de calibração foram definidos os valores dos parâmetros utilizados pela metaheurística. Uma característica interessante da metaheurística GLS é o fato dela apresentar poucos parâmetros para serem calibrados. De acordo com a literatura o único parâmetro a ser calibrado é o λ . No entanto, pelo fato do problema abordado apresentar os mesmos atributos em todas as componentes da solução, é necessário estabelecer a periodicidade em que devem ser reinicializadas as penalizações. Além disso, esses testes foram usados para indicar qual método de busca local foi capaz de produzir os melhores resultados.

Os testes de calibração foram realizados com um dia atípico da semana, assim foi escolhido o dia de terça-feira que conta com 58 veículos e 674 tarefas. Os resultados dos testes de calibração apresentam a melhor solução encontrada após a realização de 10 execuções para cada parâmetro analisado. O primeiro teste de calibração foi realizado para avaliar a eficiência dos métodos de busca implementados. Os resultados obtidos podem ser vistos na Tabela 1.

Os resultados mostram que as buscas locais utilizando o VND são mais eficientes do que as buscas usando somente as técnicas de melhor melhora e primeira melhora. Isso ocorre porque o VND tem a liberdade de realizar trocas com vizinhanças de tamanho menor do que o limite a ele imposto. Ou seja, quando se passa um tamanho de vizinhança 4 para o VND, por exemplo, ele realiza trocas de vizinhança de tamanhos 1, 2, 3 e 4. O mesmo não ocorre quando se passa um tamanho de vizinhança igual a 4 para os métodos primeira melhora e melhor melhora. Estes farão a busca apenas nas vizinhanças associadas à realocação/troca de 4 tarefas subsequentes entre jornadas. Com base nesses resultados, o método de busca local adotado para os testes a seguir foi o VND com a busca primeira de melhora.

Nesses testes também foi analisado o tamanho ideal para a estrutura de vizinhança, dada pelo valor de k . Segundo os resultados da Tabela 1, o melhor tamanho para a

vizinhança foi para k igual a 4. Esse valor de k representa a quantidade de tarefas consecutivas que pode ser movimentada entre as jornadas. Como este valor já se encontra no limite de tarefas realizadas em uma jornada, não foi possível testar valores maiores. Desta forma, foi adotado o valor k igual a 4.

O teste de calibração apresentado no Gráfico 1 foi realizado para encontrar o melhor valor para o parâmetro λ da metaheurística GLS. Os resultados mostram os valores médios obtidos para a FO após 10 execuções para cada valor de λ .

O valor da FO atinge os menores valores para $\lambda = 12$. No Gráfico 1 pode ser observado que a função objetivo inicia com um valor alto, sendo reduzido à medida que aumenta o valor de λ . O menor valor da função objetivo foi observado para λ igual a 12, valor adotado no trabalho.

O último teste de calibração foi realizado para encontrar o melhor valor para a quantidade de iterações que a GLS realiza até que as penalizações fossem reinicializadas novamente. Foram testados intervalos de 10 em 10 iterações começando em 10 e finalizando em 80 iterações entre as reinicializações. O Gráfico 2 apresenta o valor médio da FO de 10 execuções para cada valor.

Pelos dados obtidos, é possível verificar que o melhor intervalo foi de 30 iterações entre as reinicializações das penalidades. Portanto, este valor adotado no trabalho. Durante os testes computacionais, foi verificado que a cada execução de 30 minutos são realizadas aproximadamente 300 iterações do algoritmo. Portanto, as penalizações foram reinicializadas por volta de 10 vezes em cada execução. Desta maneira, foram definidos os seguintes parâmetros utilizados na metaheurística:

1. o método de busca local VND combinado com o procedimento primeiro de melhora;
2. tamanho da vizinhança k igual a 4;
3. valor de λ igual a 12; e
4. número de iterações entre as reinicializações das penalidades igual a 30.

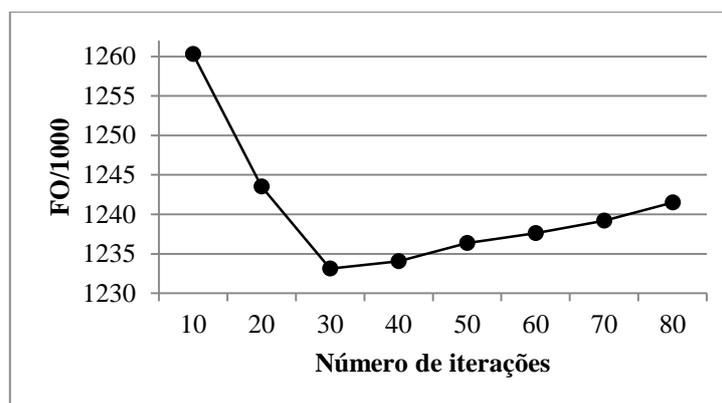


Gráfico 2. Variação do valor da FO em função do intervalo entre as reinicializações

Tabela 2. Dados da escala de mão de obra operada pela empresa.

	<i>Segunda</i>	<i>Terça</i>	<i>Quarta</i>	<i>Quinta</i>	<i>Sexta</i>	<i>Sábado</i>	<i>Domingo</i>
HE	86:41	85:55	106:05	120:14	108:11	54:35	26:57
Jornadas	134	130	149	162	155	124	68
DP	6	3	5	4	1	0	0
FO	1.364.404	1.322.420	1.518.460	1.651.256	1.576.564	1.253.100	686.468

Tabela 3. Resultados obtidos pela GLS com peso 5000 para as duplas pegadas

	<i>Segunda</i>	<i>Terça</i>	<i>Quarta</i>	<i>Quinta</i>	<i>Sexta</i>	<i>Sábado</i>	<i>Domingo</i>
FO Médio	1.287.299	1.236.780	1.472.452	1.599.237	1.490.054	1.152.544	608.910
Melhor FO	1.276.280	1.216.340	1.455.988	1.581.920	1.478.832	1.146.440	601.560
DP	12	8	10	13	8	7	5
HE	67:50	68:05	66:37	70:30	78:28	47:40	27:20
Jornadas	120	116	139	150	142	110	57
Desvio	0,86%	1,65%	1,12%	1,08%	0,75%	0,53%	1,21%

6.2. Testes de Desempenho da GLS

Os testes computacionais apresentados a seguir mostram uma comparação dos resultados obtidos pela GLS e aqueles produzidos pela metaheurística VNS-VLNS apresentada por Reis e Silva (2012) por este trabalho conter um conjunto maior de resultados computacionais. Também é apresentada a comparação com a solução adotada pela empresa. Foi utilizado um conjunto de sete problemas referentes a uma semana de trabalho de uma empresa de transporte público que atua na cidade de Belo Horizonte. Foram realizadas 10 execuções para cada problema, sendo que cada execução teve a duração de 30 minutos.

Os resultados apresentados nesta seção foram divididos em duas categorias, as quais consideram diferentes valores para o custo da dupla pegada. Como a empresa normalmente opera com no máximo 20% de jornadas do tipo dupla pegada é interessante avaliar o impacto gerado na quantidade de jornadas deste tipo ao variar o custo imposto para essa característica. Uma vez que a implementação da metaheurística VNS não considera o custo da hora ociosa na sua função objetivo, foi adotado o custo zero para esta componente da função objetivo da GLS. A Tabela 2 apresenta as características das soluções adotadas pela empresa para os problemas dos sete dias estudados, sendo que HE corresponde ao total de horas extras no formato hh:mm, Jornadas ao número de jornadas, DP ao total de jornadas do tipo dupla pegada e FO corresponde ao valor da função objetivo associado à solução.

6.2.1. Testes com coeficiente 5.000 para a dupla pegada

A Tabela 3 contém os resultados obtidos pela metaheurística GLS. Cada coluna apresenta os resultados obtidos para o problema daquele dia da semana. A linha FO Média apresenta o valor médio da função objetivo obtido pelas 10 execuções do algoritmo para o respectivo problema. Os dados seguintes se referem ao detalhamento da melhor solução obtida dentre as 10 execuções. A linha Melhor FO apresenta o valor da FO da melhor solução. As demais linhas têm o mesmo significado da Tabela 2. A linha Desvio mostra o desvio médio do melhor valor da FO em relação ao valor da FO Média, dada por $(FO\ Média - Melhor\ FO)/FO\ Média$. Quanto menor for esta porcentagem, mais robusto é o método, ou seja, a diferença entre as diversas soluções encontradas não é significativa e a metaheurística tem a capacidade de produzir soluções muito parecidas. A Tabela 4 apresenta os resultados obtidos com o VNS-VLNS.

Observando as Tabelas 3 e 4 é possível concluir que a GLS produziu os melhores resultados para três dos sete dias da semana, sendo que a FO Média do GLS é muito próxima da FO Média do VNS. Para realizar uma análise mais detalhada da eficiência das metaheurísticas, é apresentada na Tabela 5 a diferença entre os resultados da FO Média e Melhor FO obtidos pelo VNS menos os valores obtidos pelo GLS, assim como a porcentagem desta diferença em relação aos valores obtidos pela GLS. Nesta comparação, os valores negativos representam vantagem para o VNS e valores positivos, vantagem para a GLS.

Analisando as diferenças é possível verificar que embora os resultados do GLS sejam melhores do que o VNS em três dos sete dias, as maiores melhorias se devem ao GLS. Enquanto as maiores diferenças do VNS em relação à

Tabela 4. Características das soluções obtidas pelo VNS-VLNS

	<i>Segunda</i>	<i>Terça</i>	<i>Quarta</i>	<i>Quinta</i>	<i>Sexta</i>	<i>Sábado</i>	<i>Domingo</i>
FO Média	1.282.988	1.232.681	1.484.256	1.602.478	1.487.842	1.164.025	606.620
Melhor FO	1.270.628	1.213.176	1.471.176	1.583.644	1.471.100	1.152.552	601.296
DP	11	11	11	17	12	10	5
HE	65:07	75:44	67:24	77:41	87:55	52:18	27:35
Jornadas	120	114	140	148	139	109	57
Desvio	0,97%	1,61%	0,89%	1,19%	1,14%	1,00%	0,88%

Tabela 5. Comparação entre as soluções obtidas pelo VNS-VLNS e o GLS

	<i>Segunda</i>	<i>Terça</i>	<i>Quarta</i>	<i>Quinta</i>	<i>Sexta</i>	<i>Sábado</i>	<i>Domingo</i>
Diferença FO Média	-4.311	-4.099	+11.804	+3.241	-2.212	+11.481	-2.290
% Dif/GLS	-0,33%	-0,33%	+0,80%	+0,20%	-0,15%	+1,00%	-0,38%
Diferença Melhor FO	-5.652	-3.164	+15.188	+1.724	-7.732	+6.112	-264
% Dif/GLS	-0,439%	-0,256%	+1,031%	+0,108%	-0,519%	+0,530%	-0,043%

Tabela 6. Resultados da GLS com peso 600 para as duplas pegadas)

	<i>Segunda</i>	<i>Terça</i>	<i>Quarta</i>	<i>Quinta</i>	<i>Sexta</i>	<i>Sábado</i>	<i>Domingo</i>
FO Médio	1.264.666	1.236.004	1.443.009	1.572.460	1.487.006	1.117.079	587.960
Melhor FO	1.253.532	1.222.920	1.417.292	1.557.100	1.474.588	1.104.096	581.468
DP	23	23	14	22	19	17	9
HE	40:33	38:00	78:43	57:55	54:57	57:54	74:14
Jornadas	123	120	139	153	145	108	57
Desvio	0,88%	1,06%	1,78%	0,98%	0,84%	1,16%	1,10%

Tabela 7. Características das soluções obtidas pelo VNS-VLNS.)

	<i>Segunda</i>	<i>Terça</i>	<i>Quarta</i>	<i>Quinta</i>	<i>Sexta</i>	<i>Sábado</i>	<i>Domingo</i>
FO Média	1.235.435	1.203.345	1.409.582	1.534.114	1.451.791	1.106.525	572.411
Melhor FO	1.223.556	1.191.952	1.380.000	1.517.168	1.435.332	1.089.764	562.444
DP	19	14	19	17	14	10	7
HE	50:39	56:28	77:30	70:42	70:33	57:21	34:21
Jornadas	120	117	135	149	141	107	55
Desvio	0,96%	0,95%	2,10%	1,10%	1,13%	1,51%	1,74%

GLS são de 0,38% e 0,33%, a GLS alcança melhorias de 1,00% e 0,8% em relação ao VNS.

Em relação à qualidade das soluções, pode ser observado que o total de jornadas do tipo dupla pegada e a variação destes totais é bem maior entre as soluções do VNS, que estão entre 11 e 17, do que entre as soluções da GLS, que ficam entre 8 e 13.

De uma maneira geral as soluções são muito parecidas, visto que as diferenças das melhores soluções obtidas pelas metaheurísticas variam, em valores absolutos, entre 0,043% e 1,031%. Isso mostra que ambas apresentam o mesmo nível de qualidade nas soluções encontradas.

6.2.2. Testes com coeficiente 600 para a dupla pegada

Na Tabela 6 são apresentados os resultados obtidos pela metaheurística GLS considerando o custo de 600 para as duplas pegadas. Com este novo peso é gerado um número maior de duplas pegadas do que a quantidade máxima permitida pela empresa, desta forma estes testes servem para verificar o impacto desta flexibilização nos resultados obtidos. Em contrapartida a quantidade de horas extras é reduzida.

Na Tabela 7 são apresentados os resultados apresentados por Reis e Silva (2012), para os mesmos casos, utilizando a metaheurística VNS-VLNS.

Neste caso o VNS produz melhores soluções em todos os dias da semana, inclusive com um número inferior de duplas pegadas em relação à GLS.

6.3. Análise dos Resultados

Os resultados obtidos na primeira etapa dos testes computacionais, ou seja, considerando o peso de 5.000 para as duplas pegadas, mostraram que a GLS foi superior ao VNS-VLNS em 3 dos 7 problemas resolvidos. Isso se deve principalmente ao fato desta metaheurística produzir, na maioria dos casos, um número menor de duplas pegadas na solução e esta componente apresentar um custo elevado na função objetivo.

Nos testes realizados na segunda etapa, o custo das duplas pegadas foi reduzido e a metaheurística GLS não foi capaz de reduzir o número de jornadas além da redução das jornadas do tipo dupla pegada. Desta forma a GLS produziu resultados piores do que o VNS-VLNS. Uma possível explicação para este comportamento da GLS, quando aplicada ao PPT, é o fato da metaheurística penalizar os componentes das jornadas, ou seja, as horas extras, as horas ociosas e

as duplas pegadas, sendo que o número de jornadas nunca é penalizado. Desta forma, pode haver uma redução nas componentes penalizadas e um aumento no número de jornadas contidas na solução. Uma maneira de solucionar este problema seria uma combinação de dois procedimentos, sendo um deles responsável por melhorar as características das jornadas e outro responsável pela redução no número de jornadas na solução.

7. CONCLUSÕES

Este trabalho apresentou a utilização da metaheurística *Guided Local Search* para resolver o Problema de Programação de Tripulações do Sistema de Transporte Público. Esta metaheurística é pouco utilizada e sua aplicação no referido problema é realizada pela primeira vez neste trabalho, dado o conhecimento dos autores sobre o assunto. Com a aplicação da GLS foi possível obter, em alguns casos, soluções melhores do que aqueles obtidos pela metaheurística VNS-VLNS, largamente utilizada.

A implementação da GLS é relativamente simples e sua estratégia para sair de ótimos locais se diferencia das demais por atribuir penalizações às componentes de maior impacto na função objetivo. Desta forma, o espaço de busca sofre uma transformação e o procedimento de busca local é capaz de atingir novas soluções que são ótimas locais.

Este trabalho pode ser aprimorado incluindo o número de jornadas da solução na função objetivo. Outra possibilidade é fazer dois tipos de busca, sendo que o primeiro tem como objetivo minimizar o total de duplas e o segundo, a redução de horas extras, horas ociosas e de duplas pegadas. De qualquer forma, este trabalho abre novas perspectivas na exploração de uma metaheurística de fácil implementação aplicada a problemas relacionados com a operação de sistemas de transporte público.

AGRADECIMENTOS

Os autores agradecem ao CNPq, à Fapemig e à Universidade Federal de Ouro Preto (UFOP) pelo apoio recebido durante a realização deste trabalho. Os autores agradecem também aos *referees* anônimos por suas contribuições valiosas ao trabalho.

REFERÊNCIAS

- Adams, J., Balas, E., E Zawack, D. (1988). The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science*, v. 34, n. 3, p. 391–401. doi:dx.doi.org/10.1287/mnsc. p. 34.3.391
- Ahuja, R. K., Orlin, J. B., E Sharma, D. (2000). Very large-scale neighborhood search. *International Transactions in Operational Research*, 7(4-5), 301–317. doi:10.1111/j.1475-3995.2000.tb00201.x
- Balas, E., E Vazacopoulos, A. (1998). Guided Local Search with Shifting Bottleneck for Job Shop Scheduling. *Management Science*, v. 44, n. 2, p. 262–275. doi:dx.doi.org/10.1287/mnsc. p. 44.2.262
- 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*, v. 23, p. 1–33.
- Bouzada, C. F. (2003). *Custo do transporte coletivo por ônibus*. Belo Horizonte, MG: C/Arte,
- Desrochers, M., E Soumis, F. (1989). A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, v. 23, n. 1, p. 1–13. doi:10.1287/trsc.23.1.1
- Ernst, A. T., Jiang, H., Krishnamoorthy, M., E Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, v. 153, n. 1, p. 3–27. doi:10.1016/S0377-2217 (03)00095-X
- Fischetti, M., Martello, S., E Toth, P. (1987). The fixed job schedule problem with spread-time constraints. *Operations Research*, v. 35, n. 6, p. 849–858. doi:10.1287/opre.35.6.849
- Fores, S., Proll, L., E Wren, A. (1999). An improved ILP system for driver scheduling. In: Wilson, N. H. M. (Ed) *Computer-Aided Transit Scheduling*, p. 43–61. Springer. doi:10.1007/978-3-642-85970-0_3
- Friberg, C., E Haase, K. (1999). An exact branch and cut algorithm for the vehicle and crew scheduling problem. In: Wilson, N. H. M. (Ed) *Computer-Aided Transit Scheduling*, p. 63–80. Springer. doi: 10.1007/978-3-642-85970-0_4
- Li, J., E Kwan, R. S. K. (2003). A fuzzy genetic algorithm for driver scheduling. *European Journal of Operational Research*, v. 147, n. 2, p. 334–344. doi: 10.1016/S0377-2217(02)00564-7
- Lin, S., E KERNIGHAN, B. W. (1973). An Effective Heuristic Algorithm for the Travelling-Salesman Problem. *Operations Research*, 1, p. 498–516,
- Lourenço, H. R., Paixao, J. M. P., E Portugal, R. (2001). Multiobjective metaheuristics for the bus driver scheduling problem. *Transportation Science*, v. 35, n.3, p. 331–343. doi:10.1287/trsc.35.3.331.10147
- Marinho, E. H., Ochi, L. S., Drummond, L. M. A., M.J.F., S., E Silva, G. P. (2004). Busca Tabu aplicada ao problema de programação de tripulações de ônibus urbano. *Anais do XXXVI Simposio Brasileiro de Pesquisa Operacional SBPO*, p. 1471–1482.
- Mladenovic, N., E Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, v. 24, p. 1097–1100. doi:10.1016/S0305-0548(97)00031-2
- Reis, A. F. Da S., E Silva, G. P. (2012). Um estudo de diferentes métodos de busca e a metaheurística VNS para otimizar a escala de motoristas de ônibus urbano. *Transporte em Transformação XVI - Trabalhos Vencedores do Prêmio CNT - Produção Acadêmica 2011*, p. 47–64.
- Shen, Y., E Kwan, R. S. K. (2001). Tabu search for driver scheduling. In: S. Voss & J. R. Daduna (Eds), *Computer-Aided Scheduling of Public Transport*, p. 121–135. Springer. doi:10.1007/978-3-642-56423-9_7
- Silva, G. P., E Cunha, C. B. (2010). Uso da Técnica de Busca em Vizinhança de Grande Porte para a Programação da Escala de Motoristas de Ônibus Urbano. *Transportes*, v. 18, p. 64–75. doi:10.4237/transportes.v18i2.422
- Silva, G. P., E Reis, A. F. Da S. (2014). A study of different metaheuristics to solve the urban transit crew scheduling problem. *Journal of Transport Literature*, v. 8, n. 4, p. 227–251. doi:10.1590/2238-1031.jtl.v8n4a9

Smith, B. M., E Wren, A. (1988). A Bus Crew Scheduling System Using a Set Covering Formulation. *Transportation Research*, v. 22A, p. 97–108.
doi:10.1016/0191-2607(88)90022-2

Souza, M. J. F., Cardoso, L. X. T., Silva, G. P., Rodrigues, M. M. S., E Mapa, S. M. S. (2004). Metaheurísticas aplicadas ao Problema de Programação de Tripulações no sistema de transporte público. *Tendências em Matemática Aplicada e Computacional*, v. 5, n. 12, p. 357–368.

Talbi, E. (2009). *Metaheuristics: From Design to Implementation*. (First Edit.). John Wiley and Sons.
doi:10.1002/9780470496916.fmatter

Tsang, E. P. K., E Voudouris, C. (1997). Fast Local Search and Guided Local Search and Their Application to British Telecom's Workforce Scheduling Problem. *Operations Research Letters*, v. 20, n. 3, p. 119–127. doi:10.1016/S0167-6377(96)00042-9

Voudouris, C. (1997). *Guided Local Search for Combinatorial Optimisation Problems*. PhD. Thesis, University of Essex - United Kingdom, Department of Computer Science.

Voudouris, C., E Tsang, E. P. K. (1999) Guided Local Search and its application to the Travelling Salesman Problem. *European Journal of Operational Research*, v. 113, n. 2, p. 469–499. doi: 10.1016/S0377-2217(98)00099-X

Voudouris, C., E Tsang, E. P. K. (2003). Guided Local Search. *Handbook of Metaheuristics*, v. 57, p. 185–218.