

UMA APLICAÇÃO DA METAHEURÍSTICA *GUIDED LOCAL SEARCH* AO PROBLEMA DE PROGRAMAÇÃO DE MOTORISTAS DE ÔNIBUS URBANO

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

Tiago Alves Silva

Universidade Federal de Ouro Preto – Departamento de Computação
Campus Morro do Cruzeiro, Ouro Preto, MG – Brasil - 35400-000
thiago.silva1230@gmail.com

RESUMO

Neste trabalho é utilizada a metaheurística *Guided Local Search* (GLS) para resolver o Problema de Programação de Tripulações de Ônibus Urbano (PPT). O PPT consiste em gerar jornadas de trabalho para os tripulantes que conduzirão uma frota em operação com o menor custo. Assim, procura-se reduzir os custos fixos e variáveis referentes às tripulantes. A GLS baseia-se em penalizar características indesejáveis na solução do problema, transformando o espaço de busca. Foi utilizado o método de busca *Variable Neighborhood Descent*, que percorre diferentes estruturas de vizinhança para encontrar um mínimo local. A implementação proposta foi testada com dados de problemas reais de uma empresa de médio porte e os resultados foram comparados com resultados da literatura que utilizaram as metaheurísticas *Variable Neighborhood Search* e *Iterated Local Search*. Os resultados são similares àqueles apresentados 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.

Área principal. Logística e Transportes, Metaheurísticas

ABSTRACT

In this work the metaheuristic *Guided Local Search* (GLS) is used to solve the Crew Scheduling Problem for Urban Busses (CSP). The CSP consists on generating daily shifts to the crews responsible to conduct the busses with the lowest cost. GLS is a metaheuristic whose strategy employed to scape from local optimum consists on penalizing undesirable characteristics into the solution. So, the search space is transformed and a new local optimal can be achieved. It was adopted the local search heuristic *Variable Neighborhood Descent*, which explores different neighborhood structures to find a local optimum. The proposed implementation was tested with data from a medium sized company and the results were compared with those found in literature using the metaheuristics *Variable Neighborhood Search* and *Iterated Local Search*. The results obtained are similar to those already known, being possible to improve it once the GLS can be exploited in different ways.

KEYWORDS. *Guided Local Search*, Mass Transit Crew Scheduling Problem, Metaheuristic.

Main area. L&T

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 para auxiliar o processo de decisão. Neste contexto é que se inserem os métodos de otimização, os quais podem levar à redução dos custos dessas empresas. Entre os custos de maiores pesos se destacam os custos operacionais com os veículos e os seus motoristas e cobradores (Bouzada, 2003).

Este trabalho está voltado para a resolução do problema de alocação das tripulações de ônibus urbano que atuam no sistema de transporte público, 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 suas jornadas de trabalho, de tal forma que sejam cobertas todas as viagens da frota em operação com o menor custo possível. 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 das empresas. 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 de jornada, entre outras. Devido a estas regras, o problema se torna NP-Difícil, para o qual não se conhece algoritmo polinomial para a sua resolução (Fischetti *et al.*, 1987). Desta forma, se faz necessário utilizar métodos heurísticos para a resolução de problemas de grande porte de forma satisfatória.

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 difere das demais metaheurísticas por aplicar penalizações às características indesejáveis da solução corrente para escapar de ótimos locais. Assim, pode-se afirmar que a GLS “deforma” o espaço de soluções 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 solução na função objetivo. Quando a busca se prende em um mínimo local é selecionado o atributo de maior peso na solução e este atributo é penalizado. A escolha dos elementos a serem penalizados é um ponto importante na GLS. A metaheurística tenta encontrar a característica que irá gerar o maior impacto na função objetivo. Para tanto, a cada iteração, o algoritmo analisa todos os atributos e escolhe aquele de maior impacto. Este procedimento acelera a saída do mínimo local e permite explorar outras regiões do espaço de soluções.

A resolução do tem como objetivo minimizar os custos referentes à remuneração das tripulações. Neste caso, quando as penalidades são aplicadas, a função objetivo aumenta seu valor e um procedimento de busca local é executado por um determinado número de iterações. Portanto, quando são aplicadas sucessivas penalizações a um mínimo local, o valor desta solução aumenta até que esta deixe de ser um mínimo local. Com isso, a busca local irá encontrar outro mínimo local no espaço de soluções. Após um determinado número de iterações as penalizações são zeradas 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 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).

Este 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, as linhas gerais da metaheurística GLS e como ela foi adaptada para resolver o PPT. Na seção 5 são apresentados os testes de calibração da GLS, os resultados obtidos e a comparação com os resultados conhecidos. Finalmente, na seção 6 são apresentadas as conclusões do trabalho.

2. Revisão Bibliográfica

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* (Friberg e Haase, 1999). Por outro lado, diversas metaheurísticas foram implementadas para resolver problemas de grande porte. Dentre essas destacam-se: Algoritmos Genéticos, GRASP, Busca Tabu, *Simulated Annealing*, VNS, entre outros (Shen e Kwan, 2001; Silva e Cunha, 2010; Lourenco *et al.*, 2001; Souza *et al.*, 2004; Silva e Reis 2014).

Com o emprego de metaheurísticas é possível obter resultados muito interessantes para o problema, embora a otimalidade das soluções não sejam garantidas. Os trabalhos de Ernst *et al.* (2004a, 2004b) apresentam uma série de problemas relacionados à alocação de pessoal em diversas áreas, incluindo motoristas e cobradores de sistemas de transportes. Os autores descrevem problemas de alocação de funcionários em seus diferentes modais de transportes: no aeroviário, ferroviário, transporte urbano e no transporte interurbano. É apresentada uma revisão sobre a alocação de máquinas e pessoal do sistema de transporte público, envolvendo todas as suas etapas. Segundo Ernst *et al.* (2004b), 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 das tripulações. Desta forma, a complexidade do problema é reduzida, permitindo sua resolução de forma satisfatória.

Lourenço *et al.* (2001) utilizam duas metaheurísticas multiobjetivo para solucionar o PPT: a Busca Tabu e um Algoritmo Genético. São consideradas duas funções objetivo, de custos e de penalidades. Na Busca Tabu, os autores consideram duas listas tabu, uma de inserção e outra de remoção de jornadas. A Busca Tabu é realizada para cada função objetivo e posteriormente é executada com uma função de soma ponderada. Em um processo de intensificação, a Busca Tabu é executada 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 diversidade 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 é proposto por Li e Kwan (2003) para o PPT. Esta abordagem foi testada tanto com problemas de ônibus urbano quanto em linhas de trens de empresas inglesas. Uma heurística gulosa é utilizada para construir uma escala onde as jornadas são selecionadas sequencialmente a partir 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. 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 esta 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). Nessa 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 deve ser realizada pela mesma tripulação por não haver tempo hábil para a sua troca. A Lista Tabu é utilizada para armazenar as últimas soluções avaliadas, de modo a não permitir que o método volte para uma solução anteriormente visitada. Três implementações da Busca Tabu foram abordadas no trabalho: a Busca Tabu com

primeiro vizinho de melhora em uma parte da vizinhança; a Busca Tabu com melhor vizinho em uma vizinhança variável e a Busca Tabu com primeiro vizinho de melhora em uma parte da vizinhança com diversificação. Os testes realizados com os três tipos de busca obtiveram resultados relevantes ao serem comparados com outra heurística da literatura e com as soluções adotadas pela empresa.

O trabalho desenvolvido por Silva e Cunha (2010) apresenta o uso da metaheurística GRASP combinado com a busca em vizinhança de grande porte *Very Large-scale Neighborhood Search* – VLNS (Ahuja *et al.*, 2000) para resolver o PPT. O método de busca local 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 proporcionando 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 os resultados 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. Aliados a ela foram usados dois métodos de busca local: a *Variable Neighborhood Descent* - VND e a *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. 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 considerar o objetivo da empresa para escolher a técnica.

De acordo com uma pesquisa realizada pelos autores, não foi detectado qualquer trabalho que utilizasse a metaheurística GLS para resolver o PPT. Assim, este trabalho foi concebido para explorar tal combinação e verificar a eficiência desta metaheurística na resolução do PPT de grande complexidade.

3. O Problema de Programação de Tripulações

O Problema de Programação de Tripulações - PPT consiste em gerar jornadas de trabalho para os funcionários que conduzirão os veículos em operação, ou seja, gerar as jornadas para as tripulações: a dupla motorista e cobrador. Os dados de entrada do PPT são: *i*) a programação detalhada dos veículos que serão conduzidos; *ii*) as regras operacionais da empresa; e *iii*) as leis trabalhistas.

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 das tripulações. 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 viagens consecutivas do veículo é maior do que certo intervalo de tempo, previamente definida em um local apropriado. Caso contrário, as viagens devem ser agrupadas. Assim, todas as viagens realizadas pelos veículos da frota da empresa são agrupadas em tarefas que devem ser alocadas às tripulações com o menor custo possível.

Uma jornada é do tipo *dupla pegada* quando existe um intervalo entre duas tarefas maior do que um dado intervalo de tempo, no caso de duas horas. Este tipo de jornada existe para cobrir a demanda por viagens extras que ocorrem somente nos horários do pico da manhã e da tarde. As demais jornadas são ditas normais, ou do tipo *pegada simples*. A remuneração fixa de uma tripulação se refere ao 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 da escala.

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

As jornadas de trabalho devem respeitar uma série de acordos coletivos e de leis trabalhistas. As regras de trabalho consideradas neste trabalho e adotadas na prática são:

- i) Cada jornada pode conter no máximo 2 horas extras;
- ii) Toda jornada deve permitir que haja um tempo mínimo de descanso, em casa, de 11 horas;
- iii) Toda jornada de trabalho é remunerada por 6 horas e 40 minutos.
- iv) Nas jornadas do tipo dupla pegada, o intervalo maior ou igual a 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 jornadas. Os custos fixos decorrem da remuneração dos tripulantes e os custos variáveis representam as horas extras trabalhadas, o tempo ocioso e a quantidade de duplas pegadas na solução. A quantidade de duplas pegadas deve ser controlada visto que suas tripulações têm direito de folgar nos domingos. Desta forma, este tipo de jornada deve ser limitado. A expressão a seguir representa a função objetivo de uma solução do PPT.

$$C = \sum_{i=1}^{tot_jorn} CF_i + w_1 \times h_extra_i + w_2 \times h_ociosa_i + w_3 \times Dupla_pegada_i \quad (1)$$

Na expressão (1), *tot_jorn* corresponde ao total de jornadas na solução, CF_i representa a remuneração fixa da jornada i , w_1 representa o peso de cada minuto de trabalho extra, h_extra_i representa a quantidade de horas extras na jornada i , expressa em minutos, w_2 representa o peso de cada minuto de tempo ocioso, h_ociosa_i representa o tempo ocioso, expresso em minutos, w_3 representa o peso atribuído às duplas pegadas e $Dupla_Peg_i$ 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 *Guided Local Search* - GLS é uma metaheurística de propósito geral que guia uma estratégia de busca local dando-lhe mais eficiência e robustez (Voudouris e Tsang, 2003). Ela se baseia em penalizar sucessivamente as componentes de maior custo das soluções ótimas locais até que a busca local se desloque para outra região do espaço de soluções. A cada iteração, a função objetivo é incrementada por meio de penalizações e a busca local considera essa “função aumentada”. Para tanto, deve-se definir um conjunto de *atributos* ou *características* a serem penalizadas. Os atributos são componentes da solução que fazem parte da função objetivo. A metaheurística parte de uma solução inicial, (Figura 1, $x = a$), realiza uma busca local e chega a um ótimo local. Quando o ótimo local é atingido, a GLS seleciona alguns atributos que estão presentes na solução encontrada e os penaliza (Figura 1, $x = b$). Assim, a solução se modifica até que seu valor não seja mais um ótimo local. Quando isso acontece, o algoritmo de busca local irá escapar deste ponto e encontrar um novo ótimo local (Figura 1, $x = c$). Esse processo se repete até que algum critério de parada seja satisfeito.

Para acelerar a saída do ótimo local, a GLS penaliza os atributos de maior impacto na função objetivo. A partir de um ótimo local, o procedimento analisa os atributos presentes na solução, e seleciona o(s) mais significativo(s) e aplica a penalização neste(s). À medida que ocorre a penalização de um atributo, seu contador é incrementado. Isso é feito para que as penalizações não ocorram sempre sobre os mesmos atributos, para a diversificação das soluções. Assim, a GLS aumenta a função objetivo acrescentando penalidades a ela.

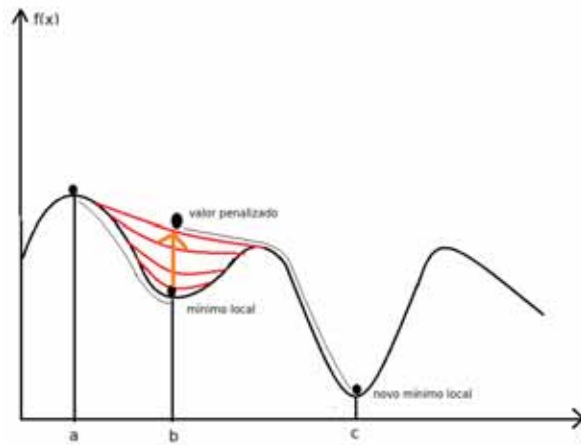


Figura 1. Deformação do espaço de soluções causado pela GLS

Para aplicar a GLS, é preciso definir os atributos a serem penalizados. As penalizações são inicializadas 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 $g(s)$, 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 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 a serem penalizados. A ideia é penalizar “características desfavoráveis” ou “características de maior importância” quando a busca 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 original e p_i é 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 for 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 soluções, 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 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 α 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. Quando isso não for possível, pode-se usar uma lista com os atributos presentes na solução que está sendo analisada. Esta 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 dos atributos a serem penalizados influencia fortemente o resultado, pois é com base em seus valores que ocorrem as penalizações. O pseudocódigo da GLS é apresentado a seguir.

Algoritmo Guided Local Search
Função GLS ($p, g, \lambda, [I_1, \dots, I_M], [c_1, \dots, c_M], M$)
 $k := 0;$
 $s := Solucao_Inicial();$
para ($i = 1$ até M) **faça**
 $p_i := 0;$ //zera as penalidades
fim_para
 $h := g + \lambda \times \sum_{i=1}^M p_i \times I_i$ //acrescenta a penalização à função original g
enquanto (critério de parada não satisfeito) **faça**
 $S_{k+1} := BuscaLocal(S_k, h)$ //realiza a busca local considerando a função h
 para ($i = 0$ até M) **faça**
 $util_i := I_i(S_{k+1} \times c_i / (1 + p_i))$ //calcula a utilidade do atributo i
 fim_para
 para cada $i \in MAX(util_i)$ **faça** //penaliza o(s) atributo(s)
 $p_i := p_i + 1$
 fim_para
fim_enquanto
 $s^* := Min(s)$ //retorna a melhor solução encontrada
retorna s^*
fim_função

Algoritmo1. Pseudo-código da metaheurística GLS

A metaheurística GLS detém bons resultados para o problema de atribuição de frequência de rádio e de agendamento de mão de obra. Para o problema do caixeiro viajante, Voudouris e Tsang (1999) usou um algoritmo GLS+FLS+20pt e conseguiu resultados empíricos melhores do que aqueles encontrados pelo algoritmo Lin-Kernighan - LK (Lin e Kernighan, 1973), que é um algoritmo específico para tratar o problema do caixeiro viajante. Além desse algoritmo, o GLS+FLS+20pt também obteve melhor resultado empírico que as implementações de *Simulated Annealing*, Busca Tabu e Algoritmos Genéticos para o Problema do Caixeiro Viajante. Quando aplicado ao problema de factibilidade SAT e MAX-SAT o GLS apresenta bons resultados, comparáveis aos produzidos pelos algoritmos tidos como estado da arte para o problema. No problema de designação, uma implementação que utiliza a GLS conseguiu resultados tão bons quantos os obtidos pela técnica tida como estado da arte para o problema (Lau e Tsang, 1998).

4.1 Aplicação da Guided Local Search ao Problema de Programação de Tripulações

Para aplicar o GLS, o PPT foi modelado como um problema de particionamento das tarefas entre as tripulações gerando suas jornadas de trabalho. O particionamento é tal que as jornadas devem respeitar todas as restrições do problema. O critério de parada adotado para a GLS foi o tempo de processamento, no caso de trinta minutos.

4.1.1 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. 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.

4.1.2 Estrutura de Vizinhança

As melhorias na função objetivo se dão devido à realocação e troca de tarefas entre as jornadas. Com base nestes dois movimentos é que foi definida a estrutura de vizinhança do problema. As diferentes estruturas de vizinhança são definidas pela quantidade de tarefas subsequentes a serem realocadas ou trocadas entre duas jornadas. Com base em experimentos realizados, o tamanho da estrutura de vizinhança adotada foi de 4 tarefas. Com isso, até 4 tarefas podem ser removidas de uma jornada e inseridas em outra. Esse movimento pode ser de dois tipos: realocação ou troca. A partir de duas jornadas i e j tomadas aleatoriamente, retira-se k tarefas consecutivas de i e as insere em j , podendo ocorrer uma das seguintes situações:

- as k tarefas podem ser inseridas em j sem a necessidade da retirada de nenhuma tarefa de j . Neste caso é realizada uma realocação das k tarefas da jornada i para a jornada j ;
- 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 resultantes respeitarem todas as restrições do problema.

4.1.3 Função Objetivo e Atributos

A Função Objetivo (FO) original do problema foi modificada inserindo-se as penalizações previstas na GLS. Para o PPT foram utilizadas múltiplas penalizações para as soluções de mínimo local. Foram escolhidos os atributos: *i*) a quantidade de horas extras, *ii*) o tempo ocioso, e *iii*) a quantidade de duplas pegadas. Assim, a cada iteração da GLS a FO recebe 3 penalizações, uma sobre cada atributo que apresentar o maior valor. Essas penalizações não são necessariamente aplicadas à mesma jornada. São penalizadas as jornadas que possuem o maior valor de cada atributo. Com isso, a função objetivo do problema fica da seguinte forma:

$$C = \sum_{i=1}^{tot_jorn} CF_i + (w_1 + \lambda \times p_{i1}) \times h_extra_i + (w_2 + \lambda \times p_{i2}) \times h_ociosa_i + (w_3 + \lambda \times p_{i3}) \times Dupla_pegada_i \quad (5)$$

A função objetivo (5) é uma adaptação da função (1) onde foram inseridas as penalizações referentes à GLS. Em cada um dos atributos, foi inserido o parâmetro λ que é o coeficiente de penalização e p que representa a quantidade 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 no espaço de busca.

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). Esta variação da GLS zera 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 sejam descartadas boas soluções devido às penalizações. Esta 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.

4.1.4 Métodos de Busca Local

Neste trabalho foram testados os principais métodos de busca local descritos na literatura: *i*) o *best improvement*, *ii*) o *first improvement*, *iii*) *variable neighborhood descent* combinado com o *best improvement*, e *iv*) *variable neighborhood descent* combinado com o *first improvement*.

5. 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 adotados os seguintes parâmetros para o problema: *i*) custo da dupla CF_i : 10.000, *ii*) custo da dupla pegada: 5.000, *iii*) custo da hora extra em minutos: 4, e *iv*) custo da hora ociosa em minutos: 1. Tais valores são utilizados na expressão (5). Foram realizadas 10 execuções de 30 minutos para cada valor dos parâmetros.

5.1 Calibração dos Parâmetros da GLS

O processo de calibração define os parâmetros utilizados na metaheurística. Uma característica interessante da GLS é o fato apresentar um único parâmetro a ser calibrado, o λ . No entanto, pelo fato do problema apresentar os mesmos atributos em todas as componentes da solução, é necessário estabelecer a periodicidade em que devem ser zeradas as penalizações. Além disso, os testes servem para indicar o método de busca local a ser utilizado.

É apresentada a melhor solução obtida após 10 execuções para cada parâmetro. Nestes testes foi utilizado o valor de 5.000 para a dupla pegada e 10.000 para o custo fixo das duplas na expressão (5). O primeiro teste foi realizado para avaliar a eficiência dos quatro métodos de busca local considerados. Os resultados obtidos são apresentados na Tabela 1.

Tabela 1. Valor da FO para os diferentes métodos de descida e valor de k

Número de tarefas k	1	2	3	4
<i>Best improvement</i>	1.290.900	1.297.096	1.319.888	1.329.004
<i>First improvement</i>	1.286.372	1.307.012	1.319.468	1.312.448
VND + <i>best improvement</i>	1.290.840	1.276.404	1.276.176	1.266.792
VND + <i>first improvement</i>	1.286.408	1.266.968	1.268.128	1.257.864

Nestes testes, o valor k representa o número de tarefas consecutivas que são realocadas/trocadas entre duas jornadas. Os resultados mostram que as buscas locais utilizando o VND são mais eficientes do que as buscas usando somente as técnicas de *best improvement* e *first improvement*. Isso ocorre porque o VND tem liberdade de realizar trocas com vizinhanças de tamanho menor do que o limite a ele imposto. Ou seja, quando se passa o valor $k = 4$, o VND realiza trocas com 1, 2, 3 e 4 tarefas. O mesmo não ocorre para os demais métodos, que trabalham com exatamente 4 tarefas consecutivas. Com base nesses resultados, foi adotada a busca local VND com *first improvement* nos testes subsequentes.

Segundo os resultados da Tabela 1, o melhor tamanho para a vizinhança foi para $k = 4$. Como este valor já se encontra no limite de tarefas realizadas em uma jornada, não foi possível testar valores maiores.

Os resultados apresentados no Gráfico 1 foram gerados para calibrar o parâmetro λ . Os resultados mostram os valores médios obtidos para a FO após 10 execuções para cada valor de λ .

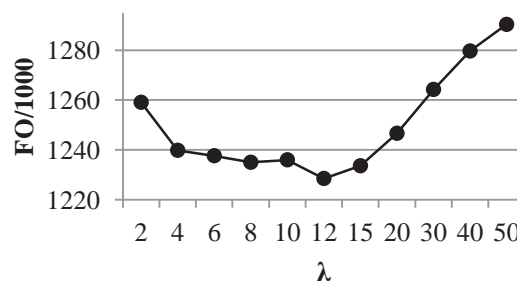


Gráfico 1. Variação do valor da função objetivo em função de lambda

No Gráfico 1 pode ser observado que a FO inicia com um valor alto, sendo reduzida com o aumento de λ . O menor valor da FO foi obtido para λ igual a 12, valor adotado no trabalho.

A última calibração foi realizada para encontrar o melhor valor para o número de iterações da GLS, até que as penalizações fossem reinicializadas (zeradas 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.

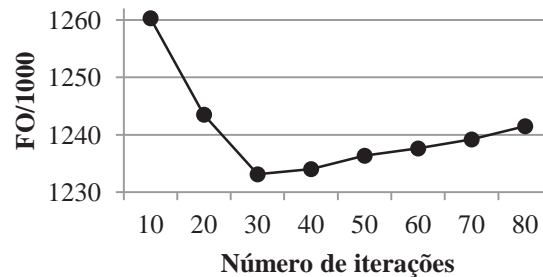


Gráfico 2. Variação do valor da FO em função do intervalo entre as reinicializações

É possível verificar no Gráfico 2 que o melhor intervalo foi de 30 iterações entre as reinicializações. Portanto, este valor foi adotado no trabalho. Desta maneira, foram definidos todos os parâmetros utilizados na metaheurística para a realização dos testes de eficiência.

5.2 Testes de Desempenho da GLS

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

Como a empresa opera normalmente com no máximo 20% de jornadas do tipo dupla pegada é interessante analisar a quantidade de jornadas deste tipo nas soluções. Uma vez que a implementação da VNS não considera o custo da hora ociosa, foi adotado o custo zero para esta componente da função objetivo da GLS.

A Tabela 1 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 a média da FO das 10 execuções, **Melhor FO** apresenta o valor da melhor solução, **DP** corresponde ao total de jornadas do tipo dupla pegada, **HE** ao total de horas extras (hh:mm), **Jornadas** ao número de jornadas, e **Desvio** o desvio médio do melhor valor dado por $(FO\ Média - Melhor\ FO)/FO\ Média$. Quanto menor for este valor, 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 2 contém as características das soluções obtidas por Reis e Silva (2012) para os mesmos problemas e os mesmos coeficientes.

Tabela 1. Resultados obtidos pela GLS com peso 5000 para as duplas pegadas

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
FO Média	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%

Tabela 2. Características das soluções obtidas pelo VNS-VLNS

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
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%

A partir das Tabelas 1 e 2 é possível concluir que a GLS produziu os melhores resultados para 3 dos 7 problemas, sendo que a FO Média do GLS é muito próxima da VNS. A Tabela 3 apresenta 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. Assim, os valores negativos representam vantagem para o VNS e vice-versa.

Tabela 3. Comparação entre as soluções obtidas pelo VNS-VLNS e o GLS

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
Diferença da 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 da 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%

Analisando as diferenças apresentadas na Tabela 3, é possível verificar que embora os resultados do GLS tenham sido melhores do que o VNS em apenas 3 dos 7 dias, as maiores melhorias se devem ao GLS. Enquanto as maiores diferenças do VNS em relação à 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 entre 0,043% e 1,031%. Isso mostra que ambas apresentam o mesmo nível de qualidade nas soluções encontradas.

5.3 Análise dos Resultados

Os resultados obtidos nos testes computacionais mostraram que a GLS foi superior ao VNS-VLNS em 3 dos 7 problemas resolvidos. Isso se deve principalmente pelo 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.

6. Conclusões

Este trabalho apresentou a utilização da metaheurística *Guided Local Search* para resolver o PPT do Sistema de Transporte Público. Esta metaheurística é pouco utilizada e sua aplicação no PPT é inédita na literatura, sendo que neste primeiro estudo foi possível obter, em alguns casos, soluções melhores do que aqueles obtidos pela metaheurística VNS, largamente utilizada. A implementação da GLS é relativamente simples e sua estratégia para sair de ótimos locais se diferencia das demais metaheurísticas 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.

Referências

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.

- 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.
- Bouzada, C. F.**, *Custo do transporte coletivo por ônibus*. Ed. C/Arte, Belo Horizonte/MG, 2003.
- Desrochers, M., e Soumis, F.** (1989) A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23(1), 1–13.
- Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., e Sier, D.** (2004a) An Annotated Bibliography of Personnel Scheduling. *Annals of Operations Research*, 127, 21–144.
- Ernst, A. T., Jiang, H., Krishnamoorthy, M., e Sier, D.** (2004b) Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1), 3–27.
- Fischetti, M., Martello, S., e Toth, P.** (1987) The fixed job schedule problem with spread-time constraints. *Operations Research*, 35(6), 849–858.
- Fores, S., Proll, L., e Wren, A.**, An improved ILP system for driver scheduling, N. H. M. Wilson (Ed.), *Computer-Aided Transit Scheduling*, Springer, Berlin, 43–61, 1999.
- Friberg, C., e Haase, K.** An exact branch and cut algorithm for the vehicle and crew scheduling problem, N. H. M. Wilson (Ed.), *Computer-Aided Transit Scheduling*, Springer, Berlin, 63–80, 1999.
- Lau, T. L., e Tsang, E. P. K.** (1998) The guided genetic algorithm and its application to the general assignment problem, *IEEE 10th International Conference on Tools with Artificial Intelligence (ICTAI'98)*, 336–343.
- Li, J., e Kwan, R. S. K.** (2003) A fuzzy genetic algorithm for driver scheduling. *European Journal of Operational Research*, 147(2), 334–344.
- Lin, S., e Kernighan, B. W.** (1973) An Effective Heuristic Algorithm for the Travelling-Salesman Problem. *Operations Research*, 21, 498–516.
- Lourenço, H. R., Paixao, J. M. P., e Portugal, R.** (2001) Multiobjective metaheuristics for the bus driver scheduling problem. *Transportation Science*, 35(3), 331–343.
- Marinho, E. H., Ochi, L. S., Drummond, L. M. A., Souza, M. J. F. 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*, 1471–1482.
- Reis, A. F. da S., e Silva, G. P.** 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*, 47–64, 2012.
- Shen, Y., e Kwan, R. S. K.**, Tabu search for driver scheduling. S. Voss e J. R. Daduna (Eds), *Computer-Aided Scheduling of Public Transport*, Springer, Berlin, 121–135 2001.
- 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*, 18, 64–75.
- 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*, 8(4), 227–251.
- Smith, B. M., e Wren, A.** (1988) A Bus Crew Scheduling System Using a Set Covering Formulation. *Transportation Research*, 22A, 97–108.
- 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*, 5(12), 357–368.
- Voudouris, C.**, *Guided Local Search for Combinatorial Optimisation Problems*. PhD. Thesis, University of Essex - United Kingdom, Department of Computer Science, 1997.
- Voudouris, C., e Tsang, E. P. K.** (1999) Guided Local Search and its application to the Travelling Salesman Problem. *European Journal of Operational Research*, 113(2), 469–499.
- Voudouris, C., e Tsang, E. P. K.** (2003) Guided Local Search. *Handbook of Metaheuristics*, 57, 185–218.