



Aplicação da Metaheurística *Iterated Local Search* à Programação de Veículos no Sistema de Transporte Público

Emiliana Mara Lopes Simões, Marcone Jamilson Freitas Souza, Gustavo Peixoto Silva

Universidade Federal de Ouro Preto (UFOP)

Departamento de Computação

mililopessimes@yahoo.com.br, marccone@iceb.ufop.br, gustavo@iceb.ufop.br

Ouro Preto, MG, CEP 35.400-000

RESUMO

Este trabalho trata da programação de veículos no Sistema de Transporte Público. Este problema consiste em criar uma rotina diária de operação para uma frota de veículos disponibilizada para uso de uma empresa, dado um conjunto de viagens com seus respectivos horários e pontos de partida e chegada e tempo de viagem. O objetivo é reduzir o custo operacional de tal atividade, fazendo o melhor aproveitamento da frota. Dada a sua complexidade combinatorial, o problema foi resolvido por meio da metaheurística *Iterated Local Search* (ILS). Para testar o método proposto foram utilizados dados reais de quatro empresas do Sistema de Transporte Público da cidade de Belo Horizonte (MG). Experimentos computacionais mostraram que o método ILS é capaz de gerar soluções finais de qualidade muito superior às utilizadas pelas empresas. Além disso, o método se mostrou robusto, pois há pouca variabilidade nas soluções finais.

Palavras-Chave: Programação de veículos, *Iterated Local Search*, Transporte Público.

ABSTRACT

This work deals with the vehicle scheduling in public mass transit. This problem consists of creating a daily routine of operation for a fleet of vehicles for use of a company, given a set of trips with its respective schedules, starting and ending times, points and travel times. The objective is to reduce the operational cost of such activity, being made the best use of the fleet. Given its combinatorial complexity, the problem was solved by means of the meta-heuristic *Iterated Local Search* (ILS). To test the proposed method, real instances of four companies of the Urban Public Transportation System of the Belo Horizonte city were used. Computational experiments showed that ILS is able to generate final solutions of quality better than the used ones by the companies. Moreover, the method is robust, since it has little variability in the final solutions.

Keywords: Vehicle Scheduling, *Iterated Local Search*, Mass Transit.

1. Introdução

Devido à sua complexidade, o processo de planejamento do sistema de transporte público é normalmente decomposto nos seguintes subproblemas: Definição das Rotas, Programação dos Horários das Viagens, Programação de Veículos, Programação de Tripulações e Rotação da Tripulação. A Figura 1 ilustra a relação entre esses subproblemas.

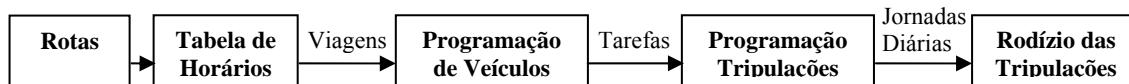


Figura 1 – Relação dos problemas envolvidos no Sistema de Transporte Público.

As definições das rotas e da tabela de horários do sistema se baseiam na disponibilidade de infraestrutura, nos serviços requeridos pelos usuários e em aspectos de demanda. Tais decisões não fazem parte da abrangência deste trabalho e são consideradas como conhecidas. Quanto ao tempo de duração de cada viagem, este pode variar ao longo do período de planejamento; entretanto é considerado determinístico e conhecido.

Estabelecidas as viagens a serem executadas, passa-se à resolução do Problema de Programação de Veículos (PPV), ou *Vehicle Scheduling Problem*, objeto deste trabalho. Este problema consiste em criar uma rotina diária de operação para uma frota de veículos disponibilizada para uso de uma empresa, tendo por objetivo fazer o melhor aproveitamento da frota.

Da programação dos veículos resulta um conjunto de blocos de viagens, sendo cada bloco uma seqüência de viagens realizadas por um mesmo veículo. Esse resultado da programação de veículos é a base para a resolução do problema de programação de tripulações (PPT). Para resolução do PPT, inicialmente os blocos de viagens são simplificados em blocos de tarefas. Cada tarefa consiste em um conjunto de viagens entre as quais não é possível realizar a troca de tripulações em função do tempo requerido para essa atividade. A solução do PPT é um conjunto de jornadas diárias de trabalho que agrupam tarefas satisfazendo a legislação trabalhista e as regras operacionais da empresa.

Resolvido o PPT para cada dia do mês, o problema seguinte é o da Rotação de Tripulações (PRT). Esse problema consiste em atribuir uma jornada de trabalho mensal a cada tripulação, considerando certas restrições trabalhistas restritas à programação mensal e que, portanto, não foram contempladas na programação diária.

Portanto, no objetivo de se minimizar os custos envolvidos nas atividades desempenhadas por um sistema de transporte público, a determinação de uma programação dos veículos de forma automatizada e eficiente se mostra muito importante.

2. Revisão de literatura

O PPV foi um dos primeiros problemas de transporte público a ser estudado e resolvido com o auxílio de computadores. Os primeiros trabalhos tratavam o problema tanto utilizando métodos exatos (Kirkman, 1968) quanto métodos heurísticos (Saha, 1970; Wren, 1972).

Martin-löf (1970) resolve o problema de programação de veículos considerando viagens com flexibilidade nos horários de partida. É apresentado um modelo de programação linear inteira, o qual é resolvido por um método simples de busca do tipo *branch-and-bound*. O método, entretanto, tem aplicabilidade limitada a problemas de pequeno porte.

Gavish *et al.* (1978) formulam o PPV como um problema de designação e pseudo-designação, que tem como objetivos primários minimizar o número de veículos utilizados durante o horário

de pico e minimizar o tempo de deslocamento fora de operação no horário entre picos. O objetivo secundário é minimizar as mudanças em relação à programação já existente.

Hoffstadt (1981) utiliza uma heurística de encadeamento de viagens para determinar o número mínimo de veículos. Posteriormente, para minimizar o custo operacional é aplicado o algoritmo húngaro (Kuhn, 1956) a um problema de designação.

Wren e Chamberlain (1988) incorporam a programação da tripulação ao sistema para programação de veículos, denominado VAMPIRES, de Wren (1972), que passa então a ser denominado BUSMAN, sendo a componente responsável pela programação de veículos denominada BUSPLAN.

Kwan e Rahin (1995) apresentam os módulos incorporados à componente BUSPLAN do sistema, os quais permitem estabelecer certa interatividade com o usuário. São apresentadas ferramentas semi-automáticas para coordenar a geração de novos horários das viagens com ligação infactível, além de melhoramentos nas heurísticas que tratam da programação com diferentes tipos de veículos.

Kwan e Rahin (1999) apresentam as características do BOOST, que é a versão orientada a objetos do sistema anterior, o BUSPLAN. Nesse sistema, uma solução inicial é gerada por meio de um método construtivo guloso que admite infactibilidade. A seguir, a programação é refinada por um procedimento iterativo *2-optimal*.

Psarras *et al.* (1997) utilizam as vantagens da *Constraint Logic Programming*, para gerar uma solução inicial para o Problema de Programação de Veículos e posteriormente avaliar as várias soluções intermediárias obtidas por uma busca local com o objetivo de verificar a conformidade com as restrições impostas ao problema. Segundo os autores, a grande vantagem desta abordagem é a obtenção de soluções em um tempo de execução satisfatório.

Em Atkinson (1998) uma heurística GRASP é aplicada a um PPV onde os intervalos de desembarque e embarque de um veículo são reduzidos, além de outras restrições adicionais. Duas formas adaptativas de busca são propostas, uma global e outra local. Em ambas, o cálculo da função gulosa é modificado por uma quantidade que mensura heurísticamente a qualidade da solução parcial que é obtida quando uma decisão é tomada. A busca global é um procedimento adaptativo tratado como uma regra de aprendizado, uma vez que ele tenta aprender com os erros anteriores fazendo uma atualização da função gulosa a cada iteração da heurística.

Baita *et al.* (2000) abordam um caso prático do PPV comparando o desempenho do modelo tradicional de designação com duas novas heurísticas propostas para o problema. O modelo de designação é resolvido com o algoritmo húngaro descrito por Carpaneto *et al.* (1988). Neste modelo, a estrutura da função de custo considera os seguintes elementos: viagens mortas entre os terminais, mudanças de linhas e tempo de espera nos terminais. A primeira heurística proposta é baseada na técnica de Inteligência Artificial denominada Programação em Lógica. Uma vez que o PPV pode ser visto como um problema de otimização multiobjetivo (minimizar frota, tempo de viagens mortas e tempo de terminal), a heurística tenta satisfazer os diferentes critérios na ordem que segue. O algoritmo tenta conectar as viagens mais próximas que pertencem à mesma linha e ao mesmo terminal, satisfazendo os tempos mínimos e máximos de terminal. Se a primeira estratégia não cobrir todas as viagens, então a restrição de pertinência à mesma linha é retirada, e são conectadas viagens que partem do mesmo terminal. Se for necessário, são conectadas viagens que partem de diferentes terminais, iniciando a busca pelos terminais mais próximos. No algoritmo genético proposto, uma programação é representada por um gene, sendo que cada alelo representa um veículo, e sua posição representa a viagem atribuída ao veículo. Para manter a factibilidade da solução após uma operação de cruzamento, aplica-se um mecanismo específico

de reparação. A escolha dos indivíduos a serem reproduzidos na próxima geração é feita utilizando dois procedimentos não convencionais, além do método tradicional, segundo o qual a probabilidade de um indivíduo ser selecionado é proporcional ao valor da função objetivo.

Silva (2001) explora modelos de fluxo em redes para resolver o PPV. Duas técnicas de redução de rede são apresentadas. A primeira combina o modelo de fluxo com custo mínimo à técnica de redução de arcos, denominada Eliminação dos Arcos Longos. Apesar da redução significativa da rede, o método se mostra limitado na sua aplicabilidade. A segunda abordagem combina a técnica de Geração de Arcos Longos com o problema de pseudo-designação. A adaptação feita, denominada Geração de Arcos, considera, além da minimização do número de veículos, a redução do tempo de viagem morta e de terminal. O modelo não considera o tempo máximo de operação do veículo e se mostra pouco flexível para a inclusão de outras restrições operacionais, limitando sua aplicação.

Dada a complexidade do PPV no cenário atual e a possibilidade de incluir restrições operacionais de vários tipos visando integrá-lo à programação da tripulação, neste trabalho, o PPV é abordado por meio de um procedimento heurístico, no caso, a metaheurística *Iterated Local Search*, ILS, (Glover & Kochenberger, 2003). O método é combinado com um procedimento de busca local, o Método de Descida Randômico, para refinar as soluções perturbadas a cada iteração do ILS. Esse procedimento agiliza a busca local sem comprometer a qualidade dos ótimos locais encontrados.

3. Descrição do Problema Abordado

O Problema da Programação de Veículos (PPV) consiste em criar uma rotina diária de operação para uma frota de veículos disponibilizada para uso de uma empresa de modo que todas as viagens a ela responsabilizadas sejam realizadas. Para que a programação de atividades da frota seja factível é necessário atender a uma série de restrições, as quais podem variar conforme as políticas operacionais que regem o Sistema de Transporte Público no qual a empresa atua. Neste trabalho é tratado o problema envolvendo empresas da cidade de Belo Horizonte/Brasil.

A programação dos veículos é feita a partir de uma tabela de horários, na qual se encontram todas as viagens a serem realizadas em um determinado dia. A Tabela 1 ilustra um fragmento de uma tabela de horários. Nesta tabela, cada viagem é identificada pelas seguintes características: o número da viagem, horário de início da viagem (em minutos), o ponto inicial que corresponde ao local de início da viagem, horário de término da viagem (em minutos), o ponto final que se refere ao local em que a viagem termina, a linha a qual a viagem pertence, e finalmente as duas últimas colunas que se referem respectivamente aos tempos destinados ao embarque e desembarque de passageiros, dados em minutos, e que não necessariamente são iguais, sendo o primeiro dependente do horário de início da viagem e o segundo do horário de fim da mesma.

Tabela 1 – Tabela de Horários.

Número da Viagem	Horário de Início	Ponto Inicial	Horário de Término	Ponto Final	Linha	Tempo de Embarque	Tempo de Desembarque
Viag0	5	30	70	30	30.4	1	1
Viag1	5	302	38	302	302.4	1	1
Viag2	20	30	76	30	30.4	1	1
Viag3	240	302	265	302	302.4	1	1
Viag4	255	30	309	30	30.3	1	2
Viag5	260	302	281	302	302.1	1	1

Para exemplificar, considere a primeira viagem “Viag0” apresentada na Tabela 1. Essa viagem começa no ponto “30” aos 5 minutos do dia, ou seja, às 00:05 horas e é finalizada no ponto “30”

aos 70 minutos do dia, que corresponde às 01:10 horas. As duas últimas colunas indicam que o veículo que fará a “Viag0” permanecerá um minuto no terminal para embarque e após chegar ao destino mais um minuto de terminal deverá ser cumprido para permitir o desembarque.

Além do conjunto de viagens, devem ser fornecidas também informações relativas às distâncias entre os vários pontos de parada dos veículos, incluindo a garagem. De posse dessas informações é possível determinar a distância que um veículo percorre fora de operação, ou seja, sem passageiros para sair de um ponto e chegar a outro. A este tipo de deslocamento é dado o nome de viagem morta.

As restrições que foram levadas em consideração para geração da programação diária dos veículos neste trabalho, foram as seguintes: (a) Um veículo deve permanecer, no mínimo, trinta minutos consecutivos na garagem, por dia. Este tempo pode ser cumprido quando o veículo pára na garagem entre duas viagens consecutivas pertencentes à sua jornada diária ou mesmo depois de completada toda a sua jornada de trabalho; (b) Um mesmo veículo não pode realizar duas viagens simultaneamente, o que caracterizaria uma sobreposição; (c) Antes de se iniciar e após finalizar uma viagem, um veículo deve obrigatoriamente cumprir um tempo mínimo no terminal, que corresponde ao tempo necessário para o embarque e desembarque dos passageiros; (d) Um veículo não deve ficar duas horas ou mais em um terminal esperando para executar a sua próxima viagem. Se o tempo de espera exceder esse limite, o veículo deverá se deslocar para a garagem ou outro local onde possa esperar o horário de início da viagem subsequente. Diz-se, nesse último caso, que o veículo está fazendo dupla pegada; (e) Todas as viagens de responsabilidade de uma empresa, sem exceções, devem ser realizadas por algum veículo da frota.

Os objetivos do PPV considerados nesse trabalho são os seguintes: (i) Diminuir o tempo ocioso dos veículos, isto é, que os veículos não fiquem parados entre duas viagens consecutivas descontando-se os tempos de embarque e desembarque que são obrigatórios; (ii) Diminuir o número de deslocamentos fora de operação (deslocamentos que não equivalem a nenhuma viagem) e (iii) Diminuir o número de trocas de linha entre as viagens dos veículos da frota.

A solução para o Problema de Programação de Veículos consiste, portanto, em atribuir um conjunto de viagens para um número de veículos menor ou igual à frota disponível pela empresa, considerando as restrições apresentadas acima, de tal forma que todas as viagens sejam realizadas com o menor custo operacional possível. Cada conjunto de viagens a serem executadas por um veículo é chamado de “bloco do veículo”.

4. Metodologia proposta

4.1. Representação de uma solução

Uma solução s para o PPV consiste em um vetor de veículos, sendo que a cada veículo está associado um vetor de viagens a serem por ele executadas durante um dia de trabalho, ordenadas por seus horários de início.

4.2. Determinação da Solução Inicial

Uma solução inicial para o PPV é gerada por um procedimento construtivo guloso. Tal procedimento consiste na alocação, a cada passo, de uma nova viagem a um veículo da frota, de tal forma que esta combinação seja a que implique no menor custo possível de operação determinado pela aplicação da função de avaliação descrita na seção 4.3. O pseudocódigo do método considerado é descrito na Figura 2.

- 1) Seja T o conjunto de viagens a serem realizadas, com T ordenado crescentemente por horário de início de suas viagens.
- 2) Seja F o conjunto de veículos da empresa.
- 3) Enquanto ($T \neq \emptyset$) faça
 - a) t recebe a primeira viagem de T .
 - b) Faça $melhorDelta \leftarrow \infty$, onde $melhorDelta$ é a melhor variação ocorrida na função de avaliação ao se alocar a viagem t .
 - c) Para cada veículo $v \in F$ faça
 - i) $custoAnterior \leftarrow$ custo do veículo v antes de receber a viagem t .
 - ii) Aloque a viagem t ao veículo v .
 - iii) $custoPosterior \leftarrow$ custo do veículo v após inserir a viagem t em sua jornada de trabalho.
 - iv) Calcule a variação de custo Δ do veículo v após receber a viagem t ,
 $\Delta = custoPosterior - custoAnterior$
 - v) Se ($\Delta < melhorDelta$) então $melhorVeiculo \leftarrow info(v)$ e $melhorDelta \leftarrow \Delta$, isto é, armazene uma informação que permita identificar o veículo da frota que irá contribuir menos para o valor da função de avaliação ao receber a viagem t .
 - vi) Remova a viagem t do veículo v , ou seja, desfça a alocação realizada anteriormente para que possa ser analisada a atribuição da viagem t a outros veículos.
 - d) Aloque a viagem t ao veículo da frota F indicado por $melhorVeiculo$.
 - e) $fo \leftarrow fo + melhorDelta$, ou seja, atualize o valor da função de avaliação.
 - f) Remova t de T .
- 4) Retorne, como solução, a programação da frota F da empresa.

Figura 2 – Algoritmo Guloso para geração de uma solução inicial.

4.3. Função de Avaliação

Seja v um veículo pertencente à frota F de uma empresa, T o conjunto das suas n viagens diárias e i e j viagens consecutivas de sua jornada de trabalho. O custo $C(v)$ atribuído ao veículo v é calculado pela expressão (1):

$$C(v) = CD_{g1} + CD_{ng} + \sum_{(i,j) \in T} C_{ij} \quad (1)$$

Na equação (1), CD_{g1} e CD_{ng} são, respectivamente, os custos dos deslocamentos (isto é, das viagens mortas) do veículo v da garagem para o ponto de início de sua primeira viagem e do ponto final da sua última viagem para a garagem. Observa-se que não é atribuído um custo pelo uso do veículo; porém, esta função indiretamente contempla o fato de que o tamanho da frota é uma característica a ser otimizada quando penaliza os deslocamentos fora de operação que todo veículo deve realizar ao iniciar e finalizar a sua jornada diária de trabalho, ou seja, sempre que um novo veículo sai da garagem. C_{ij} é o custo associado à realização das viagens consecutivas i e j e é calculado conforme a expressão (2):

$$C_{ij} = \begin{cases} CD_{ij} + CS_{ij} & \text{se } t_{ij} < 0 \\ CT_{ij} + CD_{ij} & \text{se } 0 \leq t_{ij} \leq TempoMaxTerm \\ CD_{ig} + CD_{gj} & \text{se } t_{ij} > TempoMaxTerm \end{cases} \quad (2)$$

em que CT_{ij} , CD_{ij} e CS_{ij} correspondem, respectivamente, aos custos de permanência no terminal, deslocamento realizado fora de operação (viagem morta) e sobreposição entre as viagens i e j .

Na expressão (2), $TempoMaxTerm$ é o tempo máximo que um veículo pode permanecer no terminal, estipulado segundo políticas operacionais que regem o Sistema de Transporte Público, e t_{ij} corresponde ao período de tempo compreendido entre o fim da viagem i e o início da viagem j após descontado o tempo de viagem morta, de embarque e desembarque entre elas. Note que t_{ij} corresponde ao tempo real de ociosidade do veículo v entre as viagens i e j . Portanto, o valor t_{ij} poderá ser: (a) Negativo, indicando que há sobreposição de horário entre as viagens i e j . Assim, além da penalização associada à viagem morta realizada pelo veículo v entre as viagens i e j , há

um custo atribuído ao tempo de sobreposição entre elas; (b) Não-negativo e menor ou igual ao tempo máximo de terminal permitido para um veículo (*TempoMaxTerm*), indicando que o veículo v , após a realização da viagem i , ficará esperando no terminal pelo horário de início de sua próxima viagem (viagem j). Portanto, o custo de se realizar as viagens i e j consecutivamente é obtido penalizando-se o tempo de deslocamento do veículo fora de operação e o tempo que o mesmo permanece no terminal entre as viagens; (c) Maior que o tempo máximo de terminal permitido para um veículo (*TempoMaxTerm*), indicando que o veículo v terá que retornar à garagem entre as viagens i e j , ou seja, realizará uma dupla pegada; não havendo, dessa forma, a necessidade de espera no terminal. O custo de se realizar as viagens, nesse caso, é calculado penalizando-se os tempos de deslocamento do veículo fora de operação.

Na prática, é desejável não apenas obter uma solução factível para o PPV com tempos de terminal e de deslocamento baixos na soma global da programação, mas também que exista uma homogeneidade na distribuição desses tempos, isto é, deve ser evitado que haja paradas no terminal excessivamente demoradas ou que os tempos de deslocamentos de um veículo fora de operação sejam muito elevados. Para cumprir esse objetivo, atribui-se um custo individual para cada espera no terminal ou para cada viagem morta de um veículo na expressão (2), de acordo com uma função exponencial. Os custos CD_{ij} e CT_{ij} são obtidos conforme as equações (3) e (4), apresentadas a seguir:

$$CD_{ij} = \begin{cases} (\alpha_1 \times d_{ij})^{\beta_1} & \text{se } d_{ij} \leq \text{TempDesloc_Lim} \\ d_{ij} \times \text{peso_TempDesloc} & \text{se } d_{ij} > \text{TempDesloc_Lim} \end{cases} \quad (3)$$

$$CT_{ij} = \begin{cases} (\alpha_2 \times t_{ij})^{\beta_2} & \text{se } t_{ij} \leq \text{TempTerm_Lim} \\ t_{ij} \times \text{peso_TempTerm} & \text{se } t_{ij} > \text{TempTerm_Lim} \end{cases} \quad (4)$$

em que:

- t_{ij} é o tempo real de ociosidade do veículo v entre as viagens i e j ;
- d_{ij} é a duração da viagem morta do veículo v entre as viagens i e j ;
- *TempDesloc_Lim* é o tempo limite para a duração de uma viagem morta;
- *peso_TempDesloc* é a penalização, por unidade de tempo (em minutos), atribuída a uma viagem morta caso a sua duração supere o limite dado por *TempDesloc_Lim*;
- α_1 e $\beta_1 \in (0,1)$ são parâmetros utilizados para o cálculo da penalização atribuída a uma viagem morta caso a sua duração seja menor ou igual ao limite dado por *TempDesloc_Lim*;
- *TempTerm_Lim* é o tempo limite para a duração de uma parada no terminal;
- *peso_TempTerm* é a penalização, por unidade de tempo (em minutos), atribuída a uma espera no terminal que exceda o limite dado por *TempTerm_Lim*;
- α_2 e $\beta_2 \in (0,1)$ são parâmetros utilizados para o cálculo da penalização atribuída a uma espera no terminal caso a sua duração seja menor ou igual ao limite dado por *TempTerm_Lim*.

Observe que os custos CT_{ij} e CD_{ij} são dependentes da duração da espera no terminal ou da viagem morta efetuada pelo veículo v entre as viagens i e j . A um tempo de terminal ou de deslocamento aceitável é associado um custo mais baixo, calculado com base em uma função exponencial. Caso esse tempo esteja acima de um tempo limite, o custo é mais elevado e é calculado com base em uma função linear para diferenciar o nível de inviabilidade.

O motivo de se utilizar uma função exponencial está no fato de que ela estabelece uma variação suave do custo da espera no terminal (ou viagem morta) quando a duração é pequena e mais acentuada quando essa duração é maior. Assim, a tempos de terminal (ou viagem morta)

toleráveis estão associados custos mais próximos e para tempos quase intoleráveis, os custos são elevados.

Uma vez que a existência de sobreposição em uma solução para o PPV é uma inviabilidade, não se aplica para o cálculo de seu custo a mesma metodologia que a adotada anteriormente para penalizar o tempo de terminal e a viagem morta. Assim, o custo CS_{ij} é obtido de forma simples penalizando cada minuto de sobreposição pelo valor $peso_TempSobrep$, conforme equação (5):

$$CS_{ij} = - (t_{ij}) \times peso_TempSobrep \quad (5)$$

Note que, uma vez que o valor t_{ij} é sempre negativo quando há sobreposição entre as viagens i e j , é necessário multiplicá-lo por (-1) para que um custo positivo seja associado a esta sobreposição.

Considerando o exposto, uma solução s para o Problema de Programação de Veículos é avaliada com base na seguinte função f , dada pela expressão (6), a qual deve ser minimizada:

$$f(s) = tot_DPeg \times peso_DPeg + tot_TrocasLinha \times peso_TrocaLinha + \sum_{v \in F} C(v) \quad (6)$$

em que:

- tot_DPeg é a quantidade de duplas pegadas da programação que supera um certo número admissível, previamente definido e $peso_DPeg$ é a penalização correspondente;
- $tot_TrocasLinha$ é a quantidade de trocas de linha realizadas pelos veículos da frota e $peso_TrocaLinha$ é a penalização associada;
- $\sum_{v \in F} C(v)$ é a soma dos custos de todos os veículos pertencentes à frota F da empresa.

4.4. Estruturas de Vizinhança

Para exploração do espaço de soluções do PPV foram utilizados dois tipos de movimentos: realocação e troca. A realocação consiste em transferir uma viagem pertencente à jornada de trabalho de um dado veículo para a jornada de outro veículo. Já o movimento de troca corresponde a permutar uma viagem da jornada de um dado veículo com uma viagem pertencente a um outro veículo. Observa-se que o movimento de troca é de complexidade maior por envolver a manipulação de duas viagens ao passo que o movimento de realocação envolve somente uma.

4.5. ILS aplicado ao PPV

A metaheurística *ILS* (Glover & Kochenberger, 2003) foi adaptada para resolver o PPV, sendo seu pseudo-código apresentado na Figura 3.

Para refinar a solução perturbada, o procedimento de busca local utilizado é o Método Randômico de Descida (*MRD*), usado em duas versões distintas: uma que realiza apenas movimentos de troca (*DescidaRandomicaComTroca*), e outra que executa somente movimentos de realocação (*DescidaRandomicaComRealocacao*) durante a busca. O Método Randômico de Descida é utilizado porque ele é mais barato computacionalmente que o Método de Descida clássico. Há doze níveis de perturbação, que se diferenciam tanto quanto aos tipos de movimentos realizados quanto ao número de vezes que estes são aplicados. Para cada nível k ímpar, aplica-se $\lfloor k/2 \rfloor + 1$ movimentos de realocação. Para os níveis k pares, aplicam-se $k/2$ movimentos de troca.

- 1) Seja T o tempo de processamento corrente do algoritmo e T_{max} o tempo máximo admissível de processamento.
- 2) Seja $ciclos$ o número de ciclos realizados e $CiclosMax$ o número máximo de ciclos permitido.
- 3) Seja $nivel$ o nível de perturbação corrente.
- 4) $T \leftarrow 0$; $ciclos \leftarrow 0$; $nivel \leftarrow 1$;
- 5) Gere uma solução inicial s a partir do método guloso descrito na seção 4.2.
- 6) $s \leftarrow DescidaComRealocacao(s)$;
- 7) $s \leftarrow DescidaComTroca(s)$;
- 8) Faça $s^* \leftarrow s$, onde s^* é a melhor solução encontrada até então.
- 9) Enquanto $((ciclos < CiclosMax) \text{ e } (T < T_{max}))$ faça
 - a) $s' \leftarrow perturbacao(s^*, nivel)$;
 - b) $s' \leftarrow DescidaRandomicaComRealocacao(s')$;
 - c) $s' \leftarrow DescidaRandomicaComTroca(s')$;
 - d) Se $(f(s') < f(s^*))$ então
 - i) $s^* \leftarrow s'$, ou seja, atualize a melhor solução até então encontrada.
 - ii) $ciclos \leftarrow 0$; $nivel \leftarrow 1$;
 - e) Se $(f(s') = f(s^*))$ então
 - i) $s^* \leftarrow s'$, ou seja, atualize a melhor solução até então encontrada.
 - ii) $nivel \leftarrow nivel + 1$;
 - iii) Se $(nivel > numNiveis)$ então $ciclos \leftarrow ciclos + 1$ e $nivel \leftarrow 1$; onde $numNiveis$ é o número máximo de níveis de perturbação.
 - f) Se $(f(s') > f(s^*))$ então
 - i) $nivel \leftarrow nivel + 1$;
 - ii) Se $(nivel > numNiveis)$ então $ciclos \leftarrow ciclos + 1$ e $nivel \leftarrow 1$;
- 10) Retorne a solução s^* .

Figura 3 – Algoritmo ILS aplicado ao PPV.

5. Apresentação e Análise dos Resultados

O algoritmo ILS foi desenvolvido na linguagem C++ usando o compilador Borland C++ Builder 6.0 e testado em um microcomputador com processador Pentium IV, 2.4 GHz, com 512 MB de memória RAM, sob sistema operacional Windows XP Service Pack 2.

Para validar o método foram disponibilizadas pela empresa responsável pelo gerenciamento do Sistema de Transporte Público da cidade de Belo Horizonte, instâncias reais pertencentes a uma Bacia de Linhas de tal cidade. Foram fornecidas também, as programações de veículos adotadas atualmente pelas empresas que atuam na Bacia em estudo. É importante salientar que em tais instâncias os tempos de embarque e desembarque de todas as viagens são iguais à zero.

Os parâmetros utilizados no método ILS foram $T_{max} = 60$ minutos e $CiclosMax = tot_viagens \times tot_veiculos$, onde $tot_viagens$ e $tot_veiculos$ são, respectivamente, o total de viagens a serem realizadas pela empresa e o tamanho máximo da frota. Como estratégia de resolução do problema considerou-se que o tamanho da frota não deve exceder o total de veículos empregados pela programação feita manualmente por cada empresa e que o número máximo de duplas pegadas aceitáveis sem que haja qualquer penalização é igual a 60% do tamanho da frota disponível.

Nas tabelas 2 e 3 são apresentados os valores dos parâmetros utilizados para o cálculo da função de avaliação (6) apresentada na seção 4.3. Na Tabela 3 os valores $TempDesloc_Lim$ e $TempTerm_Lim$ são vinculados a cada instância e representam os maiores tempos de viagem morta e terminal, nessa ordem, em minutos, encontrados na solução gerada pela empresa. Isto foi feito para permitir uma comparação mais justa entre as soluções produzidas pela metodologia heurística proposta e pela empresa.

Tabela 2 – Parâmetros utilizados e seus respectivos valores.

Parâmetros	Valor	Parâmetros	Valor
<i>TempoMaxTerm</i>	119 minutos	<i>peso_TempDesloc</i>	50
α_1	2	<i>peso_TempTerm</i>	40
β_1	$\ln(16) / \ln(\alpha_1 \times \text{TempDesloc_Lim})$	<i>peso_TempSobrep</i>	80
α_2	1	<i>peso_DuplaPeg</i>	80
β_2	$\ln(16) / \ln(\alpha_2 \times \text{TempTerm_Lim})$	<i>peso_TrocaLinha</i>	5

No cálculo de β_1 e β_2 da Tabela 2, o valor 16 corresponde à penalidade máxima atribuída a uma viagem morta ou tempo de espera no terminal que obedece ao limite de tempo estabelecido por *TempDesloc_Lim* ou *TempTerm_Lim* apresentado na Tabela 3.

A Tabela 3 mostra também uma comparação entre as programações de veículos obtidas pelo método ILS e aquelas atualmente utilizadas pelas empresas em termos da função de avaliação (6) apresentada na seção 4.3. Foram analisadas quatro das empresas que operam na Bacia contemplada, considerando quatro dias da semana que apresentam quadros de horários distintos: segunda-feira, sexta-feira, sábado e domingo. Nesta tabela, considera-se que **FO** é o custo da programação realizada pela empresa e **Melhor FO**, **FO Médio**, **Desvio**, **PM** e **PMM** são, respectivamente, o menor custo encontrado pelo método considerado, a média dos custos em 10 execuções, o desvio dado por $(\text{FOMedio} - \text{MelhorFO})/\text{MelhorFO}$, o percentual de melhora de **FO** considerando o **Melhor FO** e o percentual de melhora do **FO** em relação ao **FO Médio**.

Tabela 3 – Comparação de Resultados (Empresa x ILS).

Instâncias	<i>Temp Desloc _Lim</i>	<i>Temp Term_ _Lim</i>	Solução Empresa	Solução ILS				
			FO	Melhor FO	FO Médio	Desvio (%)	PM (%)	PMM (%)
G02_SEG_260V	18	82	3450	2506	2509,8	0,15	27,36	27,25
G02_SEX_260V	18	77	3611	2552	2559,2	0,28	29,33	29,13
G02_SAB_172V	18	75	1921	1491	1491	0,00	22,38	22,38
G02_DOM_90V	18	83	1027	840	840,5	0,06	18,21	18,16
G46_SEG_505V	11	67	5429	4100	4114,7	0,36	24,48	24,21
G46_SEX_498V	14	59	4571	3572	3589,8	0,50	21,86	21,47
G46_SAB_372V	11	76	3874	2733	2747,9	0,55	29,45	29,07
G46_DOM_286V	14	70	2394	1900	1906,7	0,35	20,63	20,36
G48_SEG_468V	14	47	3605	3102	3133,4	1,01	13,95	13,08
G48_SEX_468V	14	67	3877	3446	3463,4	0,50	11,12	10,67
G48_SAB_359V	14	74	3098	2686	2709,5	0,87	13,30	12,54
G48_DOM_298V	14	45	2514	2285	2314,9	1,31	9,11	7,92
G69_SEG_639V	19	74	6564	4624	4662,4	0,83	29,56	28,97
G69_SEX_639V	19	50	6570	4718	4739,5	0,46	28,19	27,86
G69_SAB_441V	19	87	3935	2940	2958,3	0,62	25,29	24,82
G69_DOM_332V	19	70	2940	2328	2361,1	1,42	20,82	19,69
Médias						0,58	21,56	21,10

Com base nos resultados obtidos na Tabela 3, verifica-se que a heurística ILS supera todas as soluções das empresas, considerando tanto o valor médio quanto o melhor valor da função de avaliação. Além disso, o ILS proporcionou um percentual de melhora considerável da solução conforme pode ser visto analisando-se os valores dos percentuais de melhora **PM** e **PMM**, em que a média de melhora foi superior a 21% em ambos os casos. Adicionalmente, observa-se que os valores de desvio são baixos, todos inferiores a 1,5%, evidenciando a robustez do método.

A Tabela 4 mostra detalhadamente as características das melhores soluções geradas pela heurística ILS e de três empresas em diferentes dias da semana em relação ao número de veículos utilizados **NVEIC**, total de viagens a serem realizadas **NVIAG**, tempo de espera no terminal **TE**, tempo de viagem morta **TVM**, número de duplas pegadas **NDP** e função objetivo **FO**.

Tabela 4 – Resultado detalhado das empresas G46, G48 e G69.

Empresa	Característica	Solução Empresa				Solução ILS			
		SEG	SEX	SAB	DOM	SEG	SEX	SAB	DOM
G46	NVEIC	53	55	38	28	52	50	29	22
	NVIAG	505	498	372	286	505	498	372	286
	TE (hh:mm)	100:19	87:10	89:42	69:35	44:44	50:05	42:35	52:13
	TVM (hh:mm)	23:42	24:18	12:18	08:42	23:12	20:59	11:04	07:13
	NDP	26	26	3	1	22	18	5	1
	FO	5429	4571	3874	2394	4100	3572	2733	1900
G48	NVEIC	34	35	27	21	33	34	24	19
	NVIAG	468	468	359	298	468	468	359	298
	TE (hh:mm)	123:24	109:52	94:06	84:54	70:24	70:49	60:03	59:12
	TVM (hh:mm)	16:33	17:00	11:36	09:02	16:15	15:56	10:11	08:03
	NDP	12	13	6	5	8	6	2	1
	FO	3605	3877	3098	2514	3102	3446	2686	2285
G69	NVEIC	63	64	45	35	62	64	32	26
	NVIAG	639	639	441	332	639	639	441	332
	TE (hh:mm)	162:54	133:30	108:58	95:50	79:47	69:46	72:53	63:01
	TVM (hh:mm)	59:33	62:18	33:24	22:38	45:34	49:30	21:08	16:08
	NDP	40	45	13	6	13	17	2	0
	FO	6564	6570	3935	2940	4624	4718	2940	2328

Pela Tabela 4 verifica-se que para as empresas G46 e G48, as soluções otimizadas apresentaram redução significativamente no número de veículos utilizados, considerando tanto cada dia da semana individualmente ou simplesmente o dia com maior número de veículos utilizados. Para a empresa G46, por exemplo, na solução da empresa o número máximo de veículos utilizados foi 55 na sexta-feira, já com a heurística ILS foi 52 na segunda-feira. Por outro lado, para a empresa G69, apesar de ter havido uma redução do número de veículos na maioria dos dias, o total de veículos utilizados na sexta-feira foi mantido e este determinou também o número de veículos da frota. Assim, apesar de a solução ter sido melhorada, não houve redução da frota.

É importante salientar que a otimização do número de veículos utilizados por uma empresa pode ser avaliada de dois pontos de vista diferentes: É de interesse da empresa gestora reduzir somente o número máximo de veículos utilizados por uma empresa, isto é, minimizar a frota a ser mantida pela empresa de transporte. Porém, para as empresas responsáveis pelas viagens, qualquer redução do número de veículos operacionais em um dado dia, é interessante. Portanto, uma vez que a heurística ILS resolve a programação de cada dia da semana para cada empresa separadamente, é procurado sempre reduzir o número de veículos utilizados em cada dia, o que conseqüentemente acabará contemplando o desejo de se minimizar a frota.

6. Conclusões

Este trabalho apresenta uma metodologia heurística, baseada em *Iterated Local Search* (ILS), para resolver o Problema de Programação de Veículos (PPV). Para executar a busca local é utilizado o Método Randômico de Descida, o qual faz uso de movimentos de realocação e troca de viagens entre os veículos para explorar o espaço de soluções.

O método proposto foi aplicado a instâncias reais e se mostrou robusto e capaz de resolver eficientemente o PPV, produzindo soluções de melhor qualidade do que aquelas utilizadas pelas

empresas analisadas. Além da redução no número de veículos utilizados, que é um dos grandes contribuintes para a diminuição do custo da programação, a heurística ILS também proporcionou significativa redução do tempo de espera no terminal, tempo de viagem morta e número de duplas pegadas. O tempo de ociosidade do veículo no terminal foi reduzido a valores muito inferiores àqueles praticados pelas empresas. A maioria dos deslocamentos fora de operação foi reduzida em horas, o que representa uma grande economia em termos de combustível e tripulante. Por fim, a redução no número de duplas pegadas contribuiu tanto para diminuição do tempo de ociosidade do veículo quanto para a diminuição dos deslocamentos fora de operação.

A continuidade da pesquisa se dará com a introdução, no PPV, de algumas características da programação de tripulações (PPT), de modo a facilitar a resolução deste último problema. Além disso, a integração desses dois problemas é uma meta a ser alcançada.

Referências Bibliográficas

- Atkinson, J B (1998) A greedy randomised search heuristic for time-constrained vehicle scheduling and the incorporation of a learning strategy. *The Journal of the Operational Research Society*, v. 49, p. 700-708.
- Baita, F.; Pesenti, R.; Ukovich, W.; Favaretto, D. (2000) A comparison of different solution approaches to the vehicle scheduling problem in a practical case. *Computers and Operations Research*, v.27, p.1249-1269.
- Carpaneto, G.; Martello, S.; Toth, P. (1988) Algorithms and codes for the assignment problem. *Annals of Operations Research*, v.13, p. 193-223.
- Gavish, B.; Schweitzer, P.; Shlifer, E. (1978) Assigning buses to schedules in a metropolitan area. *Computers and Operations Research*, v.5, p. 129-138.
- Glover, F.; Kochenberger, G. (2003) *Handbook of Metaheuristics*. Kluwer Academic Publishers.
- Hoffstadt, J. (1981) Computerized vehicle and driver scheduling for the Hamburger Hochbahn Aktiengesellschaft. In: *Computer Scheduling of Public Transport*, Wren A. (ed.), North-Holland, Amsterdam, p. 35-52.
- Kirkman, F. (1968) Problems of innovation in the transport industry: a bus scheduling program. In: *Proceedings of PTRC Public Transport Analysis Seminar, Planning and Transport Research and Computation Co. Ltd.*, London, v.1, p. 1-15.
- Kwan, R. K.; Rahin, M. A. (1995) Bus scheduling with trip co-ordination and complex constraints. In: *Computer-Aided Transit Scheduling*, J. R. Daduna; I. M. Branco; J. M. P. Paixão (eds.), Springer-Verlag, Berlin, p. 91-101.
- Kwan, R. K.; Rahin, M. A. (1999) Object oriented bus vehicle scheduling - the BOOST system. In: *Computer-Aided Transit Scheduling*, N. H. M. Wilson (ed.), Springer-Verlag, Berlin, p. 177-191.
- Martin-Löf, A. (1970) A branch-and-bound algorithm for determining the minimal fleet size of a transportation system. *Transportation Science*, v.4, p.159-163.
- Psarras, J.; Stefanitsis, E.; Christodoulou N. (1997) Combination of local search and CLP in the vehicle-fleet scheduling problem. *European Journal of Operational Research*, v. 98, p. 512-521.
- Saha, J. L. (1970) An algorithm for bus scheduling problems. *Operational Research Quarterly*, v.21, p. 463-474.
- Silva, G. P. (2001) Uma metodologia baseada na técnica de geração de arcos para o problema de programação de veículos. Tese de doutorado. Escola Politécnica da USP, São Paulo.
- Wren, A.; Chamberlain, M. P. (1988) The development of Micro-BUSMAN: scheduling on micro-computers. In: *Computer-Aided Transit Scheduling*, J. R. Daduna; A. Wren (eds.), Springer-Verlag, Berlin, p.160-174.
- Wren, A. (1972) Bus scheduling: an interactive computer method. *Transportation Planning and Technology*, v.1, p.115-122.