

UM ALGORITMO HEURÍSTICO HÍBRIDO PARA RESOLUÇÃO DO PROBLEMA DE PROGRAMAÇÃO INTEGRADA DE VEÍCULOS E TRIPULAÇÕES

Marcone Jamilson Freitas Souza

Departamento de Computação, Programa de Pós-Graduação em Engenharia Mineral
Universidade Federal de Ouro Preto
35.400-000 Ouro Preto (MG)
marcone@iceb.ufop.br

Sabir Ribas, Igor Machado Coelho

Universidade Federal de Ouro Preto
35.400-000 Ouro Preto (MG)
{sibirribas, igor.machado}@yahoo.com.br

RESUMO

Este trabalho trata da programação integrada de veículos e tripulações. Dada a natureza combinatória do problema, a abordagem utilizada é heurística. O algoritmo proposto é baseado nos procedimentos *Iterated Local Search*, *Variable Neighborhood Descent* e Busca Tabu. Para explorar o espaço de busca são usados movimentos envolvendo realocações e trocas de viagens nos veículos e tarefas nas tripulações. Para testar o algoritmo foram utilizados dados reais da cidade de Belo Horizonte, Brasil. Os experimentos computacionais realizados mostraram que o algoritmo proposto foi capaz de produzir resultados superiores àqueles alcançados por um método clássico de solução dessas duas etapas do planejamento de transporte público.

PALAVRAS CHAVE. Programação integrada de veículos e tripulações, Transporte Público, Metaheurísticas.

Área principal: Logística e Transportes (LGT)

ABSTRACT

This work deals with the integrated vehicle and crew scheduling. Due to the combinatorial complexity of the problem, the approach to solve it is heuristic. A hybrid algorithm, based on *Iterated Local Search*, *Variable Neighborhood Descent* and *Tabu Search*, is proposed. In order to explore the solution space, some moves based on reallocation and swap of both, trips of the vehicles schedule and tasks of the duties schedule are used. The algorithm was tested with real data of the city of Belo Horizonte, Brazil. Computational results show that the proposed algorithm was able to produce better solutions than a traditional method applied to solve these two stages of the public transport planning.

KEYWORDS. Integrated Vehicle and Crew Scheduling, Public Transport, Metaheuristics.

Main area: Logistic and Transports

1. Introdução

Dada a complexidade do planejamento do transporte no Sistema de Transporte Público, ele é geralmente dividido em cinco etapas: planejamento das rotas, definição da tabela de horários, programação dos veículos, programação das tripulações e rodízio das tripulações. Normalmente, a saída de uma etapa corresponde à entrada da etapa subsequente. Neste trabalho, trata-se da resolução de duas delas, a programação de veículos e a de tripulações.

Segundo Freling *et al* (2003), existem três diferentes abordagens para resolver o problema de programação de veículos e de tripulações: Sequencial, Independente e Integrada.

Na abordagem sequencial os problemas de programação de veículos e tripulações são resolvidos separadamente, sendo a solução de um utilizada como entrada para a resolução do outro. A abordagem sequencial, por sua vez, é dividida em tradicional e inversa.

A abordagem sequencial tradicional foi a primeira a ser estudada e é a mais adotada por empresas de transporte. Nesta estratégia, é resolvido primeiramente o problema de programação de veículos (PPV) e, a partir dessa solução, é resolvido o problema de programação das tripulações (PPT). O inconveniente dessa abordagem é que a solução dos veículos guia a programação das tripulações (FRELING *et al.*, 2003), a qual tem o custo mais relevante entre os custos do sistema (BOUZADA, 2003). Uma alternativa para resolver esse problema é aplicar a abordagem tradicional, mas considerando algumas características dos tripulantes na programação dos veículos, deixando outras características para serem tratadas posteriormente, durante a resolução do PPT. Ainda que isto implique em acréscimos no custo da programação de veículos, essa alternativa facilita a resolução do PPT e geralmente produz soluções melhores que as obtidas pela abordagem tradicional (FRELING *et al.*, 2003). Scott (1985) e Darby-Dowman (1988), *apud* Freling *et al.* (2003), e Reis (2007) utilizam esta última estratégia.

Na abordagem sequencial inversa, o PPT é resolvido primeiro, diretamente da tabela de horários e, a seguir, é resolvido o PPV. Esta estratégia é justificada por dois motivos principais: 1) pelo fato de os custos com as tripulações serem mais elevados que os com os veículos e 2) o conjunto de restrições de veículos é muito menor que o de tripulações, o que sugere resolver o problema mais restrito primeiro, que tem menos alternativas de solução. Bassi *et al* (2007) propõem uma heurística baseada em GRASP (Feo e Resende, 1995) para resolver o problema de programação de tripulações, em que as tripulações são criadas a partir da tabela de horários, independentemente da solução dos veículos. Em um segundo momento, o problema de programação de veículos é resolvido tendo como entrada a solução das tripulações, definida anteriormente. Um mecanismo de Reconexão por Caminhos foi implementado para melhorar a solução. A heurística foi testada para várias empresas de transporte e os resultados comparados com aqueles obtidos pela abordagem tradicional. De acordo com os autores, o procedimento proposto é capaz de produzir soluções com custos menores que os obtidos pelo procedimento sequencial tradicional.

Na abordagem independente tanto a programação dos veículos quanto a dos tripulantes é feita diretamente a partir da tabela de horários. Geralmente, os resultados obtidos por essa abordagem não são viáveis na prática, pois alguns veículos operam sem um tripulante, e alguns tripulantes operam em veículos inexistentes, que não pertencem à programação dos veículos. Entretanto, essa abordagem permite determinar os limitantes inferiores para a solução ótima do problema de programação de veículos e tripulações.

Na abordagem integrada os dois problemas são resolvidos simultaneamente. Obviamente, a dificuldade de se resolver o problema de programação integrada de veículos e tripulações (PPVT) é maior por que o espaço de soluções é imenso. O PPVT contém todas as combinações possíveis de tripulações para cada uma das possíveis combinações de veículos. A natureza combinatória desse problema é ressaltada em Borndröfer *et al.* (2006) e em Freling *et al.* (2003). Entretanto, com o aumento na velocidade das CPUs, combinado com o progresso dos algoritmos, pesquisadores vêm dando mais ênfase a esse tópico recentemente. São exemplos de trabalhos utilizando esse tipo de abordagem: Freling *et al.* (1995), Haase e Friberg (1999) e Freling *et al.* (1999).

Segundo Freling *et al.* (2003), a abordagem integrada do PPVT sempre permitirá obter resultados melhores ou iguais aos resultados da abordagem seqüencial, e resultados piores ou iguais aos da abordagem independente.

Este trabalho apresenta um novo algoritmo para resolver o PPVT. O algoritmo combina os procedimentos heurísticos *Iterated Local Search*, *Variable Neighborhood Descent* e Busca Tabu com Relaxação Adaptativa. O uso de métodos heurísticos é justificado pela natureza combinatória do PPVT. A utilização das técnicas propostas é justificada tendo em vista os resultados obtidos para cada problema isolado (Vide Souza *et al.* (2007) para o problema de programação de veículos e Marinho *et al.* (2004) para a programação de tripulações).

O restante deste artigo é organizado como segue. Na Seção 2 descreve-se o problema tratado. O algoritmo proposto é apresentado na Seção 3. Os resultados computacionais são apresentados e analisados na Seção 4 enquanto a última seção conclui o artigo.

2. Definição do Problema

De acordo com Freling *et al.* (2003), o problema de programação veículos e tripulações pode ser definido como segue: dada uma tabela de horários de viagens a serem realizadas, encontrar a programação de custo mínimo para os veículos e para as tripulações tal que ambas sejam factíveis e mutuamente compatíveis. Cada viagem possui horários de início e fim e o tempo de viagem entre todos os pontos de ônibus são conhecidos. A programação dos veículos é factível se (1) cada viagem é atribuída a um único veículo, e (2) cada veículo possui uma seqüência factível de viagens, isto é, todo veículo deve ser capaz de executar consecutivamente as viagens a ele atribuídas. Determinadas viagens não pertencem a nenhuma linha e, assim, são denominadas viagens mortas. O tempo de viagem morta é o tempo em que um veículo viaja entre dois locais (terminais ou garagens) sem passageiros. Tal procedimento é necessário para reposicionar um veículo do ponto final de uma viagem ao ponto inicial da próxima viagem. As viagens atribuídas a um veículo definem sua jornada diária de trabalho, sendo essa jornada conhecida como bloco do veículo. Cada bloco é dividido em oportunidades de troca, definidos por um horário e um local, onde e quando pode haver uma troca de tripulação. Uma tarefa é definida por um conjunto de viagens entre duas oportunidades de troca consecutivas e representa a mínima porção de trabalho que pode ser associada a uma tripulação. O conjunto de tripulações deve executar todas as tarefas assim como o de veículos deve executar todas as viagens. As tarefas que são associadas a uma mesma tripulação definem sua jornada de trabalho. A programação das tripulações é factível se (1) cada tarefa é associada a um único tripulante, e (2) a seqüência de tarefas de cada um dos tripulantes pode ser realizada na prática por apenas um tripulante. Em particular, cada tripulante deve satisfazer certos quesitos. Um exemplo típico é a restrição de tempo máximo de trabalho. O custo de um tripulante é uma combinação de custos fixos e custos variáveis. A Seção 3.4 detalha os custos considerados neste trabalho.

As empresas analisadas neste trabalho estimam que em horários de pico, períodos em que há um número maior de passageiros em trânsito, o tempo necessário para a ocorrência de uma troca de tripulações é de dois minutos, e fora desses horários esse tempo é de um minuto. Segundo elas, existem dois horários de pico ao longo do dia: das 5:00 às 7:59 e das 16:00 às 18:59. Em função disso, são considerados dois tipos de jornada de trabalho: jornada simples e jornada com dupla pegada. Este último tipo de jornada existe em função da necessidade de um número maior de tripulações nos horários de pico. Assim, em geral, nas jornadas com pegada dupla, as tripulações trabalham durante o primeiro pico e retornam depois para atender viagens do segundo pico. Para caracterizar uma jornada com dupla pegada entre o final da primeira parte da jornada e o início da subsequente deve haver um intervalo de pelo menos duas horas.

Neste trabalho considera-se que os veículos estão sempre disponíveis ao longo do dia e que há apenas uma garagem.

3. Metodologia Proposta

3.1. Representação de uma solução

Uma solução para o PPVT consiste em um par (s_V, s_T) , em que s_V representa uma

solução para os veículos e s_T representa uma solução para as tripulações.

A solução s_V consiste em um conjunto de veículos, em que a cada um é associada uma lista de viagens a serem feitas ao longo do dia. A solução s_T consiste em um conjunto de jornadas de trabalho para as tripulações, em que a cada qual é associada uma lista de tarefas a serem feitas diariamente.

3.2. Construção de uma solução inicial

Uma solução inicial para o PPVT é feita primeiramente construindo uma solução inicial para os veículos e depois, a partir desta, é gerada uma solução para as tripulações. As duas soluções são construídas por procedimentos gulosos.

Para a construção de uma solução inicial do PPV é aplicado um procedimento que, a cada iteração, insere uma nova viagem a um veículo, de acordo a função de avaliação relativa aos veículos (veja a equação (2) da Seção 3.4). A escolha do veículo que receberá a viagem é totalmente gulosa, o que significa que o veículo que apresentar o menor custo relativo à alocação é escolhido. O procedimento é finalizado quando todas as viagens forem alocadas. O conjunto de viagens alocadas a cada veículo é chamado de bloco do veículo.

A partir do bloco de viagens de cada veículo é construída uma solução para o PPT. Inicialmente, cada veículo tem suas viagens agrupadas em tarefas. A seguir, as tarefas são ordenadas pelo horário de início. A cada passo há sempre uma jornada sem tarefas, além do conjunto de jornadas não vazias. O procedimento funciona como segue. A primeira tarefa é associada à jornada do primeiro tripulante. Depois, para cada tarefa da seqüência, é avaliada a inserção na jornada de cada tripulante do conjunto e mais na jornada do tripulante sem tarefas. A avaliação das jornadas é feita de acordo com a equação (8) da Seção 3.4, relativa à programação das tripulações. A tarefa é associada à jornada que apresentar menor custo de alocação. O procedimento pára quando todas as tarefas tiverem sido alocadas.

3.3. Estruturas de vizinhança

Para explorar o espaço de busca foram definidos seis tipos de movimentos. As vizinhanças N_V^R , N_V^S , N_C^R e N_C^S usam movimentos baseados em realocação (*Reallocation*) e troca (*Swap*), envolvendo viagens de veículos (*Vehicle*) e tarefas de tripulantes (*Crew*), respectivamente. As vizinhanças N_V^{RP} e N_V^{SP} consistem em realocação e troca de viagens em veículos, respectivamente, seguidas de uma reconstrução da programação das tripulações afetadas pelos movimentos de acordo com a seção anterior.

3.4. Função de avaliação

O custo de uma solução para o PPVT é calculado pela equação (1):

$$f(s) = \alpha \times f_V(s) + \beta \times f_T(s) \quad (1)$$

em que $f_V(s)$ representa a componente de f que avalia s quanto à programação dos veículos (equação 2) e $f_T(s)$ avalia s quanto à programação das tripulações (equação 8) e α e β são pesos. As funções $f_V(s)$ e $f_T(s)$ foram propostas em Souza *et al.* (2007) e Marinho *et al.* (2004), respectivamente, e modelaram satisfatoriamente problemas reais de Transporte Público.

$$f_V(s) = \sum_{k \in F} f_V^k(s) + \omega_{TL} \times numTrLinha + \omega_{EDP} \times excessoDuplaPegV \quad (2)$$

Na equação (2), $f_V^k(s)$ é o custo do veículo k , calculado pela equação (3); $numTrLinha$ é o número de trocas de linhas realizadas por todos os veículos e ω_{TL} é a penalidade associada; $excessoDuplaPegV$ representa o excesso de vezes em que os veículos, durante suas jornadas, retornam à garagem e permanecem lá por duas horas ou mais e ω_{EDP} é a penalidade para esse quesito. Por restrições operacionais, o número de duplas pegadas na programação dos veículos deve ser inferior a 60% do tamanho da frota.

$$f_V^k(s) = \sum_{(i,j) \in T} C_{ij} + CD_{g1} + CD_{ng} \quad (3)$$

Na equação (3), T é o conjunto das n viagens do veículo k , C_{ij} é o custo para realizar as viagens i e j consecutivas, calculado pela equação (4), CD_{g1} é o custo da viagem do veículo k da

garagem ao ponto de início da primeira viagem e CD_{ng} é o custo da viagem do veículo k do ponto final de sua última viagem à garagem.

$$C_{ij} = \begin{cases} CD_{ij} + CC_{ij} & \text{se } ocioso_{ij} < 0 \\ CT_{ij} + CD_{ij} & \text{se } 0 \leq ocioso_{ij} \leq TempoMaxTerm \\ CD_{ig} + CD_{gj} & \text{se } ocioso_{ij} > TempoMaxTerm \end{cases} \quad (4)$$

Na equação (4), CT_{ij} , CD_{ij} e CC_{ij} representam, respectivamente, o tempo em que o veículo permanece no terminal entre as viagens i e j , o tempo de viagem morta entre i e j , e o tempo em minutos em que a viagem i é conflitante com a viagem j . $TempoMaxTerm$ é o tempo máximo que um veículo pode permanecer no terminal, estabelecido por políticas operacionais do Sistema de Transporte Público e $ocioso_{ij}$ corresponde ao tempo entre o fim da viagem i e o início da viagem j menos o tempo de viagem morta, se for necessário o reposicionamento do veículo.

Nas equações (5) e (6), em que são determinados os custos CD_{ij} e CT_{ij} , tem-se que: tvm_{ij} é o tempo de viagem morta entre as viagens i e j ; $TempoDeslocLim$ é o tempo limite de viagem morta, w_{TD} é o peso se o tempo de viagem morta for maior que $TempoDeslocLim$, w_{TT} é o peso se o tempo de permanência no terminal ultrapassar o $TempoTermLim$, α_1 e β_1 são pesos, no intervalo $[0, 1]$, usados para ajustar as funções custo (5) e (6), respectivamente.

$$CD_{ij} = \begin{cases} (\alpha_1 \times tv_{m_{ij}})^{\beta_1} & \text{se } tv_{m_{ij}} \leq TempoDeslocLim \\ tv_{m_{ij}} \times w_{TD} & \text{se } tv_{m_{ij}} > TempoDeslocLim \end{cases} \quad (5)$$

$$CT_{ij} = \begin{cases} (\alpha_2 \times ocioso_{ij})^{\beta_2} & \text{se } ocioso_{ij} \leq TempoTermLim \\ ocioso_{ij} \times w_{TT} & \text{se } ocioso_{ij} > TempoTermLim \end{cases} \quad (6)$$

O conflito CC_{ij} entre duas viagens i e j é avaliado pela equação (7):

$$CC_{ij} = - (ocioso_{ij}) \times w_{TC} \quad (7)$$

em que w_{TC} é o peso associado a esse quesito.

Seja $R = \{r_1, r_2, \dots, r_{|R|}\}$ o conjunto de linhas a cargo da empresa e $S_1, S_2, \dots, S_{|S|}$ subconjuntos de R , chamados grupos de linhas, com $R = S_1 \cup \dots \cup S_{|S|}$ e $S_1 \cap S_2 \cap \dots \cap S_{|S|} = \emptyset$.

A avaliação das tripulações é feita pela equação (8) a seguir:

$$\begin{aligned} f_T(s) = & \sum_{i \in Trip} (penalidadeTempoOcioso_i + penalidadeHoraExtra_i + \\ & + \mu_3 \times numTrVPerm_i + \mu_4 \times numTrLPerm_i + \mu_5 \times numTrPPerm_i + \\ & + \lambda_1 \times tempoSobrepi + \lambda_2 \times tempoExcTrabi + \lambda_3 \times tempoInsufEJorni + \\ & + \lambda_4 \times numTrocaVPr oib + \lambda_5 \times numTrLPr oibi + \lambda_6 \times numTrPPr oibi) + \\ & + \lambda_7 \times excessoDuplaPeg + \delta \times numTripulantes \end{aligned} \quad (8)$$

em que:

- (i) $Trip$ é o conjunto de tripulantes da componente s_T ;
- (ii) μ_i e λ_i são as penalidades aplicadas às restrições e objetivos, respectivamente;
- (iii) $penalidadeTempoOcioso_i$ assume o valor μ_1 se ociosidade na jornada i for maior que duas horas e o valor $0.347 \times TempoOcioso_i^{0.8}$ caso contrário, onde $TempoOcioso_i$ corresponde ao tempo em que o tripulante fica ocioso ao executar a jornada i ;
- (iv) $penalidadeHoraExtra_i$ assume μ_2 se o tempo de hora extra for maior que duas horas, caso contrário assume valor $1.357 \times HoraExtra_i^{0.6}$, onde $HoraExtra_i$ indica o tempo, em minutos, em que o tripulante que executa a jornada i faz hora extra;
- (v) $numTrVeiculosPerm_i$ é o número de vezes em que ocorre troca de veículos permitida durante a jornada i . Uma troca de veículos é permitida quando há tempo para se efetuar a

- troca de tripulações;
- (vi) $numTrLPerm_i$ é o número de vezes em o tripulante associado à jornada i na rota r_j , é realocado a uma outra rota r_k , com as duas pertencentes ao mesmo grupo, isto é, $r_j \in S_m \subset R$ e $r_k \in S_m$;
 - (vii) $numTrPPerm_i$ indica o número de vezes em que um tripulante é realocado a um ponto (terminal) diferente durante a jornada i . Uma troca de terminal é permitida se: (1) os terminais envolvidos na troca pertencerem ao mesmo grupo terminais; (2) as tarefas pertencerem ao mesmo veículo; (3) houver mais que duas horas de ociosidade entre duas tarefas consecutivas da jornada i , isto é, em uma pegada dupla, a realocação de um tripulante entre diferentes terminais é permitida. Uma troca de ponto (terminal) é proibida caso não satisfaça a nenhum dos três quesitos anteriores;
 - (viii) $tempoSobrep_i$ é a quantidade de tempo, em minutos, em que duas tarefas são realizadas simultaneamente na jornada i , isto é, em duas tarefas consecutivas quando a segunda inicia antes do termino da primeira;
 - (ix) $tempoExcTrab_i$ representa o tempo, em minutos, da jornada i que ultrapassa nove horas de trabalho. O tempo trabalhado em cada jornada i é calculado como: $TempoTrab_i = (\text{horário de término da ultima tarefa da jornada } i) - (\text{horário inicial da primeira tarefa}) - (\text{duração do intervalo entre as tarefas que caracterizam pegada dupla, se a jornada possuir pegada dupla})$;
 - (x) $tempoInsufEJorn_i$ é a quantidade de tempo, em minutos, que falta para que a diferença entre o horário final da jornada i em um dia e o horário inicial no dia subsequente atinja o tempo mínimo entre jornadas de 11 (onze) horas. Esse valor é calculado como: $tempoInsufEJorn_i = \max \{0, 11 \times 60 - (1440 - \text{horário de término da jornada } i + \text{horário inicial da jornada } i)\}$;
 - (xi) $numTrVProib_i$ é o número de vezes em que ocorreu troca proibida de veículos durante a jornada i . Uma troca de veículos é proibida quando não há tempo para se efetuar a troca;
 - (xii) $numTrLProib_i$ é o número de vezes em que ocorreu troca proibida de linhas durante a jornada i . Uma troca de linha é proibida se as linhas envolvidas pertencerem a grupos diferentes, isto é, se dadas duas linhas r_j e r_k tivermos $r_j \in S_m \subset R$ e $r_k \notin S_m$;
 - (xiii) $numTrPProib_i$ é o número de vezes em que ocorreu troca proibida de pontos (terminais) durante a jornada i (veja o item (vii));
 - (xiv) $excessoDuplaPeg$ representa o número de jornadas classificadas como de pegada dupla que excede $1,25 \times$ (tamanho da frota de veículos). Esse valor é uma estimativa da empresa, que assume que no máximo 50% dos tripulantes podem fazer pegadas duplas;
 - (xv) $numTripulantes$ é o número de tripulantes escalados para executar o conjunto de jornadas e δ é a penalização correspondente.

Nos itens (iii) e (iv) são usadas funções cujos valores estão no intervalo $[0, 16]$ e $[0, 24]$, respectivamente. Nestas funções, de domínio $[0, 120]$, quanto maior for o tempo de ociosidade ou hora-extra, mais acentuada é a penalidade. Elas procuram, também, levar em consideração o fato de que a hora-extra é 50% mais cara que a hora de trabalho normal. Maiores detalhes sobre a relação dos pesos utilizados podem ser vistos em Marinho (2005).

3.5. Uma nova heurística integrada de refinamento

Foram concebidos dois procedimentos de busca local para refinar soluções do PPVT. O primeiro, chamado de Descida Randômica de Viagens em Veículos com Propagação (DRVVP), tem por objetivo melhorar estritamente a componente s_V , relativa à programação dos veículos, mas mantendo-se coerente a componente s_T , relativa à programação das tripulações. Para isso, exploram-se as vizinhanças Realocação de Viagens de Veículos (N_V^R) e Realocação Propagada de Viagens de Veículos (N_V^{RP}). Nessa versão obtém-se, a cada iteração, um vizinho qualquer da solução corrente por meio de um movimento m de acordo com a vizinhança N_V^R . Caso o vizinho v'_1 gerado proporcione melhora no valor da função de avaliação, a nova solução corrente será o vizinho v'_2 obtido por meio de um movimento que possua as mesmas características de m .

(veículos e viagens; tripulantes e tarefas), mas agora, de acordo com a vizinhança N_V^{RP} ; caso contrário, um novo vizinho é gerado. Esse procedimento se repete até que se exceda *DescidaMax* iterações sem melhora na solução. Para verificar se um vizinho proporciona melhora, esse método considera apenas o custo referente à programação dos veículos.

O segundo procedimento é a Descida Randômica de Viagens em Veículos Integrada (DRVVI). O objetivo, nesse caso, é refinar uma solução s utilizando-se vizinhanças que explorem a integração dos dois problemas, no caso, N_V^{RP} , que faz a realocação propagada de viagens em veículos e N_V^{SP} , que faz a troca propagada de viagens em veículos. Essas são estruturas de vizinhança que exigem mais processamento que as demais e, por isso, a navegação no espaço de busca é geralmente mais lenta. As duas estruturas consistem em uma alteração simples, realocação ou troca, na programação de veículos e em uma adição gulosa de tarefas à programação de tripulações. A adição gulosa é a parte mais custosa do processo, por isso deve restringir-se a momentos propícios. No caso, apenas para as soluções pouco “distantes” da solução corrente, com respeito ao custo da programação dos veículos, são aplicados os movimentos integrados. Mais especificamente, nesta busca local gera-se, a cada iteração, um vizinho qualquer da solução corrente por meio de um movimento m de acordo com a vizinhança integrada escolhida. Se a estrutura escolhida for a N_V^{RP} , o movimento m será obtido pela vizinhança N_V^R , já se a estrutura escolhida for a N_V^{SP} , o movimento m será obtido pela vizinhança N_V^S . Caso o vizinho v'_1 gerado possua custo $f_V(v'_1)$ inferior ao valor obtido pela multiplicação de uma constante ρ ($\rho > 1$) pelo custo $f_V(v)$ dos veículos na solução corrente v , um novo vizinho v'_2 é gerado por meio de um movimento m' pertencente à vizinhança escolhida e que utiliza os mesmos veículos e viagens do movimento m . Se o custo global $f(v'_2)$, relativo a veículos e tripulações do vizinho v'_2 , for inferior ao custo global da solução corrente $f(v)$, a nova solução corrente será v'_2 . Assim como no método DRVVP, esse procedimento é repetido até que se exceda *DescidaMax* iterações sem melhora na solução global.

3.6. O algoritmo ILS_VND_BTRA

Descreve-se, a seguir, o algoritmo ILS-VND-BTRA para resolver o PPVT. Este algoritmo combina características dos procedimentos *Iterated Local Search* (ILS), *Variable Neighborhood Descent* (VND) e Busca Tabu com Relaxação Adaptativa (BTRA).

Para fundamentá-lo, seja N^k ($k=1, \dots, k_{max}$) um conjunto finito de estruturas de vizinhança pré-selecionadas, e $N^k(s)$ o conjunto de soluções da k -ésima vizinhança de s . Chama-se uma solução s' de ótimo local com respeito a $N^k(s)$, se não existe uma solução $s \in N^k(s')$ tal que $f(s) < f(s')$. O procedimento heurístico VND (Hansen e Mladenovic, 2003) é baseado em três princípios básicos: (1) um ótimo local para uma vizinhança pode não necessariamente ser para outra; (2) um ótimo global é um ótimo local para todas as possíveis vizinhanças; e (3) em vários problemas, ótimos locais de uma vizinhança são relativamente próximos dos ótimos de outras.

O procedimento VND_2N, proposto para fazer uma busca local no espaço de soluções do PPVT, está descrito na Figura 1. Ele inspira-se na idéia da heurística VND no sentido de que não somente as vizinhanças são alteradas, mas também os procedimentos de busca local usados. No passo 1 aplica-se a uma solução s_0 o procedimento DRVVP utilizando a vizinhança N_V^{RP} resultando uma solução s_1 . Como DRVVP é um procedimento de descida, a solução s_1 ou é melhor que a solução s_0 ou é a própria solução s_0 . Assim, a solução s_1 é atribuída à solução s_0 . O algoritmo continua com a chamada ao procedimento DRVVP usando a vizinhança N_V^{SP} , gerando a solução s_2 . Se o custo da solução s_2 for inferior ao custo de s_0 , então se inicia novamente a execução do passo 1. O Passo 1 termina quando os procedimentos DRVVP não produzirem melhoria nas soluções. As execuções dos Passos 2 e 3 são semelhantes à do Passo 1, sendo que no Passo 2 o refinamento de uma solução é feito por meio dos procedimentos BTRA_R e BTRA_S e, no Passo 3, o refinamento é feito pelo procedimento DRVVI usando a vizinhança N_V^{RP} e depois a N_V^{SP} . Os procedimentos BTRA_R e BTRA_S são baseados na metaheurística Busca Tabu (Glover, 1997) com oscilação estratégica (ou Relaxação Adaptativa) usando as vizinhanças N_C^R e N_C^S , respectivamente. Estes procedimentos consistem em mudar periodicamente o valor dos pesos durante a busca, promovendo a geração de inviabilidades para

explorar regiões ainda não visitadas do espaço de soluções. A descrição deste procedimento, especializado para resolver o PPT, está descrito em Marinho *et al.* (2004).

Proc.	VND_2N
Início	Seja s_0 uma solução inicial
Passo 1	$s_1 \leftarrow \text{DRVVP}(s_0, N_V^{RP})$ $s_2 \leftarrow \text{DRVVP}(s_1, N_V^{SP})$ Se s_2 for melhor que s_0 , então faça $s_0 \leftarrow s_2$ e volte ao início do Passo 1
Passo 2	$s_3 \leftarrow \text{BTRA_R}(s_2)$ $s_4 \leftarrow \text{BTRA_S}(s_3)$ Se s_4 for melhor que s_2 , então faça $s_2 \leftarrow s_4$ e volte ao início do Passo 2
Passo 3	$s_5 \leftarrow \text{DRVVI}(s_4, N_V^{RP})$ $s_6 \leftarrow \text{DRVVI}(s_5, N_V^{SP})$ Se s_6 for melhor que s_4 , faça $s_4 \leftarrow s_6$ e volte ao início do Passo 3 Se houve alguma melhora no Passo 3, volte ao Passo 2
Passo 4	Retorne s_6

Figura 1 – Procedimento VND_2N aplicado ao PPVT

Iterated Local Search – ILS (Lourenço *et al.* 2003) é uma metaheurística com quatro componentes básicas: “Geração de solução inicial”, “Busca Local”, “Perturbação” e “Critério de aceitação”. O primeiro módulo tem por objetivo gerar uma boa solução inicial para o problema. “Busca local” é o módulo que retorna um ótimo local para a busca. Já “Perturbação” consiste em realizar modificações nos ótimos locais s de forma a gerar uma nova solução s' próxima de s . O módulo “Critério de aceitação” consiste em determinar se o resultado s'' da aplicação da busca local à solução s' vai ser aceita ou não como nova solução corrente. Neste trabalho, considera-se que a solução s'' é aceita, e escreve-se $s \leftarrow s''$, se ela for melhor que o ótimo local anterior, isto é, se $f(s'') < f(s)$. As perturbações no PPVT consistem em realizar movimentos aleatórios na vizinhança N_V^{RP} . O número de movimentos aleatórios é mensurado pela variável *nível*, de forma que quanto maior esse número, maior é a “distância” da solução perturbada ao ótimo local corrente. Uma perturbação de nível i consiste em realizar $i + 2$ movimentos aleatórios. O número de modificações no ótimo local, isto é, o nível de perturbação, é aumentado sempre que as buscas locais não forem bem sucedidas em ILS_{max} vezes. O algoritmo é encerrado quando um número máximo de modificações, dado por $NivelMax$, ou um tempo limite é alcançado. Na Figura 2 apresenta-se o pseudocódigo do algoritmo ILS-VND-BTRA proposto para resolver o PPVT.

Algoritmo ILS-VND-BTRA

```

1:  Entrada:  $s, f(\cdot), NivelMax, ILS_{max}, TempoLimite$ 
2:   $s^* \leftarrow s; \quad nível \leftarrow 0;$ 
3:  enquanto  $nível < NivelMax$  e  $tempo\_de\_processamento < TempoLimite$  faça
4:     $iter \leftarrow 0;$ 
5:    enquanto  $iter < ILS_{max}$  e  $tempo\_de\_processamento < TempoLimite$  faça
6:       $s \leftarrow \text{Perturbação}(s, nível);$ 
7:       $s \leftarrow \text{VND\_2N}(s);$ 
8:      se  $f(s) < f(s^*)$  então
9:         $s^* \leftarrow s; \quad nível \leftarrow 0; \quad iter \leftarrow 0;$ 
10:     senão
11:        $s \leftarrow s^*; \quad iter \leftarrow iter + 1;$ 
12:     fim se
13:   fim enquanto
14:    $nível \leftarrow nível + 1;$ 
15: fim enquanto
16: retorne  $s^*$ 

```

Figura 2 – Algoritmo ILS-VND-BTRA aplicado ao PPVT

4. Resultados computacionais

O algoritmo descrito foi implementado na linguagem C++ usando o ambiente de desenvolvimento e compilador Borland C++ Builder 6.0 e testado em um microcomputador com processador Intel® Pentium®, 3.00 GHz, com 1 GB de memória RAM e sistema operacional Windows XP Professional.

Para validar o algoritmo foram disponibilizadas pela BHTRANS, empresa responsável pelo gerenciamento do Sistema de Transporte Público da cidade de Belo Horizonte, problemas-teste reais pertencentes a uma Bacia de Linhas de tal cidade.

Os parâmetros do método ILS_VND_BTRA foram empiricamente ajustados em uma bateria preliminar de testes. O número de iterações sem melhora do ILS (*ILSmax*) foi fixado em 50, enquanto *BTmax* (número de iterações sem melhora no método BTRA) foi fixado em 50. O valor escolhido para *NivelMax* foi 5 e o número de iterações sem melhora (*DescidaMax*) nas chamadas dos procedimentos DRVVP e DRVVI foi dado por *número de viagens* \times *tamanho máximo da frota de veículos*. O valor do parâmetro ρ do procedimento DRVVI foi fixado em 1,2. O número máximo de duplas pegadas na programação dos veículos foi fixado em 60% do tamanho da frota referente à solução produzida manualmente pelas empresas analisadas. Já o número máximo de duplas pegadas na programação das tripulações foi fixado em 1,25 vezes o tamanho da frota. Esta última relação foi adotada considerando-se que, na prática, a cada veículo estão associados 2,5 tripulantes e que não é desejável se ter mais que 50% das tripulações fazendo dupla pegada.

A primeira fase da abordagem sequencial tradicional usa o algoritmo ILS descrito em Souza *et al.* (2007) e foi interrompido com 15 minutos de processamento. Já a segunda fase usa o algoritmo BTRA descrito em Marinho *et al.* (2004). O tempo total de processamento foi limitado em meia hora. Portanto, na metodologia sequencial tradicional foi destinado 15 minutos para resolver o problema de programação de veículos e 15 minutos para resolver o problema de programação de tripulações.

Nas tabelas 1 e 2 são apresentados os valores dos parâmetros utilizados para o cálculo da função de avaliação apresentada na seção 5.3. Na Tabela 2 os valores *TempoDeslocLim* e *TempoTermLim* são vinculados a cada problema-teste e representam os maiores tempos de viagem morta e terminal, nessa ordem, em minutos, encontrados nas soluções utilizadas por duas empresas do Sistema. Tais parâmetros são os mesmos utilizados pelos algoritmos que resolvem o PPV e o PPT de forma sequencial. Os pesos $\alpha = 1$ e $\beta = 2,5$ relativos à programação de veículos e tripulações, respectivamente, foram utilizados com o objetivo de se dar uma importância maior à programação das tripulações.

Tabela 1 - Parâmetros utilizados na função de avaliação do PPVT

Parâmetro	Valor	Parâmetro	Valor
<i>TempoMaxTerm</i>	119 minutos	μ_1	18
α_1	2	μ_2	40
β_1	$\ln(16) / \ln(\alpha_1 \times \text{TempoDeslocLim})$	μ_3	19
α_2	1	μ_4	5
β_2	$\ln(16) / \ln(\alpha_2 \times \text{TempoTermLim})$	μ_5	5
ω_{TD}	50	λ_1	40
ω_{TT}	40	λ_2	40
ω_{TC}	80	λ_3	40
ω_{EDP}	80	λ_4	45
ω_{TL}	5	λ_5	35
δ	100	λ_6	35
α	1	λ_7	45
β	2.5		

A Tabela 3 contém os resultados encontrados, relativos à função de avaliação, para as

duas metodologias de solução do PPVT em 10 execuções para cada problema teste. Pode-se observar que há redução tanto nos melhores valores quanto nos valores médios da função de avaliação em todos os casos analisados. O desvio do algoritmo proposto é pequeno, provando sua robustez, pois há pouca variabilidade nos valores das soluções finais. As melhores soluções produzidas pelo algoritmo proposto têm valores da função de avaliação até 6,43% melhores que aqueles relativos à metodologia sequencial tradicional.

Tabela 2 – Problemas-teste e seus respectivos tempos limites relativos ao PPV

Probl.-teste	TempoDeslocLim	TempoTermLim	Probl.-teste	TempoDeslocLim	TempoTermLim
G02_DOM	18	73	G27_DOM	23	110
G02_SEG	18	118	G27_SEG	23	47
G02_SEX	18	113	G27_SEX	23	42
G02_SAB	18	62	G27_SAB	23	57

Tabela 3 - Comparação de resultados (Sequencial Tradicional × Integrada)

Problema-teste	Método Sequencial Tradicional			Método Integrado			
	Melhor Valor	Valor Médio	Desvio	Melhor Valor	Valor Médio	Desvio	Melhora
G02_DOM	7036	7160,35	6,47%	6725,5	6841,5	1,72%	4,41%
G02_SEG	22507,5	22828,55	8,26%	21087,5	21297,7	1,00%	6,31%
G02_SEX	23045,5	23354,1	7,50%	21724,5	21910,1	0,85%	5,73%
G02_SAB	12938,5	13260,5	8,51%	12220	12468,05	2,03%	5,55%
G27_DOM	3497,5	3550,75	4,51%	3397,5	3442,3	1,32%	2,86%
G27_SEG	8299	8463,25	7,30%	7887,5	7992,55	1,33%	4,96%
G27_SEX	8429,5	8486,25	7,59%	7887,5	8015,55	1,62%	6,43%
G27_SAB	5193,5	5346,65	5,73%	5057	5176,35	2,36%	2,63%

As tabelas 4 e 5 comparam, para cada metodologia, algumas características das melhores soluções encontradas em quatro dias da semana das empresas G02 e G27.

Tabela 4 – Características das soluções de G02: Método Seq. Tradicional × Método Integrado

Empresa G02	Método Sequencial Tradicional				Método Integrado			
	DOM	SEG	SEX	SAB	DOM	SEG	SEX	SAB
Número de viagens	90	260	260	172	90	260	260	172
Número de veículos	11	40	40	23	11	40	39	23
Tempo de terminal (hh:mm)	16:13	22:26	5:56	21:02	13:44	8:12	22:14	21:02
Tempo de viagem morta (hh:mm)	8:24	12:50	12:14	16:48	9:50	15:06	14:58	18:36
Número de tripulantes	21	62	63	37	19	62	63	34
Tempo ocioso (hh:mm)	15:13	15:04	9:23	15:44	11:07	7:52	6:28	15:12
Tempo de hora-extra (hh:mm)	5:04	6:44	7:34	20:08	14:58	8:28	12:31	19:44
Trocas de linhas	1	5	7	1	4	4	6	2
Trocas de pontos	2	18	19	9	3	11	8	10
Trocas de veículos	6	62	66	11	2	34	34	15
f_v	841	2510	2553	1491	893	2740	2887	1585
f_T	2478	7999	8197	4579	2333	7339	7535	4254
$f = f_v + 2.5 \times f_T$	7036	22507.5	23045.5	12938.5	6725.5	21087.5	21724.5	12220

Apesar de a função de avaliação dos veículos não visar a redução do número de veículos, pelo menos diretamente, percebe-se pelas Tabelas 4 e 5 que o método integrado produz soluções com número igual ou menor de veículos que o método tradicional. Comparando outras características das soluções produzidas, observam-se as seguintes vantagens do método integrado sobre o método sequencial tradicional: redução no número de tripulações (de 21 para 19 aos domingos e de 37 para 34 aos sábados na empresa G02, de 24 para 23 às segundas e de 25 para 23 às sextas na empresa G27), redução no tempo de ociosidade das tripulações, redução no tempo de espera em terminal (à exceção das sextas-feiras nas duas empresas). O aumento no número de horas-extras é recompensado pela melhoria em alguma outra característica da solução como a

diminuição do número de tripulações ou do número de trocas de veículos. Por exemplo, na empresa G02 há às sextas um aumento na quantidade de horas-extras de quase 5 horas; no entanto, há uma redução significativa no número de trocas de veículos (de 66 para 34). Já na empresa G27, há às sextas um aumento de mais de 7 horas-extras; no entanto, este aumento é acompanhado de uma redução de duas tripulações, bem como de 7 trocas de veículos (o que representa mais da metade do número de trocas de veículos da solução produzida pelo método tradicional). A despeito do aumento no tempo de espera em terminal às sextas-feiras nas duas empresas, valem as mesmas observações anteriores, acrescidas ao fato de que, para a empresa G02, este aumento foi acompanhado da redução de um veículo.

Tabela 5 – Características das soluções de G27: Método Seq. Tradicional × Método Integrado

Empresa G27	Método Seq. Tradicional				Método Integrado			
	DOM	SEG	SEX	SAB	DOM	SEG	SEX	SAB
Número de viagens	52	98	98	69	52	98	98	69
Número de veículos	5	14	13	7	5	14	13	7
Tempo de terminal (hh:mm)	12:57	10:25	8:45	11:12	10:47	10:22	11:08	10:54
Tempo de viagem morta (hh:mm)	3:50	16:06	15:20	8:26	4:36	16:52	15:20	9:12
Número de tripulantes	10	24	25	15	10	23	23	14
Tempo ocioso (hh:mm)	6:30	10:52	13:44	10:14	5:06	10:06	8:57	8:54
Tempo de hora-extra (hh:mm)	6:53	5:58	6:29	7:33	6:35	11:38	14:42	13:19
Trocas de linhas	0	0	0	0	0	0	0	0
Trocas de pontos	0	3	2	1	0	3	3	0
Trocas de veículos	4	14	12	4	2	6	5	3
f_v	370	1019	952	661	385	1050	1005	682
f_T	1251	2912	2991	1813	1205	2735	2753	1750
$f = f_v + 2.5 \times f_T$	3497.5	8299	8429.5	5193.5	3397.5	7887.5	7887.5	5057

Pelas tabelas 4 e 5 verifica-se, também, que o algoritmo proposto foi capaz de produzir soluções melhores, sob o ponto de vista da função de avaliação, para todos os dias considerados. Observa-se que, quanto à integração, há uma piora no custo dos veículos e uma redução significativa no custo dos tripulantes, produzindo assim soluções melhores para o conjunto. A aplicação da metodologia desenvolvida, com os pesos adotados neste trabalho, é vantajosa em situações nas quais, para a empresa, os custos envolvidos são maiores com as tripulações.

5. Conclusões

Este artigo apresentou um algoritmo, baseado nas heurísticas *Iterated Local Search*, *Variable Neighborhood Descent* e Busca Tabu com Relaxação Adaptativa, para resolver o problema de programação integrada de veículos e tripulantes. Os resultados obtidos pelo algoritmo proposto foram comparados com aqueles obtidos por um método tradicional sequencial, que resolve primeiramente o problema de programação de veículos e depois o problema de programação de tripulantes. Para testá-lo foram utilizados dados reais de empresas do Sistema de Transporte Público da cidade de Belo Horizonte (MG).

Os experimentos computacionais realizados mostraram que o algoritmo proposto foi capaz de produzir resultados superiores àqueles alcançados pela clássica metodologia sequencial. Foram produzidas soluções piores para os veículos, mas melhores para as tripulações, gerando, no entanto, soluções melhores para o conjunto. A melhora global proporcionada pelo algoritmo foi de até 6,43%, o que mostra que é promissor explorar a interação desses problemas. Adicionalmente, a variabilidade das soluções finais foi relativamente baixa com relação ao valor da função de avaliação, no caso, inferior a 2,36%. Como trabalho futuro, faz-se necessário fazer uma análise de sensibilidade do algoritmo com respeito à adoção de diferentes pesos para a função de avaliação.

Agradecimentos

Os autores agradecem à FAPEMIG (processo TEC 679/06), ao CNPq (processo 474831/2007-8) e à BHTRANS pelo apoio ao desenvolvimento deste trabalho.

Referências

- Bassi, H. V., Silva, G. P., Marinho, E. H., Souza, M. J. F.** (2007), GRASP and Path-relinking Applied to the Independent Crew Scheduling Problem, In: International Conference on Computers in Urban Planning and Urban Management, 2007, Foz do Iguaçu, Reviewed papers in the Conference Proceedings of CUPUM 2007, v.1, p.1 – 11.
- Borndröfer, R.; Grötschel, M. and Pfetsch, M.** (2006), Public transport to the fORe!, *ORMS Today*, v. 33(2), p. 30-40.
- Bouzada, C. F.** (2003), *Custo do Transporte Coletivo por Ônibus*, Editora C/Arte, Belo Horizonte.
- Darby-Dowman, K., Jachnik, J. K.; Lewis, R. L. e Mitra, G.** (1988), Integrated decision support systems for urban transport scheduling: Discussion of implementation and experience, in J. R. Daduna e A. Wren (eds.), *Computer-Aided Transit Scheduling: Proceedings of the Fourth International Workshop*, Springer Verlag, Berlin, p. 226-239.
- Feo, T.A., Resende, M.G.C.** (1995), Greedy randomized adaptive search procedures, *Journal of Global Optimization*, v. 6, p. 109-133.
- Freling, R. C.; Boender, G. E. e Paixão, J. M. P.** (1995), An integrated approach to vehicle and crew scheduling, Technical Report 9503/A, Econometric Institute, Erasmus University Rotterdam, Rotterdam.
- Freling, R., Huisman, D., Wagelmans, A.** (2003), Models and Algorithms for Integration of Vehicle and Crew Scheduling, *Journal of Scheduling*, v. 6(1), p. 63–85.
- Freling, R.; Wagelmans, A. P. M. e Paixão, J. M. P.** (1999), An overview of models and techniques for integrating vehicle and crew scheduling, in N. H. M. Wilson (ed.), *Computer-Aided Transit Scheduling*, Springer Verlag, Berlin, p. 441-460.
- Garey, M. R., Johnson, D. S.** (1979), Computers and Intractability, In W. H. Freeman (ed.), *A Guide to the Theory of NP-Completeness*, New York.
- Glover, F., Laguna, M.** (1997), *Tabu Search*, Kluwer Academic Publishers.
- Haase, K. e Friberg, C.** (1999), An exact branch and cut algorithm for the vehicle and crew scheduling problem, in N. H. M. Wilson (ed.), *Computer-Aided Transit Scheduling*, Springer Verlag, Berlin, p. 63-80.
- Hansen, P., Mladenovic, N.** (2003), Variable neighborhood search: Principles and applications, *European Journal of Operations Research*, v. 130, p. 449-467.
- Lourenço, H. R.; Martin, O.; Stützle, T.** (2003), Iterated Local Search, In F. Glover and G. Kochenberger (ed.), *Handbook of Metaheuristics*, Kluwer Academic Publishers.
- Marinho, E. H.** (2005), Heurísticas Busca Tabu para o Problema de Programação de Tripulações de Ônibus Urbano, Dissertação de mestrado, Niterói: Universidade Federal Fluminense, 89 p.
- Marinho, E. H., Ochi, L. S., Drummond, L. M. A., Souza, M. J. F., Silva, G. P.** (2004), Busca Tabu aplicada ao Problema de Programação de Tripulações de Ônibus Urbano, Simpósio Brasileiro de Pesquisa Operacional, 2004, São João Del Rey, Anais do XXXVI SBPO. Rio de Janeiro: SOBRAPO, v.1. p.1471 – 1482.
- Reis, J. v. A.** (2007), Programação integrada de veículos e tripulações de ônibus urbano, Exame de qualificação de mestrado, Programa de pós-graduação em Engenharia de Transportes, São Paulo: Escola Politécnica da Universidade de São Paulo.
- Scott, D.** (1985), A large linear programming approach to the public transport scheduling and cost model, in J. M. Rousseau (ed.), *Computer Scheduling of Public Transport 2*, North Holland, Amsterdam, p. 473-491.
- Souza, M. J. F., Silva, G. P., Simões, E. M. L.** (2007), Programação de ônibus urbano: uma abordagem heurística In: *Transporte em transformação XI: trabalhos vencedores do prêmio CNT de Produção Acadêmica 2006*. Brasília: Positiva, 2007, v.1, p. 39-57.