

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Engenharia de Produção

**Pesquisa Operacional Aplicada no Problema de
Planejamento do Transporte Público**

Douglas Baroni Rizzato

TCC-EP-26-2011

Maringá - Paraná
Brasil

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Engenharia de Produção

**Pesquisa Operacional Aplicada no Problema de
Planejamento do Transporte Público**

Douglas Baroni Rizzato

TCC-EP-26-2011

Trabalho de Conclusão de Curso apresentado ao Curso de
Engenharia de Produção, do Centro de Tecnologia, da
Universidade Estadual de Maringá

Orientador: Prof. Dr. Ademir Aparecido Constantino

**Maringá - Paraná
2011**

DEDICATÓRIA

A Deus, por mais uma conquista.

Aos meus pais Sidney e Rosemarie e minha irmã
Gabriela, pelo apoio incondicional.

“A persistência é o menor caminho do êxito”. (Charles Chaplin)

AGRADECIMENTOS

A Deus, por iluminar meu caminho e ter me guiado até aqui.

A toda minha família, em especial, aos meus pais Sidney e Rosemarie e minha irmã Gabriela por serem minha inspiração e pela paciência, apoio e incentivo em todos os momentos.

A minha namorada Aline e aos meus velhos amigos por estarem sempre comigo nos momentos difíceis e nos momentos de alegria.

Aos meus amigos de sala, pelos momentos em que superamos dificuldades, em que compartilhamos alegria, pela troca de experiências, e, principalmente, por toda colaboração e ajuda mútua.

Ao meu orientador Ademir, que com muita sabedoria me conduziu na realização deste e de outros trabalhos.

Aos professores de Engenharia de Produção por compartilharem seus conhecimentos.

A Universidade Estadual de Maringá.

A todos que de alguma forma fizeram parte desta caminhada.

RESUMO

Utilizar os recursos disponíveis de forma eficiente é um grande desafio enfrentado pelas organizações. Dentre os diversos problemas que envolvem a melhor forma de alocar recursos, este trabalho está focado no problema de planejamento operacional do transporte público, mais especificamente no problema de escalonamento dos motoristas, que juntamente com os veículos representam os principais recursos do sistema de transporte público. O problema de escalonamento de motoristas é classificado como NP-Difícil (Leone *et al.*, 2010) e consiste em formar jornadas diárias de trabalho para os motoristas, através do sequenciamento das viagens, sujeito a uma série de restrições, provenientes da legislação trabalhista, de acordos com o sindicato dos motoristas e de políticas da empresa de transporte.

Diante disso, neste trabalho é investigado um algoritmo heurístico para resolver tal problema buscando minimizar os custos da solução. O algoritmo trabalha em duas fases: construção da solução inicial e melhoramento, sendo que esta fase pode ser resolvida considerando duas abordagens diferentes. Cada abordagem considera diferentes configurações dos algoritmos, com variações dos procedimentos de melhoria e na função de custo da solução inicial. Para validar o algoritmo foram realizados testes computacionais para as diferentes versões, utilizando as mesmas instâncias utilizadas em Calvi (2005). Em geral, os resultados alcançados obtiveram custos menores que os utilizados para comparação, evidenciando os benefícios das técnicas apresentadas.

Palavras-chave: Problema de Escalonamento de Tripulações. Problema de Escalonamento de Motoristas. Otimização Combinatória. Problema de Atribuição.

ABSTRACT

Deal with the available resources efficiently is a major challenge faced by organizations. Among the many problems that deal with the best way to allocate resources, this work focuses on the public transport operational planning problem, more specifically on the bus drivers scheduling problem (BDSP), which with the vehicles represents the main resources of public transport system. The BDSP is known to be NP-Hard (Leone *et al.*, 2010) and consists in generating daily work schedules for the drivers, through the sequencing of the trips, subject to a number restriction, from labor laws, agreements with the drivers union and the policies of the company.

Therefore, this paper investigates a heuristic algorithm to solve this problem aiming to minimize the total cost of the solution. The algorithm works in two phases: construction of the initial solution and improvement phase, which is solved considering two different approaches. For both approaches are considered different configurations of the algorithms, as improvement procedures variations and the change of the cost function for the initial solution. To validate the algorithm, computational tests are performed for the different versions, using the same instances used in Calvi (2005). In general, the results achieved lower costs than those used for comparison, showing the benefits of the presented techniques.

Keywords: Crew Scheduling Problem. Bus Drivers Scheduling Problem. Combinatorial Optimization. Assignment Problem.

SUMÁRIO

LISTA DE ILUSTRAÇÕES.....	ix
LISTA DE TABELAS.....	x
LISTA DE ABREVIATURA E SIGLAS.....	xi
1 INTRODUÇÃO	1
1.1 JUSTIFICATIVA	3
1.2 DEFINIÇÃO E DELIMITAÇÃO DO PROBLEMA	3
1.3 OBJETIVOS	4
1.3.1 <i>Objetivo geral</i>	4
1.3.2 <i>Objetivos específicos</i>	4
1.4 METODOLOGIA	4
2 REVISÃO DA LITERATURA.....	6
2.1 PROBLEMA DE ESCALONAMENTO DE PESSOAL.....	6
2.2 PROBLEMA DE ESCALONAMENTO DE MOTORISTAS	9
2.2.1 <i>Métodos de resolução</i>	10
2.3 PROBLEMA DE ATRIBUIÇÃO	18
2.4 GRAFOS.....	19
2.5 ALGORITMOS HEURÍSTICOS E META-HEURÍSTICAS	20
3 DESENVOLVIMENTO.....	22
3.1 CARACTERIZAÇÃO DO PROBLEMA	22
3.2 ALGORITMOS PROPOSTOS	24
3.2.1 <i>Considerações Iniciais</i>	24
3.2.2 <i>Construção da solução inicial</i>	25
3.2.3 <i>Fase de Melhoramento</i>	28
3.2.4 <i>Descrição geral dos algoritmos</i>	34
4 RESULTADOS	38
4.1 CONSIDERAÇÕES INICIAIS.....	38
4.2 IMPLEMENTAÇÃO	38
4.3 RESULTADOS OBTIDOS	38
4.4 COMPARAÇÃO ENTRE OS PROCEDIMENTOS DE MELHORAMENTO.....	45
4.5 RELAÇÃO ENTRE O NÚMERO DE VIAGENS E O TEMPO DE EXECUÇÃO	46
5 CONSIDERAÇÕES FINAIS.....	47
REFERÊNCIAS	49

LISTA DE ILUSTRAÇÕES

FIGURA 1: O PROCESSO DE PLANEJAMENTO DE TRANSPORTES PÚBLICOS. ADAPTADO DE: PRATA, 2010.	2
FIGURA 2: (A) GRAFO DIRECIONADO (B) GRAFO NÃO DIRECIONADO. ADAPTADO DE: ZIVIANI, 2004.....	20
FIGURA 3: ESTRUTURA DA MATRIZ DE CUSTOS NA CONSTRUÇÃO DA SOLUÇÃO INICIAL.....	26
FIGURA 4: PASSOS PARA A CONSTRUÇÃO DA SOLUÇÃO INICIAL	27
FIGURA 5: EXEMPLO DE SOLUÇÃO INICIAL	28
FIGURA 6: ESTRUTURA DA MATRIZ DE CUSTOS NA EXECUÇÃO DO PCR	29
FIGURA 7: EXEMPLO DE POSSÍVEIS RECOMBINAÇÕES DO PCR AO RESOLVER A CAMADA 3	29
FIGURA 8: EXEMPLO DE RECOMBINAÇÃO APÓS EXECUÇÃO DO PCR NA CAMADA 3	30
FIGURA 9: PASSOS PARA A EXECUÇÃO DO PCR.....	30
FIGURA 10: ESTRUTURA DA MATRIZ DE CUSTOS NA EXECUÇÃO DO K-SWAP	31
FIGURA 11: EXEMPLO DE POSSÍVEIS RECOMBINAÇÕES DO 1-SWAP AO RESOLVER CAMADA 4	32
FIGURA 12: EXEMPLO DE RECOMBINAÇÃO APÓS EXECUÇÃO DO 1-SWAP NA CAMADA 4	33
FIGURA 13: PASSOS PARA A EXECUÇÃO DO PROCEDIMENTO K-SWAP	33
FIGURA 14: APRESENTAÇÃO DA SOLUÇÃO MELHORADA	34
FIGURA 15: DESCRIÇÃO RESUMIDA DO ALGORITMO PROPOSTO CONSIDERANDO A TROCA DE UM PROCEDIMENTO QUANDO ESTE NÃO REALIZAR UMA MELHORIA NA SOLUÇÃO.....	35
FIGURA 16: DESCRIÇÃO RESUMIDA DO ALGORITMO PROPOSTO CONSIDERANDO A TROCA DE PROCEDIMENTOS A CADA ITERAÇÃO	36
FIGURA 17: MODELO GERAL DE NOMENCLATURA UTILIZADA PARA AS DIFERENTES VARIAÇÕES DO ALGORITMO..	37
FIGURA 18: DIFERENÇA ENTRE CUSTOS X TEMPO DE PROCESSAMENTO.....	41
FIGURA 19: GRÁFICO DE CUSTO X ITERAÇÕES PARA O MÉTODO K(D)P_F1	42
FIGURA 20: GRÁFICO DE CUSTO X ITERAÇÕES PARA O MÉTODO K(D)P_F2	43
FIGURA 21: GRÁFICO DE CUSTO X ITERAÇÕES PARA O MÉTODO K(D)P-VND_F2	43
FIGURA 22: REDUÇÃO PERCENTUAL CONSIDERANDO OS DIFERENTES PROCEDIMENTOS DE MANEIRA ISOLADA COMPARADOS COM A MELHOR SOLUÇÃO OBTIDA	45
FIGURA 23: TEMPO DE PROCESSAMENTO X NÚMERO DE VIAGENS	46

LISTA DE TABELAS

TABELA 1: QUADRO-RESUMO DOS TRABALHOS CITADOS.....	17
TABELA 2: INSTÂNCIAS UTILIZADAS	22
TABELA 3: DURAÇÃO DA HORA DE TRABALHO (EM MIN.) E CUSTO (EM MIN. PAGOS) POR PERÍODO DO DIA	24
TABELA 4: RESTRIÇÕES DAS JORNADAS DE TRABALHO	24
TABELA 5: RESUMO DAS VARIAÇÕES DOS ALGORITMOS CONSIDERANDO A FUNÇÃO F1 NA RESOLUÇÃO INICIAL ...	37
TABELA 6: RESUMO DAS VARIAÇÕES DOS ALGORITMOS CONSIDERANDO A FUNÇÃO F2 NA RESOLUÇÃO INICIAL ...	37
TABELA 7: RESULTADOS ENCONTRADOS CONSIDERANDO A FUNÇÃO F1 NA RESOLUÇÃO INICIAL	39
TABELA 8: RESULTADOS ENCONTRADOS CONSIDERANDO A FUNÇÃO F2 NA RESOLUÇÃO INICIAL	40
TABELA 9: COMPARAÇÃO DOS MELHORES RESULTADOS COM OS RESULTADOS DE CALVI (2005)	41
TABELA 10: TEMPO DE EXECUÇÃO QUANDO UM CUSTO MENOR QUE O DE CALVI (2005) É ENCONTRADO.....	44

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
ATP	Algoritmo de Treinamento Populacional
BT	Busca Tabu
BVGP	Busca em Vizinhança de Grande Porte
CLT	Consolidação das Leis do Trabalho
GAMS	<i>General Algebraic Modelling System</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
ILS	<i>Iterated Local Search</i>
MD-VCSP	<i>Multiple-Depot Integrated Vehicle and Crew Scheduling</i>
PA	Problema de Atribuição
PCC	Problema de Cobertura de Conjuntos
PCR	Procedimento de Cortes e Recombinações
PEM	Problema de Escalonamento de Motoristas
PET	Problema de Escalonamento de Tripulação
PL	Programação Linear
PLI	Programação Linear Inteira
PO	Pesquisa Operacional
PPC	Problema de Particionamento de conjuntos
PPT	Problema de Programação da Tripulação
SA	<i>Simulated Annealing</i>
VND	<i>Variable Neighborhood Descent</i>
VNS	<i>Variable Neighborhood Search</i>

1 INTRODUÇÃO

Utilizar os recursos disponíveis de forma eficiente é um problema com o qual o homem lida desde as primeiras civilizações e que nos dias de hoje tem importância fundamental na produtividade e desempenho das organizações. E é diante da necessidade de alocar recursos de forma eficiente, que durante a Segunda Guerra Mundial surge a Pesquisa Operacional (PO), uma ciência multidisciplinar voltada para a resolução de problemas e auxílio à tomada de decisões (HILLIER e LIEBERMAN, 2010).

Após o fim da guerra, o sucesso da PO no empreendimento bélico despertou o interesse de sua aplicação fora do ambiente militar (HILLIER e LIEBERMAN, 2010). O grande crescimento industrial pós-guerra e a complexidade dos problemas enfrentados pelas organizações, aliados a melhoria das técnicas da PO e o desenvolvimento de computadores fizeram com que a PO fosse disseminada para as empresas e outras organizações, alcançando resultados fantásticos e contribuindo significativamente para aumentar a eficiência e a produtividade destas organizações (HILLIER e LIEBERMAN, 2010).

Dentre os diversos problemas que envolvem a melhor forma de alocar recursos, este trabalho está focado no problema de planejamento operacional do transporte público. O problema consiste de um conjunto de atividades desenvolvidas pelas empresas gestoras visando organizar a operação do sistema de transporte coletivo, de forma a alocar da melhor maneira os recursos necessários ao atendimento da demanda e realizar os serviços com maior eficiência e eficácia possível (SILVA e CUNHA, 2010).

As primeiras etapas do planejamento estão mais relacionadas a pesquisas de demanda, para definir as rotas que serão atendidas, as linhas e a frequência com que os veículos passarão por elas e os pontos de parada. Após essa fase de levantamento da demanda, deve ser montada a tabela de horários, que transformará as rotas, linhas e frequência em viagens (PRATA, 2010). Geralmente, essas etapas iniciais são feitas pelo órgão gestor de transportes do município. A partir daí, a empresa deve realizar o escalonamento dos veículos que cobrirão as viagens, o escalonamento dos motoristas para os veículos e o escalonamento mensal/semanal dos motoristas com base na escala diária (PRATA, 2010). Na Figura 1 pode ser visualizado o esquema do processo de planejamento de transportes públicos.

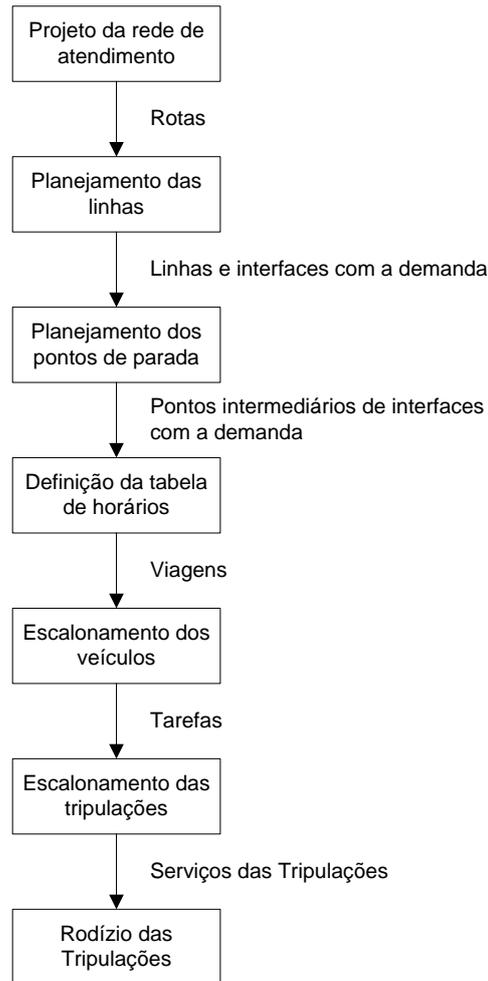


Figura 1: O processo de Planejamento de Transportes Públicos. Adaptado de: Prata, 2010.

Durante o planejamento, as companhias de transporte público precisam lidar com importantes desafios, agravados com o crescimento da população, exigência de um serviço de qualidade e pressão do governo para melhor utilização dos recursos. Segundo Lourenço *et al.* (2001), devido à alocação correta dos recursos ser um processo bastante complicado, há uma necessidade crescente de ferramentas computacionais que auxiliem na tomada de decisão.

Sabe-se que os veículos e os motoristas são os principais recursos do sistema de transporte público, e a forma como são alocados tem impacto direto na qualidade e no custo do serviço (SILVA e CUNHA, 2010). Dessa forma, otimizar o uso desses recursos significa importante redução nos custos da empresa, com consequências para os usuários, refletidas no preço da passagem (CALVI, 2005).

Além da redução de custos direta, há o benefício proporcionado aos funcionários, que poderiam cumprir jornadas menos exaustivas, aumentando a qualidade do serviço. Além

disso, com jornadas de trabalho dentro das leis, evitam-se possíveis custos passivos (futuros) devido a ações trabalhistas (CALVI, 2005).

Neste trabalho é abordado o Problema de Escalonamento de Motoristas (PEM), também conhecido como Problema de Escalonamento de Tripulação (PET) ou Problema de Programação da Tripulação (PPT), pelo fato dos custos com pessoal representarem a maior parte dos custos operacionais nas empresas de transporte coletivo (CALVI, 2005; SANTOS, 2008).

1.1 Justificativa

Devido ao grande número de combinações possíveis para a resolução do PEM e das diversas restrições que devem ser consideradas, o problema é de difícil solução e pertence à classe NP-Difícil (Leone *et al.*, 2010), ou seja, não pode ser resolvido por métodos exatos em tempo polinomial, exigindo assim o uso de heurísticas. Essa dificuldade fomenta a pesquisa de novas técnicas computacionais, que por sua vez contribuem para maior conhecimento do problema e avanço das técnicas computacionais, que podem ser empregadas a outros problemas. Em vista disso, o presente trabalho dá continuidade ao trabalho de Calvi (2005), focando no emprego de novas técnicas de forma a ampliar a investigação científica.

Do ponto de vista prático, o trabalho se justifica por abordar a resolução de um problema real considerando instâncias de dados reais. Sendo que a otimização desse processo pode representar um impacto significativo no desempenho da empresa, com escalas dentro da lei, redução de custos operacionais e maior satisfação dos colaboradores (CALVI, 2005).

1.2 Definição e delimitação do problema

O PEM consiste em determinar a melhor programação diária dos condutores de forma a obedecer às restrições, minimizar custos operacionais e cumprir a escala de veículos para a tabela de horários pré-determinada. A importância de uma programação eficiente dos condutores está relacionada ao impacto desta nos custos totais de uma empresa de transporte coletivo, já que veículos e tripulantes são os componentes que constituem a maior parte dos custos da empresa (SILVA *et al.*, 2002; PRATA, 2010).

Outro aspecto importante do PEM é que ele apresenta algumas características específicas, que dependem das restrições regionais e das regras adotadas pelas empresas, o que faz com que modelos desenvolvidos para um determinado caso possam não servir para outro (SILVA *et al.*, 2002).

Dessa forma, o enfoque do problema será dado na resolução do PEM pela utilização de um algoritmo heurístico, baseado na resolução de consecutivos problemas de atribuição, considerando as características de um problema regional.

1.3 Objetivos

1.3.1 Objetivo geral

Investigar um algoritmo heurístico para otimizar a solução do problema de escalonamento de motoristas de ônibus.

1.3.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Investigar o problema de escalonamento de motoristas em transporte coletivo;
- Desenvolver um algoritmo heurístico que resolva o problema de escalonamento de motoristas;
- Estender a adaptar para este problema os algoritmos heurísticos propostos por Rizzato *et al.* (2010) para a resolução do problema de escalonamento de enfermeiros;

1.4 Metodologia

O presente trabalho pode ser classificado como uma pesquisa:

- Aplicada, em relação à sua natureza, pois se dirige a solução de um problema específico de aplicação prática (SILVA e MENEZES, 2005);
- Explicativa, do ponto de vista de seus objetivos, por se tratar de uma pesquisa experimental quanto aos procedimentos técnicos;

- Experimental, em relação aos procedimentos técnicos, pois é realizada de forma a analisar os efeitos da utilização de um algoritmo na resolução do problema, quando comparado com outros métodos encontrados para resolução.

A realização do trabalho foi estruturada da seguinte forma:

1. Revisão bibliográfica acerca do problema de planejamento do transporte coletivo;
2. Definição da estratégia de resolução e implementação;
3. Implementação do algoritmo;
4. Experimentação computacional;
5. Análise dos resultados.

2 REVISÃO DA LITERATURA

2.1 Problema de Escalonamento de Pessoal

O escalonamento de atividades, um problema clássico da Pesquisa Operacional também conhecido na literatura como *scheduling*, é um processo de tomada de decisão enfrentado por diversos setores, como nas atividades de produção de bens e na prestação de serviços, possuindo vasta gama de aplicações.

De acordo com Baptiste (2001), esse tipo de problema é muito difícil de ser resolvido, sendo que a maioria das aplicações possui complexidade NP-Difícil e, por causa disso, para encontrar boas soluções que respeitem todas as restrições envolvidas no problema é preciso explorar um grande espaço de busca, o que requer o uso de técnicas computacionais mais apuradas, que realizem a busca de modo eficiente.

Segundo Pinedo (2008), um problema de escalonamento lida com a alocação de recursos para determinadas atividades, durante certo período de tempo, buscando otimizar um ou mais objetivos. Esse tipo de problema abrange ampla área de aplicação, sendo que os recursos e tarefas assumem diferentes formas. Como exemplifica Pinedo (2008), os recursos podem ser máquinas de uma indústria, pista de um aeroporto, trabalhadores em um canteiro de obra, unidades de processamento em um ambiente de computação, entre outros. Enquanto as tarefas poderiam ser operações de um processo produtivo, decolagens e aterrissagens programadas no aeroporto, fases de um projeto de construção civil, execução de programas de computador, entre outros.

Dentro dos problemas de escalonamento, uma área que tem se tornado cada vez mais importante para as organizações, devido aos enormes benefícios que podem ser alcançados como redução de custos e maior satisfação dos funcionários é o escalonamento de pessoal (ERNST *et al.*, 2004).

A relevância do problema de escalonamento de pessoal deve-se ao fato de ser extremamente difícil encontrar boas soluções para os problemas, que são altamente restritivos, complexos e, geralmente, buscam soluções que minimizem custos, atendam às preferências dos

empregados, distribuam equitativamente as tarefas entre os empregados e satisfaçam todas as restrições de trabalho (ERNST *et al.*, 2004).

Nas escalas de pessoal, um aspecto importante que tem sido estudado em alguns trabalhos é levar em consideração as preferências individuais dos funcionários na elaboração das escalas, no que diz respeito aos horários e turnos de trabalho. Diante de um problema de escalas de teleatendentes, Constantino *et al.* (2006), além de considerar as restrições operacionais do problema, também consideram as restrições pessoais dos trabalhadores quanto aos horários de trabalho, de tal modo que seja possível aos funcionários frequentarem os cursos desejados no horário de folga e que os trabalhadores residentes em outras cidades não façam parte da escala de sobreaviso.

Ernst *et al.* (2004) apresenta uma descrição dos principais problemas relacionados com o escalonamento de pessoal em diferentes áreas de aplicação, brevemente citados a seguir:

- Sistemas de transporte: consiste do escalonamento das tripulações nos diversos serviços de transporte, como: companhias aéreas, companhias ferroviárias e companhias de transporte de massas, como ônibus e metrô.
- Centrais de teleatendimento: envolve o escalonamento de pessoal em diferentes horários de trabalho, de forma a cobrir a demanda dos postos de atendimento e atender à disponibilidade dos colaboradores.
- Sistemas de saúde: nesse contexto, a grande maioria das aplicações concentra-se no problema de escalonamento de enfermeiros nos hospitais levando em consideração a necessidade de disponibilizar pessoal suficiente em todos os horários, de acordo com o histórico de demanda e ainda lidar com as diferenças de especialidade e contrato dos enfermeiros.
- Serviços de proteção e emergência: o problema lida com a alocação dos funcionários de forma a cumprir padrões de serviço, disponibilizando pessoal especializado de acordo com a demanda, sazonalidade e especialidade do funcionário.
- Serviços à comunidade e utilitários: envolve a escala de pessoas nos diversos serviços prestados à comunidade, como escalas para uma central de processamento postal e no serviço militar.
- Gestão do local: consiste no planejamento e escalonamento das diversas atividades que ocorrem em um determinado local, como escalonamento do pessoal de

manutenção em um aeroporto, assim como do pessoal responsável pelo carregamento das bagagens.

- Serviços financeiros: um exemplo consiste na escala de pessoal para auditorias, na qual devem ser observadas as habilidades dos auditores e as localidades a serem auditadas.
- Turismo e Hotelaria: área em que, geralmente, o pessoal contratado possui diferentes habilidades, podendo trabalhar em mais de uma atividade, de forma que o problema consiste na melhor alocação desses funcionários às atividades baseada em uma previsão de demanda.
- Varejo: consiste na escala dos trabalhadores nas empresas de varejo, de forma a atender da melhor forma a demanda, sendo um problema muito parecido com o escalonamento de teleatendentes.
- Manufatura: é um problema associado com a obtenção do melhor equilíbrio entre a demanda dos processos e a disponibilidade de trabalhadores, de forma a obter a melhor relação custo-benefício. Um exemplo é o escalonamento de trabalhadores nas grandes obras de construção.

A área de aplicação deste trabalho encaixa-se nos problemas de escalonamento nos sistemas de transporte, que, segundo Ernst *et al.* (2004), independentemente da forma de transporte possuem duas características em comum, que são:

- O envolvimento de aspectos temporais e espaciais, sendo que cada tarefa é caracterizada por apresentar um horário e uma localização inicial e um horário e localização final.
- Todas as tarefas a serem realizadas pelos colaboradores são determinadas por uma tabela de horários, sendo uma jornada denominada como a sequência de tarefas a serem realizadas por um colaborador.

Em Ernst *et al.* (2004) pode ser encontrada uma revisão mais detalhada do processo de escalonamento de pessoal, com a citação de diversos trabalhos já realizados na área em diferentes aplicações.

2.2 Problema de Escalonamento de Motoristas

O Problema de Escalonamento de Motoristas (PEM), também conhecido na literatura como *Crew Scheduling Problem* ou *Bus Driver Scheduling Problem*, é um problema de otimização combinatória de relevância prática que está presente no planejamento operacional das empresas de transporte coletivo. O problema é classificado como NP-Difícil (Leone *et al.*, 2010) e consiste em realizar a escala diária de viagens que deverão ser realizadas pelos motoristas, visando reduzir os custos operacionais e obedecer a uma série de restrições, provenientes da legislação trabalhista, de acordos com o sindicato dos motoristas e de políticas da empresa de transporte.

De acordo com Silva e Cunha (2010), uma escala equilibrada tem grande importância para a eficiência do sistema. Sendo assim, uma programação racional das escalas deve buscar uma alocação ótima da mão de obra, considerando o número mínimo de tripulações e de horas extras trabalhadas, para não incorrer em gastos excessivos com os operadores. A importância do processo de escalonamento de motoristas também é destacada em Lourenço *et al.* (2001) em virtude da grande redução de custos que a alocação eficiente desses recursos pode propiciar.

Na literatura relacionada com o PEM geralmente são encontrados alguns conceitos particulares do problema (Huisman *et al.*, 2005; Calvi, 2005; Marinho *et al.*, 2004), descritos a seguir:

- Viagem ou tarefa: percurso realizado entre duas oportunidades de troca consecutivas, sendo que não podem ocorrer trocas de tripulação durante a viagem, apenas no início ou no final;
- Bloco: sequência de viagens a serem realizadas por um veículo, geralmente começando e terminando na garagem;
- Oportunidades de Troca: locais onde e quando podem ocorrer trocas de tripulação, geralmente garagens e terminais;
- Fração de trabalho: sequência de viagens consecutivas sem intervalo;
- Jornada: programação diária das viagens de um motorista.

2.2.1 Métodos de resolução

Um Problema de Atribuição Multinível com Gargalo é apresentado por Carraresi e Gallo (1984) para resolver o PEM de forma a balancear a distribuição de tarefas, equilibrando a carga de trabalho dos motoristas. O problema é apresentado de uma forma generalizada, onde são atribuídos para cada jornada de trabalho pesos que representem uma medida do custo da jornada para os motoristas. Esse peso pode ser simplesmente a duração da jornada a ser realizada pelo motorista ou alguma outra grandeza que possa mensurar a carga de trabalho. A partir daí, o problema é encontrar jornadas balanceadas, minimizando a jornada de maior custo. Para resolver o problema, os autores propõem uma heurística que encontra uma solução subótima para o problema, utilizando instâncias geradas aleatoriamente e problemas reais.

Lourenço *et al.* (2001) propõem meta-heurísticas multiobjetivos para a resolução de problemas reais de programação da tripulação. Os autores apresentam um método de Busca Tabu (BT) Multiobjetivo e um Algoritmo Genético (AG) Multiobjetivo, sendo que ambos utilizam o método GRASP (*Greedy Randomized Adaptive Search Procedure*) como um procedimento para resolver os subproblemas de cobertura de conjuntos enfrentados pelos algoritmos. Os métodos propostos apresentaram bons resultados quando comparados com as soluções baseadas em Programação Linear (PL), provenientes da resolução feita pelo sistema de suporte à tomada de decisão GIST, adotado pelas empresas. Além disso, os usuários do sistema em diferentes companhias de transporte destacaram a importância de considerar diversos objetivos na qualidade da escala e a possibilidade de obter diversos cenários para análise, reduzindo significativamente a necessidade de ajustar manualmente a solução final fornecida pelo sistema, o que acarretou em maior aceitação do sistema junto às companhias.

Em Dias *et al.* (2002) também é proposto um Algoritmo Genético que leva em consideração diversas características importantes para obter uma solução de qualidade, tais como: custos, número de jornadas, porcentagem da escala realizada por jornadas viáveis e duração média das jornadas. O algoritmo foi testado com problemas de escalonamento de tripulação em companhias aéreas, disponíveis em uma base de dados, e com problemas de porte médio de empresas de ônibus de Portugal. As companhias de transporte estudadas utilizam o sistema GIST, que possui um módulo para realizar os escalonamentos, entretanto as soluções não satisfazem as necessidades dos usuários e estes acabam resolvendo o problema de forma “semimanual” por meio de uma interface gráfica do próprio sistema GIST, por considerarem

suas soluções melhores. A comparação do algoritmo foi feita com essas escalas “semimanuais” e os resultados mostram que os AGs produziram soluções satisfatórias rapidamente e com qualidade que atendessem às expectativas da companhia de transporte público, que passaram a adotar o algoritmo proposto resolvendo em poucos minutos um problema que levava diversas horas.

Um procedimento heurístico para resolver o PEM tomando como base a metaheurística *Simulated Annealing* (SA) é apresentada em Silva *et al.* (2002). Os autores justificam a escolha, pelo fato desta metaheurística ser robusta, de fácil implementação e produzir soluções de boa qualidade. Para a obtenção de soluções mais eficientes, são explorados três movimentos distintos, descritos a seguir, que definem respectivamente as estruturas de vizinhança $N^{(1)}(s)$, $N^{(2)}(s)$ e $N^{(3)}(s)$ de uma solução s :

- Inserir Tarefa: consiste em atribuir uma tarefa de uma tripulação i a uma tripulação j a partir do sorteio de duas tripulações quaisquer i e j , contanto que a tarefa i tenha pelo menos uma tarefa.
- Trocar Duas Tarefas: consiste em trocar duas tarefas p e q , escolhidas aleatoriamente, entre duas tripulações i e j , também escolhidas aleatoriamente, contanto que as tarefas i e j possuam pelo menos uma tarefa.
- Trocar Dois Pedacos de Jornada: consiste em selecionar duas tarefas p e q de duas tripulações distintas i e j e, a partir disso, compor a nova jornada da tripulação i com o pedaço anterior à tarefa p selecionada mais o novo pedaço começando da tarefa q . De forma análoga, o movimento é feito para a jornada j .

Souza *et al.* (2003) faz uma continuidade de Silva *et al.* (2002), utilizando novas estruturas de vizinhança para o SA e para o Método VNS. Os movimentos utilizados são os mesmos de Silva *et al.* (2002), descritos acima, entretanto, o critério de escolha dos dois tripulantes envolvidos no movimento é definido a partir de duas listas: L1, que contém 20% dos tripulantes com as maiores jornadas, e L2, que contém 20% dos tripulantes com menores jornadas. A partir disso, é selecionado um tripulante da lista L1 e outro da L2, de forma a minimizar a ociosidade ou sobrecarga na jornada de trabalho.

No método VNS, Souza *et al.* (2003) utilizam as três estruturas de vizinhança descritas acima, por ordem crescente de complexidade ($N^{(1)}(s)$, $N^{(2)}(s)$ e $N^{(3)}(s)$) e, para evitar a análise completa de uma dada vizinhança em cada iteração, a busca na vizinhança corrente é

interrompida sempre que ocorrer uma melhoria na solução. Na aplicação do SA, foram utilizadas as três estruturas de vizinhança para elaborar quatro estratégias de busca. Nas três primeiras, foram utilizadas, as estruturas $N^{(1)}(s)$, $N^{(2)}(s)$ e $N^{(3)}(s)$, respectivamente. Na quarta estratégia, a busca utiliza uma dessas estruturas a cada iteração, escolhidas de forma aleatória.

Os resultados da pesquisa de Souza *et al.* (2003) mostram que o Método VNS e o SA com o movimento 1-opt (Inserir Tarefa) produziram soluções de qualidade e com redução do número de tripulações e horas extras quando comparadas com a solução adotada pela empresa. Entretanto, as outras estratégias de busca para o SA sequer foram capazes de encontrar uma solução factível dentro do tempo de processamento estabelecido. Dessa forma, seria necessário um tempo de processamento maior para que essas estratégias apresentassem resultados satisfatórios.

Silva *et al.* (2004) apresentam uma metodologia que formula o PET como um modelo de particionamento, e utiliza o método *Simplex* para resolvê-lo. Para reduzir o número de possíveis combinações, as tarefas localizadas no início e no final das jornadas de cada veículo foram agrupadas de tal forma que o tempo mínimo de trabalho fosse atingido, o que resultou numa redução de 1190 para 535 tarefas. Em seguida, foram consideradas todas as possíveis combinações das tarefas na geração de jornadas e apenas as jornadas viáveis foram incluídas no modelo. Para validar o método proposto foram realizados testes com instâncias reais, resolvendo um problema de particionamento para cada linha separadamente, com o uso do pacote LINGO. Essa metodologia possibilita resolver o problema de cada linha individualmente de forma exata, entretanto, considerar o conjunto de todas as linhas como um único problema, possibilitando que a tripulação execute tarefas de linhas distintas, reduziria ainda mais os custos, horas extras e tempo ocioso, mas está prática não é permitida na empresa e o problema exato seria insolúvel do ponto de vista computacional.

Mauri e Lorena (2004) propõem uma metodologia interativa baseada na aplicação do Algoritmo de Treinamento Populacional (ATP) juntamente com Programação Linear (PL) para resolver o PEM. Nessa abordagem, o problema é descrito como o de formação de uma matriz, na qual as tripulações são representadas pelas colunas e as tarefas pelas linhas, que é usada para resolver o Problema de Particionamento de Conjuntos (PPC). No algoritmo proposto, é gerada aleatoriamente uma solução inicial contendo colunas que atendam às restrições essenciais e cubram todas as tarefas. Então a PL é resolvida e novas colunas são

geradas através do ATP. Esse processo é repetido até que a solução da PL estabilize ou que o limite de iterações seja alcançado. Feito isso, a PL é convertida em Programação Linear Inteira (PLI) e resolvida pelo software de otimização CPLEX. Para validar o algoritmo foram realizados testes com instâncias geradas aleatoriamente, baseadas em problemas reais, de até 500 tarefas e na comparação com a metaheurística *Simulated Annealing*, os resultados encontrados pelo método proposto foram superiores.

Em Marinho *et al.* (2004) o problema de programação da tripulação é abordado através de um método de Busca Tabu (BT), sendo desenvolvidas várias versões do método a partir de diferentes critérios de exploração das vizinhanças. Para definir a vizinhança são considerados dois tipos de movimentos: a realocação de uma tarefa de uma jornada para outra e a troca de tarefas entre duas jornadas. As versões propostas pelos autores são comparadas com um método de VNS. Na comparação, utilizando dados reais de uma empresa de transporte público da cidade de Belo Horizonte, os resultados mostram a superioridade do método proposto, conseguindo melhores soluções e em tempo menor.

Uma solução para o PPT baseada em *Iterated Local Search* (ILS) e Busca Tabu é apresentada em Gonçalves (2010). O autor propõe duas versões da meta-heurística *Iterated Local Search* e uma versão de BT para efeitos de comparação. As estruturas de vizinhança também consideram os movimentos de Realocação e Troca de tarefas. Gonçalves (2010) também considera uma heurística para controle do número de jornadas, eliminando parte das jornadas vazias, visando reduzir o número de combinações e acelerar a busca, porém mantendo outras de forma a evitar que uma solução de qualidade superior com maior número de jornadas seja descartada. As instâncias utilizadas para teste foram geradas baseando-se em dados reais de uma empresa de transporte público que opera em Belo Horizonte. Na comparação, a meta-heurística ILS apresentou os melhores resultados quando comparada com uma abordagem ILS com memória de perturbação de curto prazo e com uma abordagem baseada em BT.

Para a resolução de pequenas instâncias do PEM, Santos e Mateus (2007) apresentam um método exato de geração de colunas que utiliza um Algoritmo Genético (AG) para a resolução dos subproblemas de geração de novas colunas, geralmente resolvidos por PL ou por PLI. No método proposto, o AG é utilizado para gerar a maioria das colunas e a PLI é utilizada caso o AG falhe na geração boas colunas. A inclusão do AG na resolução dos subproblemas mostrou que apesar dele gerar mais colunas, a maioria inútil, ele rapidamente

adiciona as melhores colunas que fazem parte da solução ótima, o que possibilita o algoritmo convergir bem mais rápido que as abordagens tradicionais de resolução dos subproblemas encontradas na literatura.

Para a elaboração de jornadas de trabalho para motoristas e cobradores de ônibus, Siqueira *et al.* (2004) baseiam-se em problemas de emparelhamento e utilizam o algoritmo do *Matching* de peso máximo aplicado à três empresas de transporte coletivo de Curitiba. Na primeira fase da abordagem, os autores resolvem o problema de programação diária das jornadas. Na segunda fase, são atribuídas jornadas de fim de semana para cada jornada de dia de semana, formando as jornadas semanais de trabalho. Por fim, na terceira fase são designados aos funcionários jornadas semanais de trabalho, de acordo com suas preferências e disponibilidade. Todas as fases são resolvidas como um problema de emparelhamento, utilizando-se do algoritmo do *Matching* de peso máximo. Foram conseguidos resultados muito satisfatórios na comparação com as escalas obtidas pelos métodos heurísticos utilizados pelas empresas.

Outra metodologia baseada no problema de emparelhamento é apresentada por Silva *et al.* (2005), baseada no trabalho de Siqueira (1999), entretanto com aplicações sucessivas do modelo de emparelhamento. Os autores apresentam duas metodologias distintas para a construção da rede sobre a qual é aplicado o algoritmo de emparelhamento com peso máximo. A primeira metodologia consiste na divisão das tarefas em períodos para forçar o agrupamento das tarefas pertencentes ao mesmo período, sendo este manhã, tarde ou noite. A segunda metodologia consiste no agrupamento das tarefas em blocos de maior duração, que representa a situação operacional, na qual a tripulação inicia a operação a partir da garagem e permanece com o veículo por um dado período mínimo de tempo, sendo que o mesmo ocorre com a tripulação que retorna com o veículo à garagem no final da operação. As soluções geradas pelas duas metodologias atenderam às restrições com custos reduzidos. Entretanto, pode-se perceber que a segunda metodologia apresentou melhores resultados do que a primeira.

Calvi (2005) propõe um novo algoritmo heurístico para a resolução do PEM baseado na resolução de sucessivos Problemas de Designação, também conhecido como Problema de Atribuição, que surgem de um modelo baseado em grafo multipartido. O algoritmo é composto de duas fases: construção da solução inicial e melhoramento da solução. Na fase

construtiva são resolvidos problemas de designação para cada camada do grafo, que relaciona os custos de atribuir uma tarefa i a uma jornada j . Na fase de melhoramento são utilizados dois procedimentos de melhoria. O primeiro realiza a divisão das jornadas em jornadas parciais à esquerda e à direita, sendo que a recombinação dessas jornadas é dada pela resolução de um Problema de Designação. O segundo procedimento consiste em separar a última tarefa das jornadas com horas extras e recombinar as jornadas parciais para obter uma nova solução.

Para formação das matrizes de custo, Calvi (2005) realiza testes com três diferentes funções de custo considerando minutos pagos, minutos ociosos e minutos extras. Para comparação foram utilizados dados reais e os resultados alcançados foram melhores que os resultados obtidos por um modelo do Problema de Cobertura de Conjuntos (PCC), sendo que as funções de custo que apresentaram os melhores resultados foram: a função que considera os minutos ociosos na jornada e a função que considera o custo como minutos ociosos na jornada mais o custo de hora extra. Na comparação com a solução da empresa o algoritmo proposto apresentou qualidade superior, cumprindo todas as regras impostas.

Leone *et al.* (2010) propõem nova formulação matemática para o PEM considerando as restrições enfrentadas por empresas de transporte italianas. A solução exata do modelo matemático proposto foi obtida utilizando as ferramentas GAMS (*General Algebraic Modelling System*) 2005 e Cplex 9.1.2 e os resultados obtidos mostram que o modelo exato pode ser aplicado apenas para instâncias de pequeno ou médio porte. Devido à complexidade do problema, soluções de boa qualidade para as grandes instâncias podem ser obtidas apenas por meio de técnicas heurísticas. Em vista disso, Leone *et al.* (2010) também propõem um algoritmo baseado na heurística GRASP, utilizando três estratégias de busca em vizinhança, que consideram: troca de tarefas entre jornadas, recombinação de jornadas parciais e recombinação de jornadas parciais em diferentes pontos. Testes realizados com instâncias de dados reais puderam comprovar a eficácia do método proposto, comparado a um método exato.

Em Silva e Cunha (2010) também é apresentado um modelo de resolução do PEM baseado na heurística GRASP, entretanto, a busca local é realizada pela técnica de Busca em Vizinhança de Grande Porte (BVGP). Os autores ressaltam que a grande vantagem de utilizar essa técnica de busca é que ela possibilita explorar um espaço de soluções muito mais abrangente do que

aqueles pesquisados por outras heurísticas encontradas na literatura. O diferencial dessa técnica BVGP é que ela possibilita realizar movimentos de realocação e de troca *2-optimal*, assim como movimentos de troca de *3-optimal* até *n-optimal*, onde n representa o número de tripulantes que compõem a escala. Para aplicar a técnica BVGP foi concebida uma rede de melhoria, que permite melhorar a qualidade das soluções por meio da detecção de ciclos válidos.

A heurística proposta por Silva e Cunha (2010) foi testada com dados operacionais reais de uma empresa de transporte público de Belo Horizonte e pôde-se perceber que o modelo é capaz de construir soluções com as mesmas características das utilizadas na prática, porém, com ganhos consideráveis tendo em vista o número de funcionários necessários e a quantidade de horas extras constantes nas jornadas.

Os trabalhos acima apresentados utilizam uma abordagem sequencial para a resolução do problema de planejamento das escalas de trabalho do transporte coletivo, resolvendo o problema de escalonamento de veículos e o problema de escalonamento da tripulação em etapas distintas. Outra abordagem para a resolução do problema é a abordagem integrada, na qual os problemas de escalonamento de veículos e motoristas são resolvidos simultaneamente.

Uma abordagem integrada do problema de escalonamento de veículos e de motoristas para diversas garagens é realizada por Huisman *et al.* (2005), no qual são propostos dois diferentes modelos e algoritmos para a resolução do *Multiple-Depot Integrated Vehicle and Crew Scheduling* (MD-VCSP). Ambos os algoritmos são baseados em uma combinação de técnicas de geração de colunas com relaxação Lagrangeana. São realizados experimentos com oito instâncias derivadas de um problema real envolvendo 1104 viagens e com diversas instâncias geradas aleatoriamente, de no máximo 400 viagens. Em todas as comparações, os algoritmos propostos apresentaram melhores resultados que uma abordagem sequencial.

Prata (2010) apresenta uma revisão bibliográfica sobre o problema de programação integrada de veículos e motoristas, apresentando os principais modelos matemáticos para a abordagem integrada e também as abordagens tradicionais sequenciais.

Na Tabela 1 é apresentado um resumo dos trabalhos citados relacionados com o PEM, o método utilizado e uma característica que o diferencie.

Tabela 1: Quadro-resumo dos trabalhos citados

Referência	Método	Particularidade
Carraresi e Gallo (1984)	Problema de Atribuição com Gargalo	Objetiva encontrar jornadas balanceadas
Lourenço <i>et al.</i> (2001)	Busca Tabu e Algoritmo Genético	Multiobjetivo e utilizam a heurística GRASP para resolver subproblemas
Dias <i>et al.</i> (2002)	Algoritmo Genético	Multiobjetivo
Silva <i>et al.</i> (2002)	<i>Simulated Annealing</i>	Utilizam três movimentos distintos de exploração
Souza <i>et al.</i> (2003)	<i>Simulated Annealing e Variable Neighborhood Search</i>	Mesmos movimentos de Silva <i>et al.</i> (2002), porém selecionando apenas os 20% dos tripulantes com maiores e menores jornadas para o movimento
Silva <i>et al.</i> (2004)	Geração de Colunas	Resolvem o problema exato para cada linha da empresa
Mauri e Lorena (2004)	Geração de Colunas	Utilizam um Algoritmo de Treinamento Populacional combinado com Programação Linear
Marinho <i>et al.</i> (2004)	Busca Tabu	São desenvolvidas diversas versões a partir de diferentes critérios de exploração da vizinhança
Gonçalves (2010)	<i>Iterated Local Search</i> e Busca Tabu	Controle do número de jornadas
Santos e Mateus (2007)	Geração de Colunas	Utiliza um Algoritmo Genético para resolver os subproblemas de geração de novas colunas
Siqueira <i>et al.</i> (2004)	Algoritmo do <i>Matching</i> de peso máximo	Resolve o escalonamento diário e semanal
Silva <i>et al.</i> (2005)	Algoritmo do <i>Matching</i> de peso máximo	Baseado em Siqueira (1999), porém com aplicações sucessivas do modelo de emparelhamento
Calvi (2005)	Problema de Atribuição	Faz uma comparação entre diferentes funções de custos de atribuição
Leone <i>et al.</i> (2010)	Exato e GRASP	Modelo exato para os menores problemas e heurística GRASP para os maiores
Silva e Cunha (2010)	GRASP	Busca local é realizada pela técnica de Busca em Vizinhança de Grande Porte (BVGP)
Huisman <i>et al.</i> (2005)	Geração de Colunas com Relaxação Lagrangeana	Abordagem integrada do Escalonamento de Veículos e Tripulação

2.3 Problema de Atribuição

O Problema de Atribuição (PA), também encontrado na literatura como Problema de Designação ou *Assignment Problem*, é um clássico problema de Otimização Combinatória em Pesquisa Operacional. De acordo com Hillier e Lieberman (2010), é um tipo especial de problema de programação linear no qual os designados são indicados para a realização de tarefas. Os exemplos mais comuns do problema de atribuição são as situações de designar tarefas para máquinas, tarefas para funcionários ou funcionários para máquinas.

A solução do PA equivale ao emparelhamento perfeito de custo mínimo em um grafo bipartido. Assim, dada uma matriz de custos de dimensões $n \times n$, o problema consiste em associar cada linha i a uma coluna j sob um custo c_{ij} , de modo que a soma dos custos seja a menor possível. O problema pode ser formulado da seguinte maneira:

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij}; \quad (1)$$

$$\text{Sujeito a: } \sum_{i=1}^n x_{ij} = 1; \quad j = 1, \dots, n; \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1; \quad i = 1, \dots, n; \quad (3)$$

$$x_{ij} \in \{0,1\}; \quad i = 1, \dots, n; j = 1, \dots, n. \quad (4)$$

A atribuição de uma linha i a uma coluna j faz com que x_{ij} assumam valor 1 ou 0, caso contrário. A função-objetivo (1) minimiza o custo da soma das atribuições entre linhas e colunas; a restrição (2) exige que para cada linha haja uma coluna associada; a restrição (3) garante que para cada coluna seja designada uma linha; a restrição (4) garante que as variáveis envolvidas assumam apenas os valores de decisão 0 e 1.

Em outras palavras, de acordo com Hillier e Lieberman (2010) o problema de designação deve satisfazer as seguintes hipóteses:

1. O número de designados e o número de tarefas são iguais, sendo representado por n .
2. Deve-se atribuir a cada designado exatamente uma tarefa.
3. Cada tarefa deve ser realizada exatamente por um designado.
4. Há um custo associado ao designado i ($i=1, 2, \dots, n$) executando a tarefa j ($j=1, 2, \dots, n$).

5. O objetivo é determinar como todas as n designações devem ser feitas para minimizar o custo total.

O modelo matemático e as hipóteses apresentadas acima tratam da minimização das associações, que geralmente tratam de custos e tempo. Entretanto, caso o objetivo seja maximizar as associações, no caso do lucro, basta utilizar valores opostos, no caso, negativos.

Os problemas que não são balanceados, ou seja, os quais não há igual número de designados e tarefas, podem ser facilmente reformulados e balanceados através da introdução de tarefas fictícias, caso o número de designados seja maior, ou de designados fictícios caso contrário (HILLIER e LIEBERMAN, 2010).

Um dos algoritmos de otimização mais utilizados para a resolução desse problema é o Método Húngaro, denominação dada em homenagem aos pesquisadores húngaros que o desenvolveram, que explora eficientemente a estrutura do PA.

Outro algoritmo para a resolução do problema foi proposto por Carpaneto e Toth (1987). Esse algoritmo combina o procedimento do Menor Caminho Aumentante com o Método Húngaro, garante solução ótima e tem complexidade $O(n^3)$. Além disso, de acordo com experimentos realizados pelos autores, o algoritmo proposto se mostrou mais rápido que o método húngaro na maioria das instâncias utilizadas.

2.4 Grafos

Grafos são modelos que representam objetos e suas relações e estão presentes em diferentes tipos de problemas, como: estrutura web, roteamento e alocações (*matching*). De acordo com Ziviani (2004) um grafo é uma estrutura $G = (V,A)$, onde V é um conjunto não vazio de elementos denominados vértices e A é um conjunto de elementos denominados arestas, ou seja, um grafo pode ser definido como um conjunto de elementos e suas relações.

Segundo Ziviani (2004), considerando um grafo $G = (V,A)$, podemos chamá-lo de grafo orientado se V é um conjunto finito de vértices e A é um conjunto de arestas com uma relação binária em V , ou seja, se $(v_i,v_j) \neq (v_j,v_i)$, onde $v_i, v_j \in V$. Quando um grafo $G = (V,A)$ possui o conjunto de arestas constituído de pares de vértices não ordenados, este é chamado

de grafo não direcionado, segundo Ziviani (2004), ou seja, quando $(v_i, v_j) = (v_j, v_i)$. Na Figura 2 são apresentados exemplos para os dois tipos de grafos.

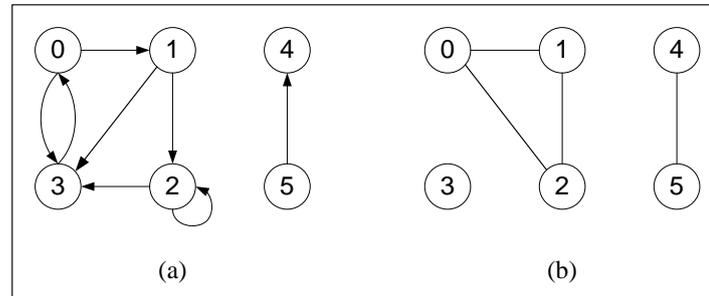


Figura 2: (a) Grafo direcionado (b) Grafo não direcionado. Adaptado de: Ziviani, 2004

De acordo com Cormen *et al.* (2002), um grafo bipartido é um grafo $G = (V, A)$ em que V pode ser particionado em dois subconjuntos disjuntos V_1 e V_2 composto de arestas $(u, v) \in A$, tais que $u \in V_1$ e $v \in V_2$, ou vice-versa. Dessa forma, o grafo fica dividido em dois conjuntos distintos, de modo que toda aresta A tem uma extremidade em V_1 e outra em V_2 .

Por sua vez, um grafo multipartido consiste de um grafo $G = (V, A)$ em que V pode ser particionado em n subconjuntos disjuntos V_1, V_2, \dots, V_n , sendo que dois vértices u e v podem ser adjacentes se pertencem a conjuntos distintos, de modo que toda aresta A tem uma extremidade em V_i e outra em V_{i+1} , $i = 1, \dots, n-1$.

2.5 Algoritmos Heurísticos e Meta-heurísticas

Para problemas que apresentam grande quantidade de informação e muitas possibilidades de combinação, torna-se inviável realizar testes com todas as combinações, ou seja, resolvê-lo de forma exata, pois o tempo necessário para tal resolução seria inviável para uma aplicação prática. Os algoritmos heurísticos foram desenvolvidos com a finalidade de se resolver estes problemas de elevado nível de complexidade em tempo computacional razoável. Assim, os algoritmos heurísticos não garantem encontrar a solução ótima do problema, mas são capazes de retornar uma solução de qualidade em tempo aceitável (CHAVES, 2003).

Impulsionadas pelos complexos problemas de otimização, surgiram diversas propostas de meta-heurísticas, que consistem em um tipo de algoritmo que basicamente tenta combinar métodos heurísticos básicos em um nível superior para uma exploração mais eficiente e eficaz

do espaço de busca (BLUM e ROLI, 2003). Entre as meta-heurísticas encontram-se os Algoritmos Genéticos, *Ant System*, *Iterated Local Search*, Busca Tabu, *Variable Neighborhood Search*, *Simulated Annealing*, GRASP, entre outros.

Neste trabalho foi utilizada a metaheurística VND (*Variable Neighborhood Descent*), que é uma variação do VNS (*Variable Neighborhood Search*). O VNS é uma meta-heurística proposta por Hansen e Mladenovic (2001) que aplica uma estratégia baseada em trocas dinâmicas entre estruturas de vizinhança. No VND é aplicada uma busca local pela melhor melhoria e caso seja encontrado um mínimo local, a busca continua com outra estrutura de vizinhança. (BLUM e ROLI, 2003).

3 DESENVOLVIMENTO

3.1 Caracterização do problema

O PEM consiste basicamente em realizar a programação da escala diária de viagens dos motoristas, com o objetivo de minimizar o custo operacional total e atender às restrições do problema, que consideram aspectos legais, acordo coletivo com o sindicato da categoria e políticas da empresa.

A programação das escalas dos motoristas geralmente recebe como dados de entrada a programação dos veículos previamente definida e as restrições do problema, impostas pela CLT (Consolidação das Leis do Trabalho), pelo acordo coletivo com o sindicato e pela política da empresa. A programação dos veículos define os blocos de veículos e as tarefas, ou viagens, são definidas como o percurso no qual não é possível realizar a troca de motoristas, conforme apresenta Silva e Cunha (2010). Dessa forma, uma tarefa apresenta informações como linha operada, pontos de origem e destino, horário de início, duração e veículo utilizado.

No estudo de caso considerado são utilizadas as mesmas instâncias de Calvi (2005), apresentadas na Tabela 2, que compreendem duas instâncias de dados reais, fornecidas por duas empresas de transporte urbano de passageiros do estado do Paraná e oito instâncias geradas aleatoriamente por Calvi (2005) a partir do problema de 2313 viagens.

Tabela 2: Instâncias utilizadas

Instância	Número de viagens	Tipo
RE412	412	Problema real
RE2313	2313	Problema real
AL130	130	Gerada aleatoriamente
AL251	251	Gerada aleatoriamente
AL512	512	Gerada aleatoriamente
AL761	761	Gerada aleatoriamente
AL1000	1000	Gerada aleatoriamente
AL1253	1253	Gerada aleatoriamente
AL1517	1517	Gerada aleatoriamente
AL2010	2010	Gerada aleatoriamente

As principais características e a descrição das restrições envolvidas no problema em estudo são apresentadas a seguir:

- A escala de veículos já está definida e segue o planejamento realizado pela empresa, conforme Calvi (2005);
- A programação das escalas dos motoristas considera um horizonte de planejamento de um dia;
- São considerados dois possíveis pontos de troca entre os motoristas: a garagem da empresa e o terminal urbano;
- As trocas de veículos entre os motoristas são permitidas, mas devem ser feitas sem excessos;
- A jornada de trabalho normal do motorista tem duração de 7 horas e 20 minutos. Caso a jornada realizada seja menor, ainda assim será pago conforme o tempo mínimo pago por uma jornada de trabalho, ou seja, 7 horas e 20 minutos;
- O tempo máximo de horas extras trabalhadas não deve exceder 2 horas, ou seja, a jornada com horas extras poderá ter duração máxima de até 9 horas e 20 minutos trabalhados;
- O valor das horas extras trabalhadas recebe um adicional de 50% sobre o valor pago pela hora normal;
- A duração máxima do trabalho contínuo é de 6 horas. Quando a duração do trabalho ultrapassar 6 horas, deverá ser concedido um intervalo para descanso;
- O intervalo para descanso deverá ter um tempo mínimo de 1 hora e 30 minutos e máximo de 5 horas;
- A extensão máxima da jornada, considerando o tempo trabalhado mais o tempo de folga, deve ser de até 13 horas. Isso ocorre, pois o intervalo entre o fim da jornada de um dia e o início da jornada do outro dia deve ser de no mínimo 11 horas, assim considerando que o motorista poderá executar a mesma jornada no dia seguinte temos que a duração da jornada total não deve ser maior do que 13 horas;
- As horas trabalhadas no período noturno, entre as 22 horas de um dia e às 05 horas do dia seguinte possuem duração de 52,5 minutos;
- Sobre as horas trabalhadas no horário noturno haverá um adicional de 20% sobre o custo da jornada normal.

Complementando a descrição do problema, a Tabela 3 apresenta de forma sintética a duração da hora de trabalho e o custo do minuto pago nos diferentes períodos do dia.

Tabela 3: Duração da hora de trabalho (em min.) e custo (em min. pagos) por período do dia

Período	Intervalo	Duração da hora de trabalho (em minutos)	Custo do minuto (em minutos pagos)
Período normal	05:00 ~ 22:00	60,0	1,00
Período noturno	22:00 ~ 05:00	52,5	1,20

Por sua vez, a Tabela 4 apresenta de forma resumida as restrições referentes às jornadas de trabalho dos motoristas.

Tabela 4: Restrições das jornadas de trabalho

Restrição	Mínimo	Máximo
Duração do trabalho normal	-	07:20
Duração do trabalho (com hora extra)	-	09:20
Duração do trabalho contínuo	-	06:00
Duração do intervalo	01:30	05:00
Duração da jornada total	-	13:00

3.2 Algoritmos Propostos

3.2.1 Considerações Iniciais

Para a resolução do Problema de Escalonamento de Motoristas é proposto um algoritmo heurístico híbrido baseado na resolução de sucessivos Problemas de Atribuição (PA). O algoritmo é dividido em duas fases: construção da solução inicial e melhoramento da solução. Para a resolução dos PAs, foi implementado o algoritmo de Carpaneto e Toth (1987), um algoritmo exato que garante a solução ótima para o PA, que combina o método húngaro com o método *shortest augmenting path*.

Na fase de construção da solução inicial, é construído um grafo multipartido no qual as jornadas são representadas e que será utilizado para realizar as recombinações na fase de melhoramento. Nesse grafo, os vértices representam os pontos de parada e as arestas correspondem à duração das viagens.

Na fase de melhoramento, são utilizados dois procedimentos baseados em recombinações de trechos entre jornadas: o PCR e o K-Swap, executados de maneira intercalada. Diferentes testes são realizados, variando a sequência de execução destes procedimentos.

3.2.2 Construção da solução inicial

O primeiro procedimento para a resolução do problema consiste em dividir todas as viagens que devem ser realizadas na programação em subgrupos de viagens, de forma que cada subgrupo contenha apenas viagens que não podem ser designadas em sequência. Esses subgrupos são muito importantes e são utilizados durante toda a resolução, pois constituem as camadas do problema, onde serão designadas as tarefas para cada motorista e realizadas as recombinações entre as jornadas, através da resolução de PAs.

A construção da solução inicial é feita de forma similar à realizada em Calvi (2005) e consiste em gerar um grafo multipartido através da resolução de sucessivos PAs, no qual cada jornada de trabalho corresponde a um caminho da primeira a última camada. Em cada camada é criada uma matriz de custos quadrada $C^k = [c_{ij}^k]$ de ordem n_k , onde n_k corresponde a soma do total de jornadas já criadas, n_{jor} , com o total de viagens que devem ser designadas na camada k , v_k . Na criação da matriz de custos é necessária a inclusão de tarefas e/ou jornadas fictícias para que a matriz se torne quadrada, com o número de linhas igual ao número de colunas, o que é um requisito para resolver o PA, conforme alerta Pentico (2007). O algoritmo inicia com o valor de n_{jor} igual a zero e conforme as viagens vão sendo alocadas o valor de n_{jor} vai sendo incrementado conforme crescem o número de jornadas.

As linhas da matriz de custos C podem ser divididas em duas partes:

- Jornadas, de tamanho n_{jor} , consistindo das jornadas já alocadas, permitindo que novas tarefas sejam alocadas a essas jornadas;
- Jornadas Fictícias, de tamanho v_k , que consistem de possíveis novas jornadas, para garantir que todas as viagens sejam cumpridas mesmo se as jornadas atuais não puderem receber estas tarefas.

Por sua vez, as colunas da matriz C também são divididas em duas partes:

- Viagens, de tamanho v_k , consistindo das novas viagens que devem ser designadas na camada k ;
- Viagens Fictícias, de tamanho n_{jor} , que podem assumir intervalo de folga ou tarefa vazia caso sejam designadas para jornadas já existentes, ou que podem ser desconsideradas caso sejam designadas para jornadas fictícias.

Com relação às tarefas vazias, caso estas sejam designadas no meio de uma jornada, representarão tempo ocioso na jornada, no qual o motorista está sendo pago, mas não realiza nenhuma viagem, o que deve ser evitado. Caso seja designada para uma jornada fictícia, esta tarefa é simplesmente desconsiderada.

Conforme explicado, as linhas e colunas da matriz são divididas em duas partes, resultando em quatro blocos, com diferentes critérios de custos, conforme ilustrado na Figura 2.

		Viagens	Viagens Fictícias
Jornadas		<u>Bloco 1</u>	<u>Bloco 2</u>
		$c_{ij}^k = f(i, j)$ ou $c_{ij}^k = \infty$	$c_{ij}^k = f(i, j) + CF$
Jornadas Fictícias		<u>Bloco 3</u>	<u>Bloco 4</u>
		$c_{ij}^k = CNJ$	$c_{ij}^k = 0$

Figura 3: Estrutura da matriz de custos na construção da solução inicial

Para mais detalhes, segue descrição dos quatro blocos de custos da matriz:

- i. Bloco 1: formado pelas viagens que devem ser alocadas na camada em resolução e pelas jornadas já atribuídas. A preferência é que as novas viagens sejam atribuídas às jornadas já existentes. Caso essa atribuição seja viável, a função $f(i, j)$ retorna o custo de atribuir a tarefa j à jornada i , caso contrário a função retorna um custo infinito, para evitar a atribuição;
- ii. Bloco 2: formado pela viagens fictícias e pelas jornadas já atribuídas. Nesse bloco, as jornadas fictícias podem representar um intervalo de folga ou uma tarefa em vazio. Nesse caso, a função $f(i, j)$ retorna o custo de atribuir um intervalo de folga ou um tempo ocioso à jornada i . Além disso, é acrescido um custo CF que torna a atribuição de viagens fictícias menos atrativas;
- iii. Bloco 3: formado pelas viagens que devem ser alocadas na presente camada e pelas jornadas fictícias. A função deste bloco é permitir que novas jornadas sejam criadas

quando as viagens não puderem ser designadas às jornadas já existentes. Para controlar o crescimento do número de jornadas, este bloco recebe um custo CNJ , cujo valor é o custo de uma nova jornada, considerando o tempo mínimo pago.

- iv. Bloco 4: formado pelas viagens fictícias e pelas jornadas fictícias. Sua função é apenas completar a matriz. Assim, este bloco recebe custo zero.

Na criação da matriz de custos, o custo de atribuir uma viagem j a uma jornada i , dado pela função $f(i, j)$, pode seguir duas diferentes equações, $f1$ e $f2$, descritas abaixo, sendo a principal diferença o fato de $f2$ não considerar o custo de horas extras, considerado pela $f1$.

$$f1(i, j) = CO(i, j) + CHE(i, j) + CTV(i, j) + Pen.nRV \quad (5)$$

$$f2(i, j) = CO(i, j) + CTV(i, j) + Pen.nRV \quad (6)$$

Nas equações (5) e (6), $CO(i, j)$ é o custo em minutos ociosos, $CHE(i, j)$ é o custo de horas extras, $CTV(i, j)$ penaliza caso haja troca de veículos, nRV é o número de restrições violadas nessa atribuição e Pen é o custo de penalidade devido às violações de restrições.

Seguindo a metodologia, um problema de atribuição é criado para cada camada do problema. A resolução do PA indica para quais jornadas devem ser designadas as tarefas, permitindo a criação de novas jornadas. A solução inicial é obtida após a resolução do PA para todas as camadas do problema. De forma resumida, a Figura 3, descreve o procedimento de construção da solução inicial.

Construção da Solução Inicial

Início;

Inicialize os dados;

Divida as viagens em subgrupos que contenham as viagens que não podem ser alocadas em sequência;

Para $a = 1$ até o total de camadas k faça:

Gere a matriz de custos C correspondente à camada a ;

Resolva o PA da matriz C ;

Atribua as viagens aos motoristas conforme o resultado obtido;

Fim.

Figura 4: Passos para a construção da solução inicial

Um exemplo de solução inicial é apresentado na Figura 4, onde são apresentadas as jornadas, as camadas, os locais de início e término das viagens e a duração das viagens. Na figura, os locais de início e término das viagens são representados pelos vértices com as letras G e T, enquanto a duração das viagens ou do intervalo de folga é representada pelo comprimento das setas.

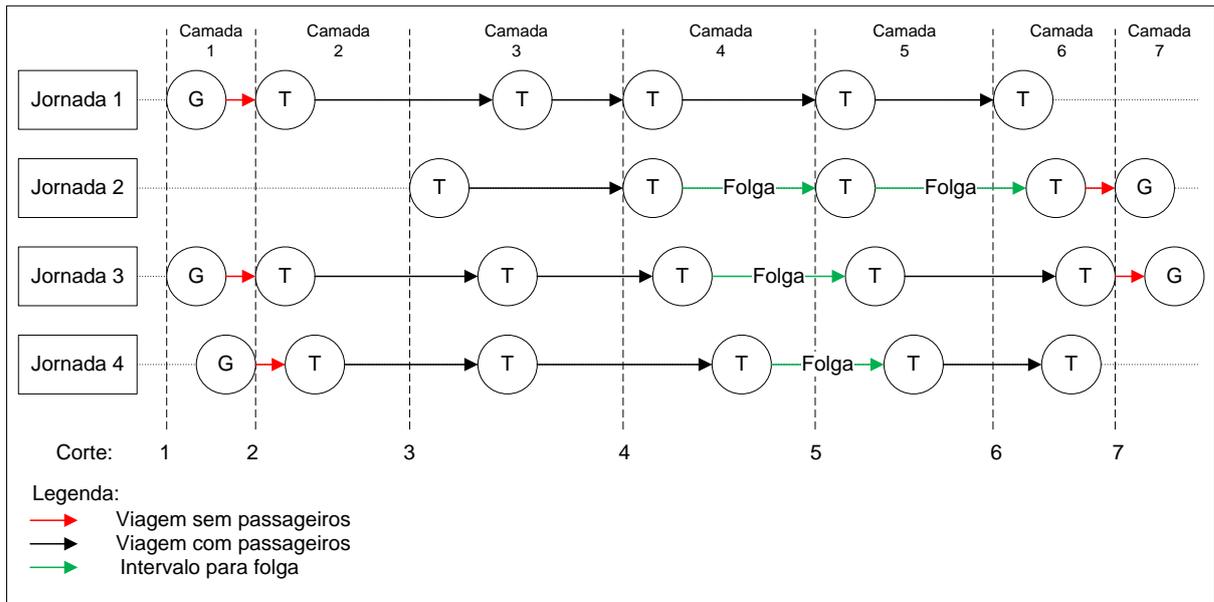


Figura 5: Exemplo de solução inicial

3.2.3 Fase de Melhoramento

A fase de melhoramento é composta por dois procedimentos, o Procedimento de Cortes e Recombinações (PCR) e o *K-Swap*, que são utilizados para minimizar o custo total da solução. De posse do grafo com as jornadas construídas, como no exemplo da Figura 4, o PCR busca reduzir o custo total da solução através da troca de pedaços entre jornadas. Para isso, realiza um corte entre duas camadas, que divide cada uma das n jornadas em duas jornadas parciais, ficando uma parte à esquerda e outra parte à direita do corte. Após o corte, é calculado o custo de se associar cada um dos n trechos de jornada à esquerda do corte com cada um dos n trechos à direita do corte. Assim, é construída a matriz de custos $D^k = [d_{ij}^k]$, de dimensões $n \times n$, onde n corresponde ao número total de jornadas e o valor de d_{ij}^k é dado por $g(i, j)$, que corresponde ao custo de associar a jornada à esquerda i com a jornada à direita j , se a associação for possível, mais possíveis penalidades. Caso essa associação não seja possível, d_{ij}^k recebe custo infinito, conforme demonstrado na Figura 5.

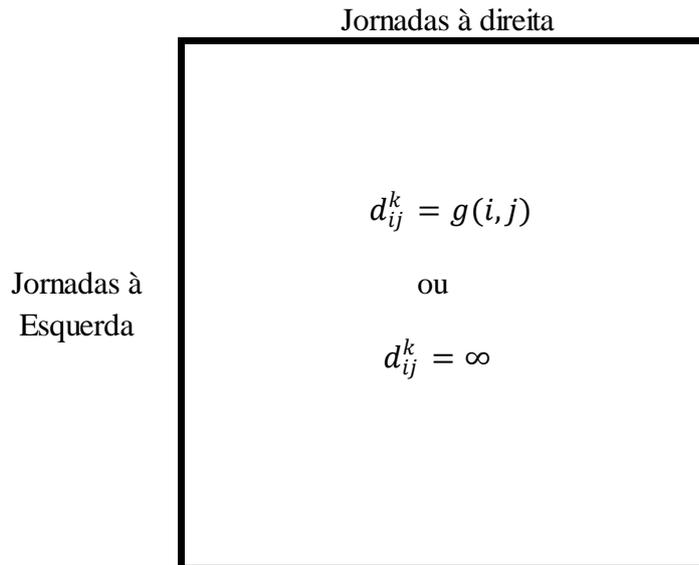


Figura 6: Estrutura da matriz de custos na execução do PCR

A Figura 6 exemplifica as possibilidades de recombinação do PCR entre as jornadas ao resolver a camada 3, representadas pelas setas pontilhadas. Note que para a Jornada 1 não é possível qualquer recombinação, uma vez que sua viagem termina após o horário de início das outras viagens da camada. O mesmo acontece ao tentar recombinar as Jornadas 3 e 4 com a Jornada 2. Nesses casos, a associação recebe um custo infinito, impedindo que essas trocas ocorram.

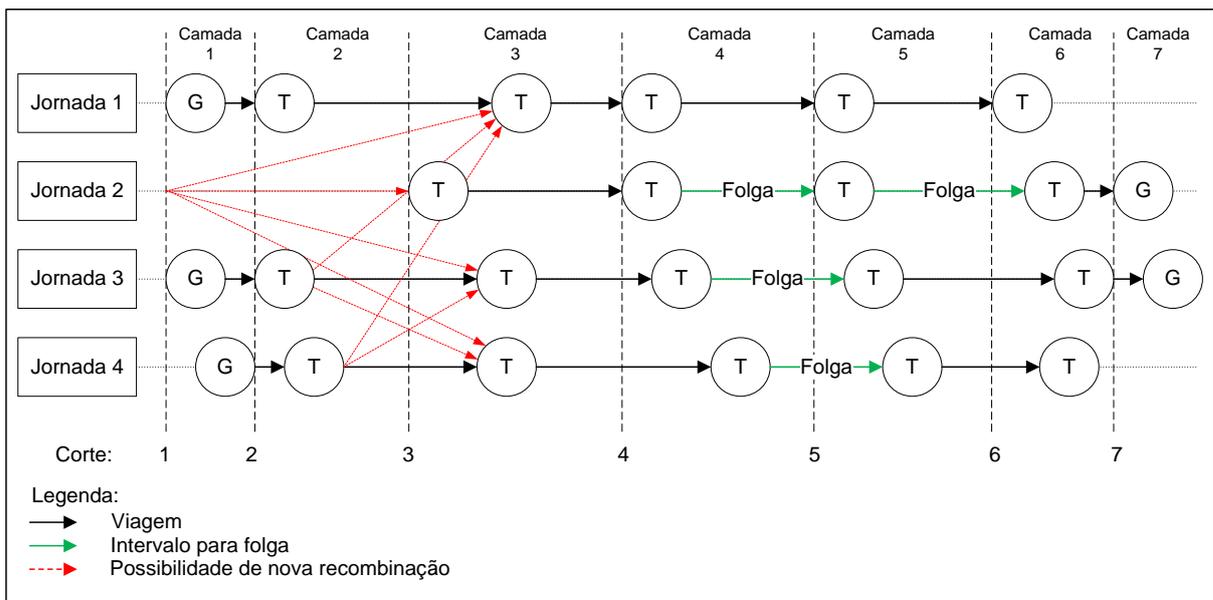


Figura 7: Exemplo de possíveis recombinações do PCR ao resolver a camada 3

De posse da matriz de custos D , é resolvido o PA para a camada correspondente e com base na solução é realizada a recombinação dos trechos, podendo acarretar na formação de novas

jornadas. Uma iteração completa do procedimento consiste em realizar cortes e recombinações antes de todas as camadas do problema, exceto antes da primeira camada. No Corte 1, realizado antes da Camada 1 o PCR não é executado pois o que ocorreria seria a troca de jornadas completas, que não resultaria em nenhuma melhoria na solução, uma vez que no problema em estudo os motoristas trabalham sobre um mesmo contrato de trabalho, sujeito às mesmas condições. A Figura 7 exemplifica como poderia ser realizada a recombinação no PCR. Note que houve uma troca entre as jornadas 3 e 4, permitindo reduzir a duração da Jornada 3.

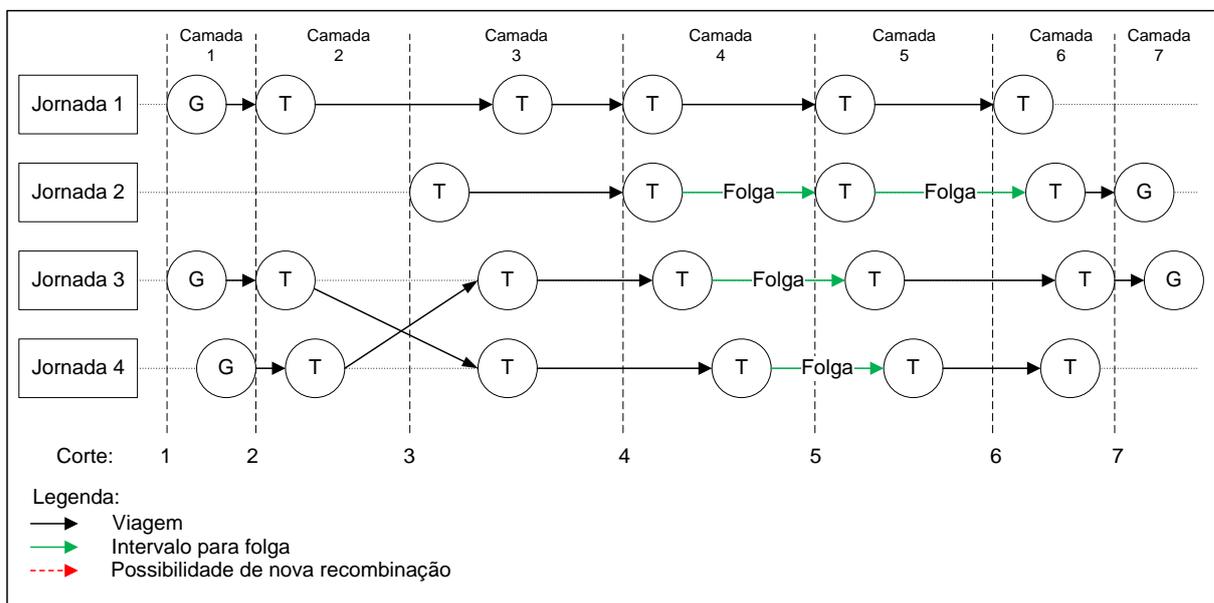


Figura 8: Exemplo de recombinação após execução do PCR na camada 3

Na Figura 8 pode ser visualizado um esquema geral que descreve os passos de execução do PCR, que é um procedimento similar ao procedimento M1 apresentado por Calvi (2005).

Procedimento PCR

Início;

Inicialize os dados;

Para $a = 1$ até o número total de camadas k faça:

Efetue uma divisão na posição de corte a ;

Gere a matriz de custos D , referente à camada a ;

Resolva o PA da matriz D ;

Faça a recombinação das jornadas conforme o resultado obtido;

Fim.

Figura 9: Passos para a execução do PCR

O segundo procedimento de melhoramento, o *K-Swap*, é um procedimento de trocas variáveis, e consiste em selecionar K camadas consecutivas e verificar a possibilidade de realizar trocas desses blocos de viagens entre todas as jornadas. Ou seja, o *K-Swap* busca minimizar o custo total da solução através da redistribuição de blocos de viagens entre os motoristas durante K camadas. A ideia deste procedimento é que eventuais trocas de trechos entre as jornadas podem melhorar a solução através da redução de tempos ociosos, eliminação de horas extras, ou melhor programação dos intervalos de folga.

O *K-Swap* realiza um corte entre duas camadas, como no PCR, depois percorre K camadas e realiza um novo corte entre camadas. O trecho entre os dois cortes, denominado bloco de viagens, é a parte que poderá ser trocada entre as jornadas. Com isso, o *K-Swap* pode assumir blocos de viagens de diferentes tamanhos, dependendo apenas do valor do índice K . De posse da variável K , é calculado o custo de se associar cada um dos n blocos de viagens com cada uma das jornadas. Assim, na matriz $E^k = [e_{ij}^k]$, referente ao procedimento *K-Swap*, e_{ij}^k é dado por $h(i, j)$, que equivale ao custo de associar o bloco de viagens j com a jornada i , caso possível, incluindo possíveis penalidades. Assim como no PCR, se a associação for impossível, e_{ij}^k recebe custo infinito, como ilustrado na Figura 9.

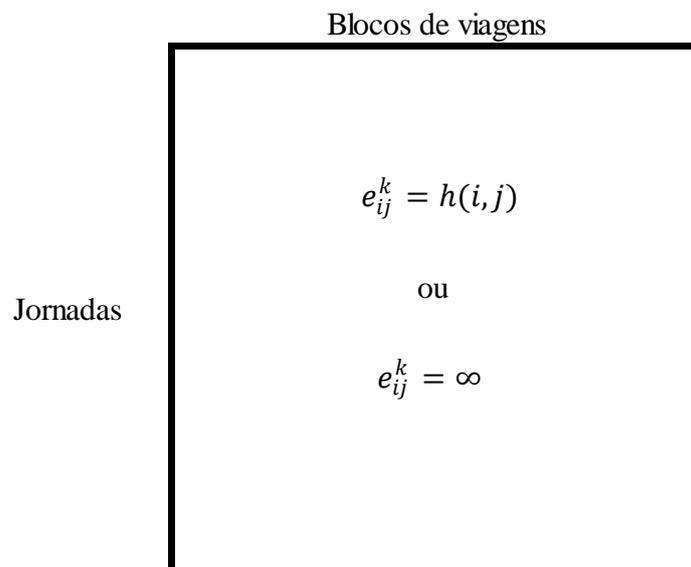


Figura 10: Estrutura da matriz de custos na execução do K-Swap

Na Figura 10 estão representadas, em linhas pontilhadas, as possibilidades de recombinação do *1-Swap*, *K-Swap* tomando K igual a 1, ao resolver a Camada 4. Como no exemplo da

Figura 10 o valor de K é igual a 1, o procedimento verifica a possibilidade de troca por apenas 1 camada. Note que para as Jornadas 1 e 2 não é viável recombinar com a Jornada 4, pois o tempo de folga resultante seria muito baixo, não satisfazendo as restrições. Por outro lado, a Jornada 3 pode se recombinar apenas com o bloco de viagens da Jornada 4, pois o tempo de término de sua última viagem é maior que o tempo de início das viagens das Jornadas 1 e 2 na Camada 4. Além disso, o início da tarefa da Jornada 3 na Camada 5 é um pouco mais tarde, já sendo suficiente para satisfazer o tempo mínimo de folga. Por sua vez, a Jornada 4 não pode se recombinar com nenhum outro bloco de viagens, pois o término de sua última viagem é maior que o tempo de início de todas as outras viagens na Camada 4.

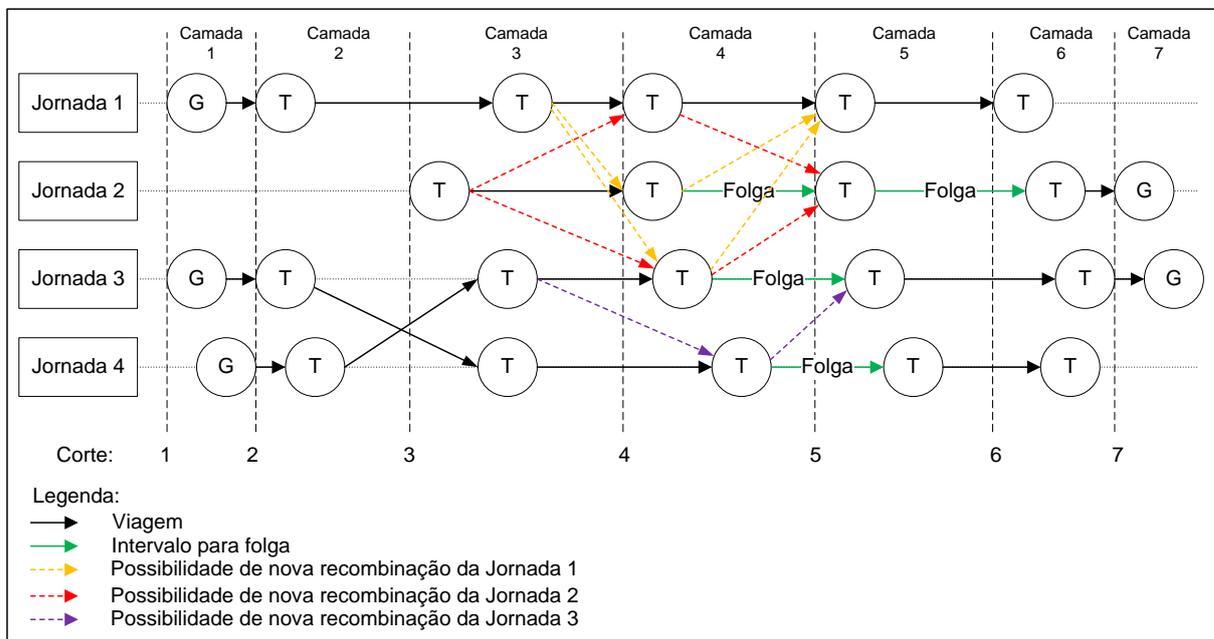


Figura 11: Exemplo de possíveis recombinações do 1-Swap ao resolver camada 4

Construída a matriz de custos E , o PA para a camada correspondente é resolvido e com base na solução é feita a recombinação dos blocos de viagens, que no caso contém apenas uma viagem devido ao fato de K ser igual a um. Aqui, diferentemente do PCR, o procedimento é executado normalmente antes de todas as camadas, sem exceções, pois a troca entre os trechos antes dos cortes pode alterar a solução em qualquer corte. Um exemplo da recombinação proporcionada pelo K -Swap, tomando K igual a 1, é apresentado na Figura 11, onde foi realizada uma troca entre os trechos das Jornadas 1 e 2, permitindo que fosse atribuído um intervalo de folga à Jornada 1.

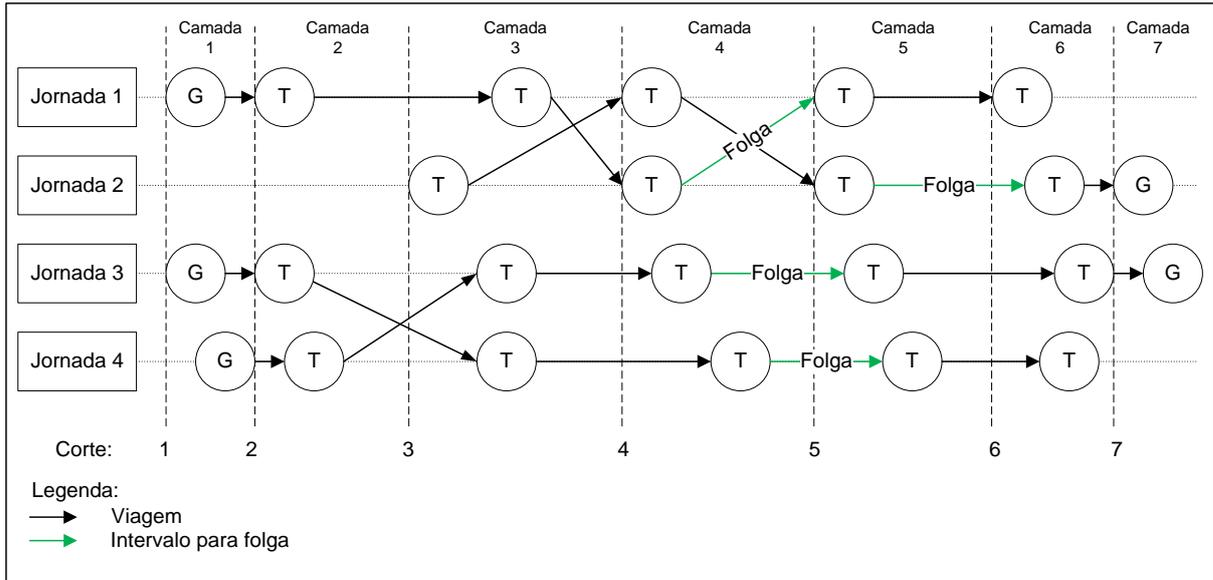


Figura 12: Exemplo de recombinação após execução do 1-Swap na camada 4

O esquema que descreve o funcionamento do procedimento *K-Swap* é apresentado na Figura 12, considerando o índice K do procedimento *K-Swap* igual a K^* . Observe que o laço de repetição pode não se estender até a última camada k , dependendo do valor de K^* . Dessa forma, o procedimento é realizado até que a soma do índice da camada em questão, a , mais o valor de K^* seja maior que o número total de camadas k .

Procedimento *K-Swap* (K^* : inteiro)

Início;

Inicialize os dados;

Para $a = 1$ **até** o número total de camadas $k + (K^* - 1)$ **faça**:

Efetue corte antes da camada a ;

Gere a matriz de custos E , considerando o bloco de atividades de tamanho igual ao índice K^* , que tem início na camada a ;

Resolva o PA da matriz E ;

Faça a recombinação das jornadas com os blocos de atividades conforme o resultado obtido;

Fim.

Figura 13: Passos para a execução do procedimento *K-Swap*

Após a resolução dos dois procedimentos de melhoria, obtém-se uma nova solução, conforme apresentado na Figura 13.

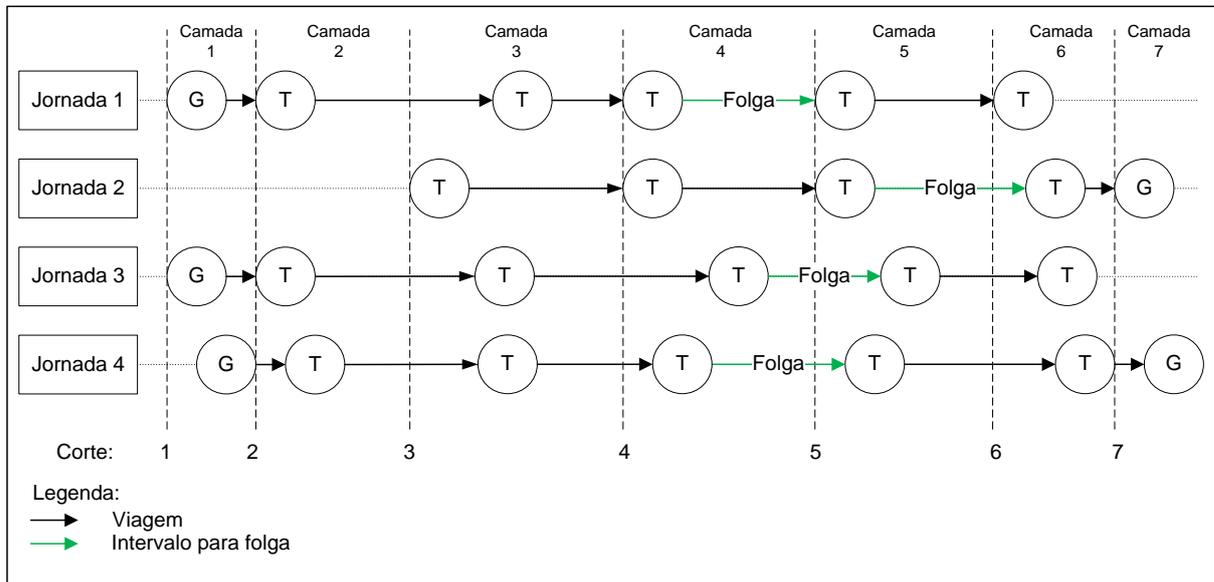


Figura 14: Apresentação da solução melhorada

3.2.4 Descrição geral dos algoritmos

O algoritmo apresentado trabalha em duas fases: construção da solução inicial e melhoramento. Na primeira fase, ocorre a construção de um grafo multipartido, que representa as jornadas dos motoristas. Na segunda fase, procura-se melhorar a solução inicialmente obtida, através de recombinações entre trechos das jornadas. Para isso, são utilizados dois procedimentos, o PCR e o *K-Swap*, sendo considerados quatro diferentes índices K : 1, 2, 3 e 4. Dessa forma, são propostas cinco vizinhanças de busca, baseadas em diferentes estratégias de recombinações: PCR, *1-Swap*, *2-Swap*, *3-Swap* e *4-Swap*.

Para a resolução da fase de melhoramento são consideradas duas abordagens. A primeira utiliza a ideia da metodologia VND (*Variable Neighborhood Descent*), uma variação do VNS (*Variable Neighborhood Search*), que explora o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança, apresentado por Hansen e Mladenovic (2001). Sendo assim, na abordagem que considera o VND são realizadas trocas entre os procedimentos de melhoria quando um procedimento não conseguir melhorar a solução atual. Essas trocas são realizadas de forma cíclica até que o critério de parada seja alcançado.

A outra abordagem considerada para a resolução da fase de melhoramento consiste em realizar a troca de procedimentos de melhoria de forma contínua, independentemente da solução encontrada. Assim, após a execução de um procedimento sempre será executado um procedimento diferente.

Sendo assim, podem ser definidos dois algoritmos gerais propostos para a resolução do PEM, um considerando o VND como estratégia de troca entre os procedimentos, apresentado no quadro da Figura 14, e outro realizando a troca contínua entre os procedimentos, apresentado no quadro da Figura 15.

As duas abordagens apresentadas para o algoritmo proposto seguem os mesmos critérios de parada, a saber:

- Execução de 30 procedimentos/iterações, ou;
- Execução de todos os cinco diferentes procedimentos sem a ocorrência de quaisquer melhorias na solução.

Algoritmo geral considerando a troca de procedimentos por VND

Início;

Inicialize os dados;

Construa Solução Inicial;

$s^* \leftarrow s$;

$x \leftarrow 1$;

Enquanto (*Critério de parada não atingido*) **faça**

Enquanto ($x \leq V$) **faça**

Execute procedimento $N^{(x)}(s)$;

Se ($f(s) < f(s^*)$) **então**

$s^* \leftarrow s$

Senão

$x \leftarrow x + 1$;

$x \leftarrow 1$; {Recomeça com a primeira vizinhança}

Fim.

Figura 15: Descrição resumida do algoritmo proposto considerando a troca de um procedimento quando este não realizar uma melhoria na solução

Algoritmo geral considerando a troca de procedimentos a cada iteração

Início;

Inicialize os dados;

Construa Solução Inicial;

$s^* \leftarrow s$;

$x \leftarrow 1$;

Enquanto (*Critério de parada não atingido*) **faça**

Enquanto ($x \leq V$) **faça**

Execute procedimento $N^{(x)}(s)$;

$x \leftarrow x + 1$;

Se ($f(s) < f(s^*)$) **então**

$s^* \leftarrow s$

$x \leftarrow 1$; {Recomeça com a primeira vizinhança}

Fim.

Figura 16: Descrição resumida do algoritmo proposto considerando a troca de procedimentos a cada iteração

Além de considerar duas diferentes abordagens para a resolução da fase de melhoramento, também podem ser configuradas diferentes versões do algoritmo alterando a ordem de execução dos procedimentos de melhoria. Considerando todas estas possibilidades, diferentes versões do algoritmo podem ser configuradas considerando escolhas em relação a:

- Função de custo $f1$ ou $f2$ na construção da solução inicial;
- Sequência de execução dos procedimentos de melhoramento;
- Sequência de variação do índice K para o procedimento K-Swap, podendo variar no sentido crescente ou decrescente;
- Utilização ou não da metodologia VND para realizar a troca entre os procedimentos de melhoramento.

Para realizar testes comparativos entre as diferentes possibilidades de configuração e com o intuito de facilitar o entendimento das diferentes configurações apresentados, foi definido um formato geral de nomenclatura, apresentada na Figura 16, que considera todas as possibilidades de configurações.

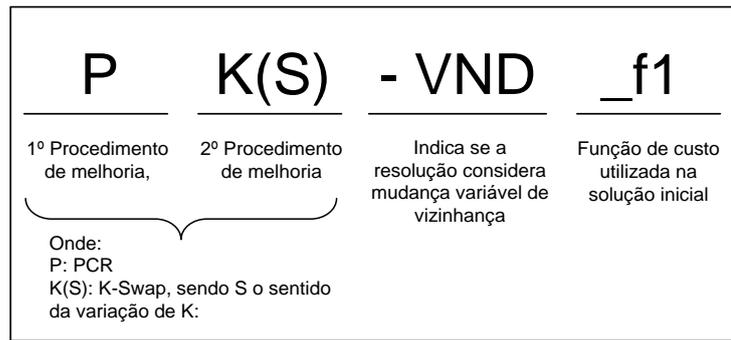


Figura 17: Modelo geral de nomenclatura utilizada para as diferentes variações do algoritmo

Dependendo da variação do índice K , o sentido S pode assumir dois valores:

- $S = C$, quando o sentido de K é crescente ($K = 1, 2, 3, 4$);
- $S = D$, quando o sentido de K é decrescente ($K = 4, 3, 2, 1$).

De forma resumida, as Tabelas 5 e 6 apresentam todas as variações do algoritmo consideradas, destacando para cada teste informações como qual procedimento de melhoria é considerado em cada vizinhança e se este considera o VND para realizar trocas entre os procedimentos.

Tabela 5: Resumo das variações dos algoritmos considerando a função f1 na resolução inicial

Teste	N ⁽¹⁾ (s)	N ⁽²⁾ (s)	N ⁽³⁾ (s)	N ⁽⁴⁾ (s)	N ⁽⁵⁾ (s)	Considera VND?
PK(C)-VND_f1	PCR	1-Swap	2-Swap	3-Swap	4-Swap	SIM
PK(D)-VND_f1	PCR	4-Swap	3-Swap	2-Swap	1-Swap	SIM
K(C)P-VND_f1	1-Swap	2-Swap	3-Swap	4-Swap	PCR	SIM
K(D)P-VND_f1	4-Swap	3-Swap	2-Swap	1-Swap	PCR	SIM
PK(C)_f1	PCR	1-Swap	2-Swap	3-Swap	4-Swap	NÃO
PK(D)_f1	PCR	4-Swap	3-Swap	2-Swap	1-Swap	NÃO
K(C)P_f1	1-Swap	2-Swap	3-Swap	4-Swap	PCR	NÃO
K(D)P_f1	4-Swap	3-Swap	2-Swap	1-Swap	PCR	NÃO

Tabela 6: Resumo das variações dos algoritmos considerando a função f2 na resolução inicial

Teste	N ⁽¹⁾ (s)	N ⁽²⁾ (s)	N ⁽³⁾ (s)	N ⁽⁴⁾ (s)	N ⁽⁵⁾ (s)	Considera VND?
PK(C)-VND_f2	PCR	1-Swap	2-Swap	3-Swap	4-Swap	SIM
PK(D)-VND_f2	PCR	4-Swap	3-Swap	2-Swap	1-Swap	SIM
K(C)P-VND_f2	1-Swap	2-Swap	3-Swap	4-Swap	PCR	SIM
K(D)P-VND_f2	4-Swap	3-Swap	2-Swap	1-Swap	PCR	SIM
PK(C)_f2	PCR	1-Swap	2-Swap	3-Swap	4-Swap	NÃO
PK(D)_f2	PCR	4-Swap	3-Swap	2-Swap	1-Swap	NÃO
K(C)P_f2	1-Swap	2-Swap	3-Swap	4-Swap	PCR	NÃO
K(D)P_f2	4-Swap	3-Swap	2-Swap	1-Swap	PCR	NÃO

4 RESULTADOS

4.1 Considerações iniciais

Neste capítulo são apresentados os resultados computacionais obtidos pelas diferentes configurações apresentadas para os algoritmos propostos, utilizando para testes as mesmas instâncias que Calvi (2005). A apresentação dos resultados inclui uma análise comparativa entre os melhores resultados dos algoritmos propostos e os melhores resultados encontrados por Calvi (2005). A comparação é realizada considerando o número de jornadas necessárias, o custo total de solução e o tempo de execução.

Também são apresentadas algumas comparações entre as diferentes configurações dos algoritmos e seu impacto na solução final. Além disso, são comparados os desempenhos de cada um dos procedimentos utilizados isoladamente e, por fim, é apresentada uma relação entre os tempos computacionais necessários para resolução e o número de viagens que caracteriza cada um dos problemas.

4.2 Implementação

Os algoritmos propostos foram implementados na linguagem Pascal, utilizando a ferramenta de programação de código aberto Lazarus. Os experimentos foram realizados em uma máquina Dell Inspiron com um processador Intel Core 2 Duo de 2,53 GHz, 4 GB de memória RAM e rodando o sistema operacional Windows 7.

4.3 Resultados obtidos

Os resultados obtidos pelos algoritmos propostos são apresentados nas Tabelas 7 e 8. A Tabela 7 apresenta os resultados para as oito diferentes configurações testadas utilizando a função $f1$ na construção da solução inicial, enquanto a Tabela 8 apresenta os resultados utilizando a função $f2$. Nas tabelas são apresentadas todas as instâncias e para cada configuração testada são apresentados o número de jornadas obtidas, o custo em minutos pagos e o tempo de execução. Em cada tabela são destacados os melhores resultados encontrados para cada instância.

Tabela 7: Resultados encontrados considerando a função f1 na resolução inicial

Instância	PK(C)-VND_f1			PK(D)-VND_f1		
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Nº Jorn.	Custo (min)	Tempo (mm:ss)
AL130	19	8.513,29	0:21	19	8.540,00	0:28
AL251	39	17.205,00	0:39	39	17.239,14	0:26
RE412	66	29.527,50	0:57	67	29.832,50	0:52
AL512	77	34.390,00	1:36	76	33.830,71	1:27
AL761	113	49.930,14	2:57	113	49.892,50	2:22
AL1000	148	65.607,50	4:10	151	66.695,00	3:50
AL1253	193	86.000,86	7:24	192	84.751,71	6:31
AL1517	232	102.488,50	9:19	230	101.760,07	10:5
AL2010	303	133.755,00	16:7	303	133.737,57	16:24
RE2313	341	152.099,00	24:27	341	150.981,21	23:48
Instância	K(C)P-VND_f1			K(D)P-VND_f1		
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Nº Jorn.	Custo (min)	Tempo (mm:ss)
AL130	19	8.419,79	0:25	19	8.513,14	0:17
AL251	39	17.242,50	0:33	39	17.242,50	0:40
RE412	67	29.922,50	0:54	70	31.137,50	0:51
AL512	80	35.290,00	1:25	78	34.522,50	1:16
AL761	113	49.915,00	2:22	113	49.787,50	2:22
AL1000	150	66.120,00	4:54	151	66.763,07	5:33
AL1253	190	84.365,57	7:26	193	85.093,07	7:12
AL1517	232	102.252,50	11:18	230	101.655,07	10:13
AL2010	303	133.739,50	16:14	306	134.922,00	15:33
RE2313	345	152.562,21	23:20	341	151.370,93	22:55
Instância	PK(C)_f1			PK(D)_f1		
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Nº Jorn.	Custo (min)	Tempo (mm:ss)
AL130	19	8.513,29	0:20	19	8.525,00	0:16
AL251	39	17.205,00	0:32	39	17.220,00	0:45
RE412	69	30.592,50	0:31	65	29.260,00	0:49
AL512	79	34.902,50	1:10	77	34.247,50	1:12
AL761	112	49.355,00	2:33	112	49.460,00	2:22
AL1000	146	64.826,43	3:54	149	65.830,00	4:38
AL1253	188	83.508,64	6:19	190	84.069,14	6:13
AL1517	226	100.130,43	8:58	225	99.522,00	9:26
AL2010	298	132.190,07	17:20	299	132.277,57	17:24
RE2313	337	149.454,21	20:47	342	151.149,50	21:36
Instância	K(C)P_f1			K(D)P_f1		
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Nº Jorn.	Custo (min)	Tempo (mm:ss)
AL130	19	8.404,79	0:36	19	8.405,00	0:21
AL251	39	17.197,50	0:43	39	17.190,00	0:42
RE412	67	29.802,50	1:32	66	29.400,00	0:53
AL512	77	34.127,50	1:34	78	34.560,00	1:16
AL761	113	49.765,00	2:20	113	49.795,43	2:45
AL1000	150	66.180,00	4:11	146	64.638,07	4:23
AL1253	191	84.370,00	6:43	193	85.070,00	6:43
AL1517	230	101.467,00	9:42	228	100.827,57	9:41
AL2010	302	133.297,57	19:90	296	131.055,07	15:19
RE2313	342	151.140,20	23:51	342	151.143,14	23:47

Tabela 8: Resultados encontrados considerando a função f2 na resolução inicial

Instância	PK(C)-VND_f2			PK(D)-VND_f2		
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Nº Jorn.	Custo (min)	Tempo (mm:ss)
AL130	19	8.464,07	0:20	19	8.404,07	0:24
AL251	40	17.742,50	0:35	40	17.615,00	0:34
RE412	65	29.312,50	1:05	65	29.327,50	0:48
AL512	77	34.592,50	1:06	78	34.732,50	0:60
AL761	111	49.509,43	1:56	109	48.554,14	1:57
AL1000	146	65.341,86	6:10	146	65.486,07	4:06
AL1253	184	82.702,36	10:02	184	82.085,57	7:49
AL1517	221	99.617,50	17:03	222	99.194,36	8:24
AL2010	292	130.734,71	15:31	292	129.600,14	15:60
RE2313	330	152.200,21	27:44	330	152.200,21	23:41
Instância	K(C)P-VND_f2			K(D)P-VND_f2		
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Nº Jorn.	Custo (min)	Tempo (mm:ss)
AL130	19	8.446,14	0:28	19	8.462,57	0:16
AL251	40	17.630,00	0:50	40	17.690,00	0:22
RE412	66	29.520,00	0:56	67	29.997,50	0:46
AL512	78	34.748,64	0:53	77	34.227,43	0:56
AL761	107	48.251,26	2:48	110	48.801,64	2:28
AL1000	146	64.998,07	4:54	145	64.933,64	4:22
AL1253	196	92.593,00	6:30	188	83.329,21	6:44
AL1517	226	100.998,36	12:13	233	103.162,50	10:51
AL2010	322	145.323,43	16:09	290	129.422,29	17:49
RE2313	349	159.902,36	28:54	331	147.425,64	25:23
Instância	PK(C)_f2			PK(D)_f2		
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Nº Jorn.	Custo (min)	Tempo (mm:ss)
AL130	19	8.510,00	0:12	19	8.400,93	0:18
AL251	40	17.600,00	0:19	40	17.600,00	0:21
RE412	67	29.780,00	0:40	67	29.885,00	0:38
AL512	77	34.292,50	1:09	77	34.292,50	2:09
AL761	108	47.932,29	1:57	108	48.058,64	3:06
AL1000	145	64.555,29	4:01	145	64.475,00	5:01
AL1253	182	81.070,71	6:23	184	81.688,64	7:45
AL1517	222	98.512,36	11:14	222	98.667,00	12:09
AL2010	289	128.367,50	19:20	289	128.217,50	13:51
RE2313	329	146.505,00	27:34	329	146.511,14	22:22
Instância	K(C)P_f2			K(D)P_f2		
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Njor	Custo	Tempo
AL130	19	8.505,29	0:22	19	8.425,43	0:15
AL251	40	17.600,00	0:18	40	17.607,50	0:20
RE412	67	29.930,00	0:36	67	29.885,00	0:33
AL512	76	33.822,50	1:28	76	34.064,79	0:58
AL761	109	48.395,00	2:18	109	48.321,14	2:35
AL1000	144	64.097,29	5:17	145	64.392,50	6:12
AL1253	182	81.071,14	6:55	182	81.047,50	8:45
AL1517	222	98.765,07	10:16	221	98.283,21	8:52
AL2010	289	128.342,00	13:37	285	126.747,71	19:27
RE2313	331	147.212,64	27:49	327	145.701,43	22:30

Na Tabela 9 são apresentados, da segunda a quarta coluna, os melhores resultados encontrados pelos algoritmos propostos e na quinta coluna a configuração que obteve tal resultado. Da sexta até a oitava coluna são apresentados os melhores resultados de Calvi (2005). Por fim, na última coluna é apresentada uma comparação entre os melhores resultados obtidos pelos algoritmos propostos e os melhores resultados encontrados por Calvi (2005), sendo estes valores encontrados pela fórmula:

$$Gap = \frac{\text{Custo algoritmo proposto} - \text{Custo Calvi (2005)}}{\text{Custo Calvi (2005)}} \cdot 100\% \quad (7)$$

Tabela 9: Comparação dos melhores resultados com os resultados de Calvi (2005)

Instância	Melhores resultados encontrados				Melhores resultados Calvi (2005)			GAP Custo
	Nº Jorn.	Custo (min)	Tempo (mm:ss)	Método	Nº Jorn.	Custo (min)	Tempo (mm:ss)	
AL130	19	8.400,93	00:18	PK(D)_F2	19	8.389,40	00:45	0,137%
AL251	39	17.190,00	00:42	K(D)P_F1	40	17.600,00	00:35	-2,330%
RE412	65	29.260,00	00:49	PK(D)_F1	66	29.512,50	00:48	-0,856%
AL512	76	33.822,50	01:28	K(C)P_F2	79	35.105,00	00:40	-3,653%
AL761	108	47.932,29	01:57	PK(C)_F2	107	47.532,90	01:50	0,840%
AL1000	144	64.097,29	05:17	K(C)P_F2	146	64.873,60	05:00	-1,197%
AL1253	182	81.047,50	08:45	K(D)P_F2	187	82.842,90	03:36	-2,167%
AL1517	221	98.283,21	08:52	K(D)P_F2	225	99.852,80	05:29	-1,572%
AL2010	285	126.747,71	19:27	K(D)P_F2	290	128.964,20	07:00	-1,719%
RE2313	327	145.701,43	22:30	K(D)P_F2	331	147.215,00	04:37	-1,028%

O gráfico da Figura 17 apresenta, para todas as instâncias, a diferença entre os custos encontrados pelos algoritmos propostos e os menores custos encontrados em Calvi (2005), assim como o tempo de resolução para cada instância.

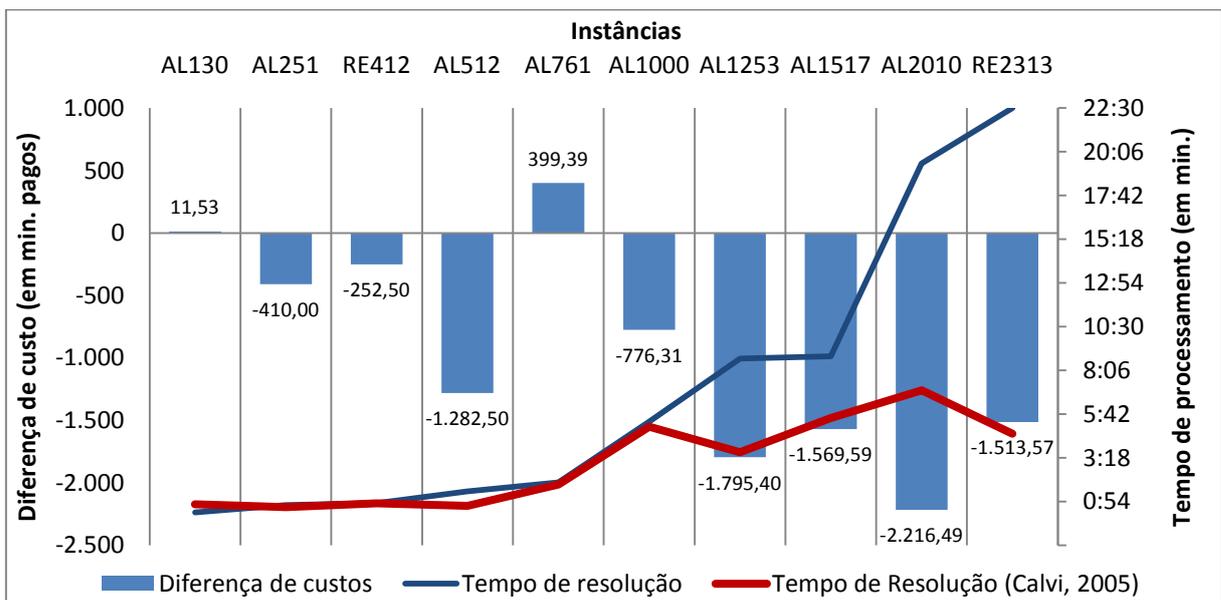


Figura 18: Diferença entre custos x tempo de processamento

Com as informações da Tabela 9 e da Figura 17 é possível constatar que os algoritmos propostos conseguiram encontrar soluções com custos menores para a maioria das instâncias, principalmente devido ao fato de ter encontrado soluções com menor número de jornadas, exceção feita às instâncias AL130 e AL761. Com relação ao tempo de processamento, pode-se observar que com o crescimento do tamanho do problema, há também um crescimento da diferença entre os tempos de processamento, sendo que os algoritmos propostos apresentam o maior tempo.

Observando os métodos que encontraram os melhores resultados, é possível constatar duas ocorrências principais:

- A maioria dos melhores resultados foi obtida utilizando a função $f2$ na construção da solução inicial;
- Dentre os melhores resultados, nenhum foi obtido por um algoritmo que considerasse a técnica VND.

Para investigar a primeira constatação e os impactos que a resolução do problema sofre pela utilização das funções $f1$ e $f2$, foi gerado um gráfico de custo por iterações para a instância RE2313 utilizando os algoritmos K(D)P_f1 e K(D)P_f2, apresentados nas figuras 18 e 19.

No início da resolução, nem todas as restrições são satisfeitas, então para transformar estas restrições em custos, foi estabelecido um custo de penalidade Pen com valor igual ao valor de uma nova jornada, ou seja, 440 minutos pagos, que correspondem às 7 horas e 20 minutos que é o tempo mínimo pago por uma jornada normal.

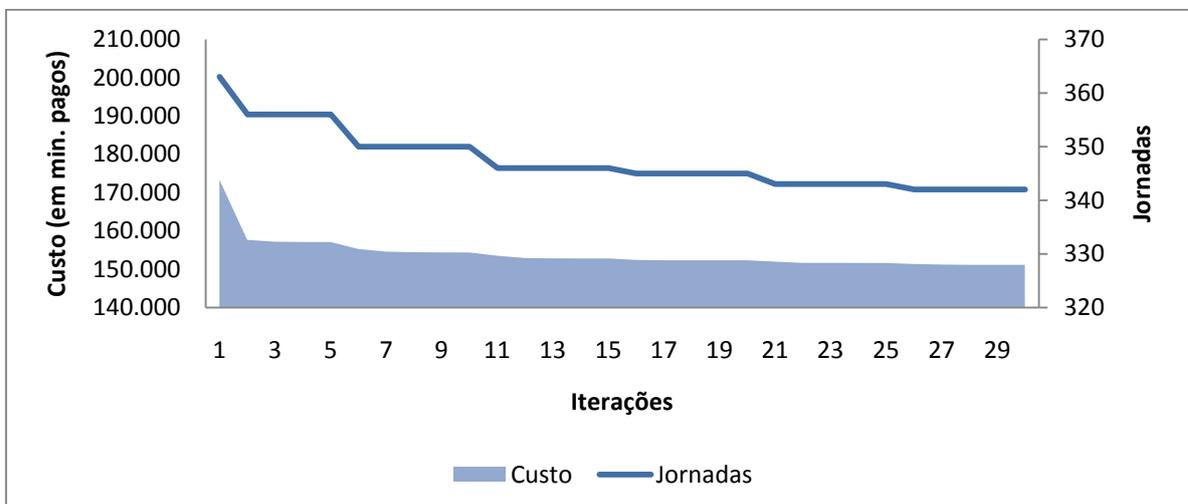


Figura 19: Gráfico de custo x iterações para o método K(D)P_f1

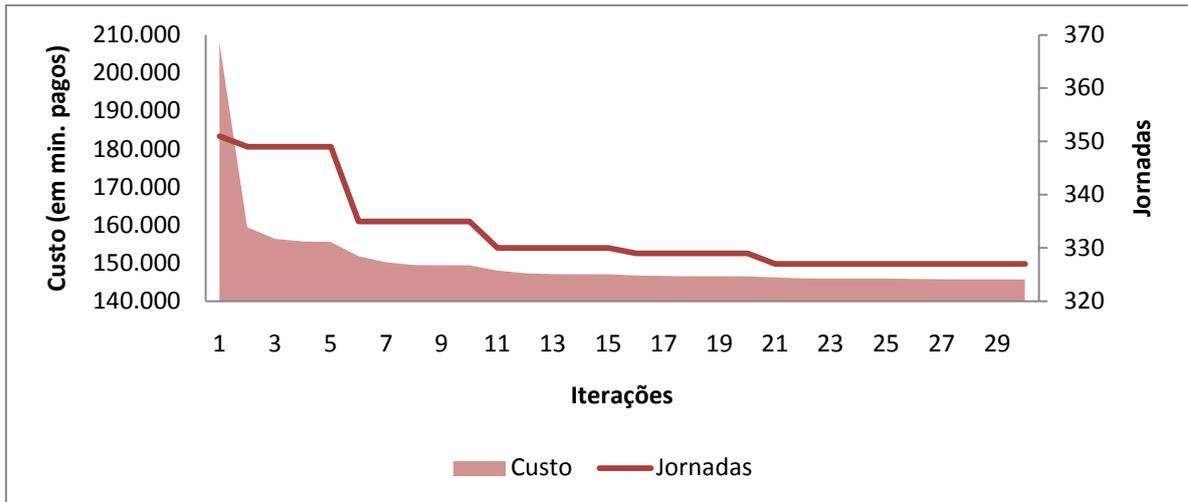


Figura 20: Gráfico de custo x iterações para o método K(D)P_f2

Pela comparação dos gráficos das Figuras 18 e 19, fica claro que pelo fato de não considerar custos de horas extras, a utilização da função $f2$ na construção da solução inicial permite encontrar uma solução com maior custo, porém com menor número de jornadas. Por outro lado, ao utilizar a função $f1$ encontra-se uma solução de custo menor, porém com número de jornadas maior. Com isso, o fato das melhores soluções para os problemas das maiores instâncias terem sido encontrados pelos algoritmos que utilizam a função $f2$ pode ser justificado pelo fato da solução inicial apresentar um número significativamente menor de jornadas, facilitando a resolução pelos procedimentos de melhoramento.

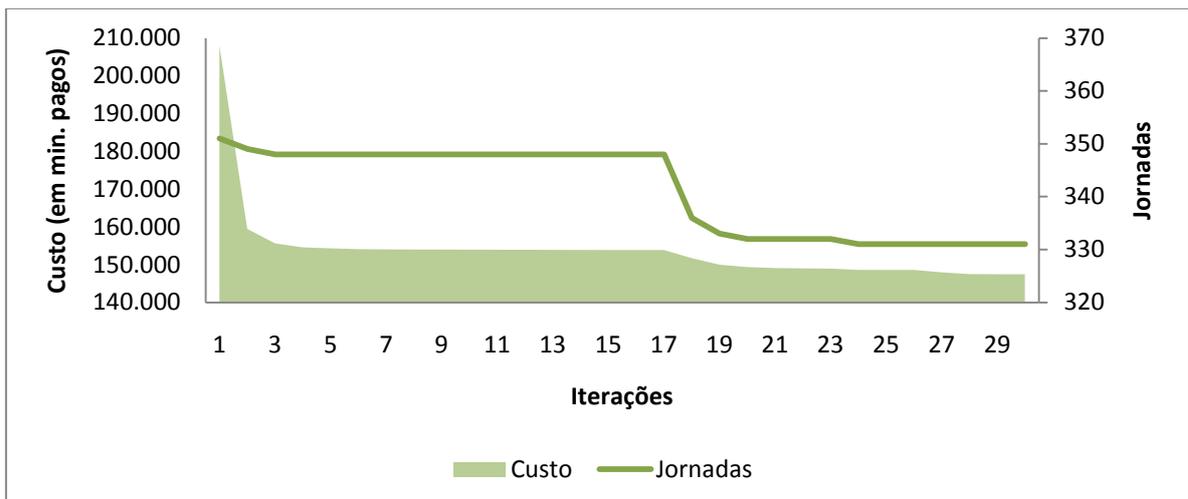


Figura 21: Gráfico de custo x iterações para o método K(D)P-VND_f2

Para investigar o motivo pelo qual os algoritmos que utilizam a técnica VND não terem produzido nenhuma das melhores soluções, foi gerado um gráfico de custo por iterações para

a instância RE2313, utilizando o algoritmo K(D)P-VND_f2. Este gráfico é apresentado na Figura 20, e pode ser comparado com o gráfico da Figura 19 que corresponde à resolução pelo algoritmo K(D)P_f2, que considera as mesmas configurações, porém sem a utilização da técnica VND.

Assim, comparando os dois gráficos nota-se que o que acontece é que pelo fato de só ocorrer troca do procedimento de melhoria quando o procedimento em execução não conseguir mais realizar melhorias, pode-se ficar muito tempo no mesmo procedimento realizando apenas pequenas melhorias, o que aumenta o tempo de convergência do algoritmo. Neste caso, pode ser que com um limite de iterações maior seja alcançado melhor resultado.

Outra questão observada na Tabela 9 refere-se ao tempo de processamento do algoritmo proposto que foi muito maior que o de Calvi (2005) para as maiores instâncias. Apesar de para este tipo de problema o custo ser muito mais importante que o tempo de execução, para investigar melhor a questão do tempo foi elaborada a Tabela 10, que apresenta o número de jornadas, o custo, a iteração e o tempo de execução do algoritmo no momento em que um custo menor que o de Calvi (2005) é encontrado.

Tabela 10: Tempo de execução quando um custo menor que o de Calvi (2005) é encontrado

Instância	Melhores resultados encontrados					Melhores resultados Calvi (2005)		
	Nº Jorn.	Custo (min)	Iteração	Tempo (mm:ss)	Método	Nº Jorn.	Custo (min)	Tempo (mm:ss)
AL130	19	8.400,93	-	-	PK(D)_F2	19	8.389,40	00:45
AL251	39	17.347,50	2	0:10	K(D)P_F1	40	17.600,00	00:35
RE412	65	29.470,00	27	0:45	PK(D)_F1	66	29.512,50	00:48
AL512	76	34.673,00	18	0:53	K(C)P_F2	79	35.105,00	00:40
AL761	108	47.932,29	-	-	PK(C)_F2	107	47.532,90	01:50
AL1000	144	64.802,50	20	3:32	K(C)P_F2	146	64.873,60	05:00
AL1253	182	82.711,36	8	2:20	K(D)P_F2	187	82.842,90	03:36
AL1517	221	99.498,36	12	3:33	K(D)P_F2	225	99.852,80	05:29
AL2010	285	128.846,64	12	7:47	K(D)P_F2	290	128.964,20	07:00
RE2313	327	147.100,86	13	9:45	K(D)P_F2	331	147.215,00	04:37

Pela análise da Tabela 10, é possível perceber que resultados melhores que os de Calvi (2005) são encontrados bem antes do tempo total de execução do algoritmo proposto e, em alguns casos, até mesmo com um tempo menor que o de Calvi (2005). Entretanto, pelo fato dos algoritmos terem sido executados em máquinas diferentes não é justo fazer tal comparação com relação ao tempo, visto que as máquinas atuais possuem desempenho superior.

4.4 Comparação entre os procedimentos de melhoramento

Para comparar os procedimentos de melhoria, foi gerada uma solução inicial utilizando a função f_2 para cada uma das instâncias. A partir disso, os diferentes procedimentos de melhoria foram empregados isoladamente seguindo os mesmos critérios de parada definidos anteriormente. No gráfico da Figura 21, é possível visualizar a redução percentual dos custos que cada procedimento foi capaz de propiciar a uma mesma solução inicial obtida. O gráfico também apresenta a redução alcançada pela combinação dos procedimentos nos algoritmos que obtiveram a melhor solução.

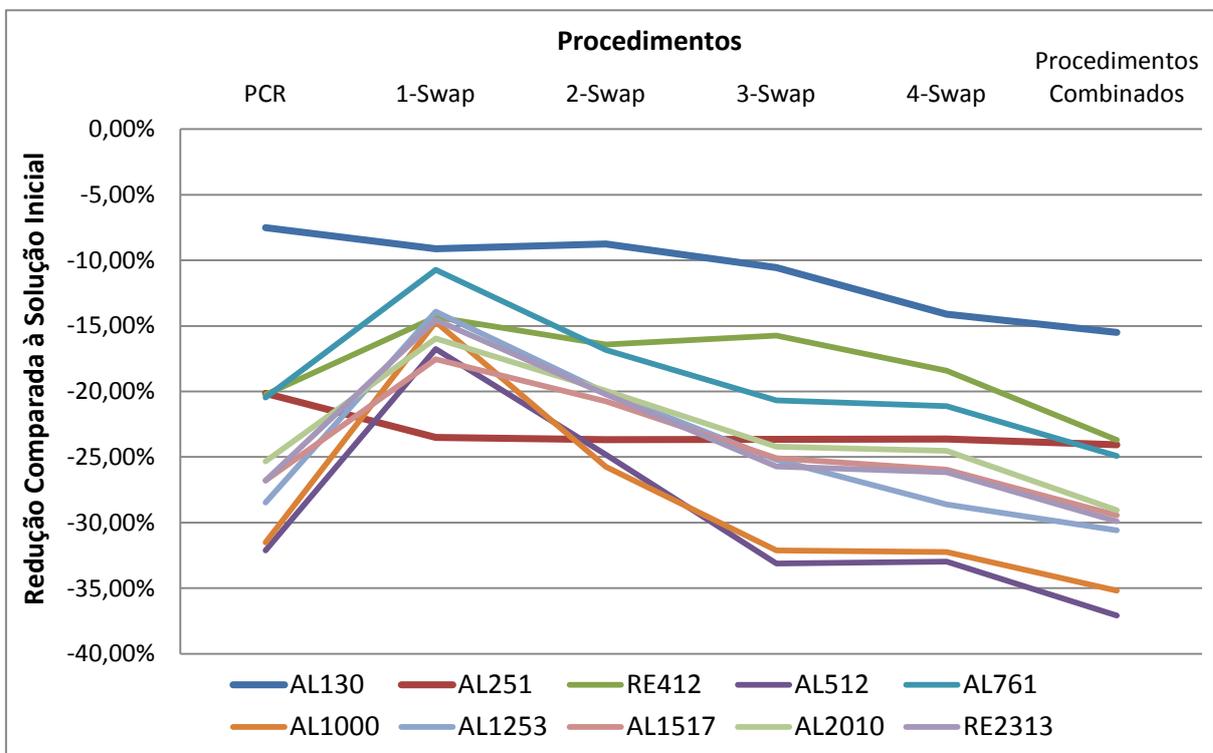


Figura 22: Redução percentual considerando os diferentes procedimentos de maneira isolada comparados com a melhor solução obtida

Pela análise do gráfico nota-se que nenhum procedimento executado de forma isolada conseguiu atingir a mesma redução que a combinação entre eles. Nota-se também, que com exceção das duas menores instâncias, AL130 e AL251, os procedimentos que proporcionaram maiores reduções nos custos foram o PCR e o 4-Swap. Para a instância AL130 o procedimento que proporcionou maior redução foi o 4-Swap. Já para a instância AL251, com exceção do PCR, os procedimentos obtiveram resultado muito próximo, próximos inclusive da melhor solução encontrada.

Também com exceção às instâncias AL130 e AL251, o procedimento que resultou na menor redução de custos foi o 1-Swap, seguido pelo 2-Swap. Por outro lado, para a instância AL130 os procedimentos que menos conseguiram reduzir os custos foram o PCR e o 1-Swap, enquanto para a instância AL251 o procedimento que menos reduziu custos foi o PCR.

Em vista disso, o 4-Swap tem se mostrado mais efetivo individualmente, seguido pelo PCR para as instâncias de maior porte. Por outro lado, o 1-Swap foi o procedimento que isoladamente menos reduziu custos para essas instâncias, enquanto para as instâncias AL130 e AL251 este papel coube ao PCR.

4.5 Relação entre o número de viagens e o tempo de execução

Para exemplificar a complexidade do problema na relação entre o número de viagens e o tempo de execução, o gráfico de dispersão da Figura 22 apresenta para todas as instâncias o tempo de processamento em segundos para a resolução do problema, considerando os algoritmos que obtiveram os melhores resultados.

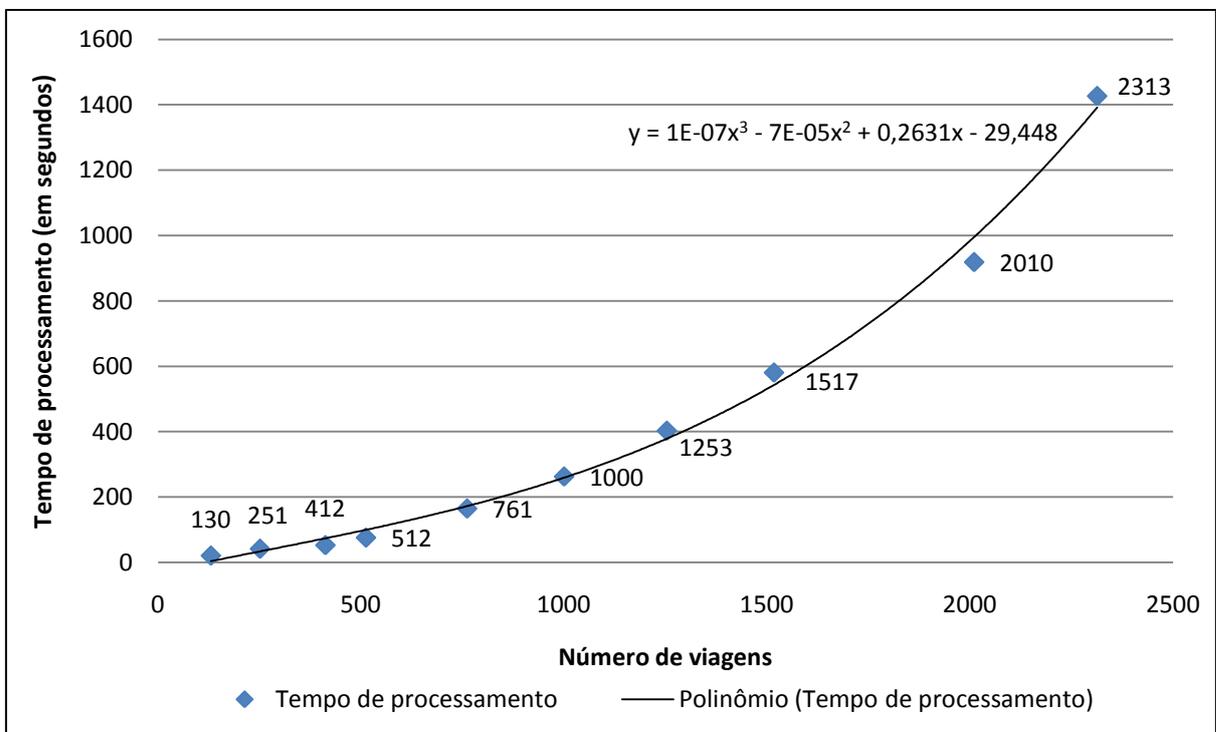


Figura 23: Tempo de processamento x Número de viagens

Por fim, nota-se que os valores obtidos assumem uma tendência polinomial, quando comparados com a função interpolada no gráfico.

5 CONSIDERAÇÕES FINAIS

A Pesquisa Operacional é uma importante ciência aplicada que possibilita avaliar e encontrar soluções eficazes para problemas de diferentes naturezas, através da introdução de elementos de objetividade e racionalidade nos processos de tomada de decisão, sem desconsiderar os elementos subjetivos inerentes a estes problemas (SOBRAPO, 2011). Com isso, a utilização das técnicas da PO permite analisar situações complexas, possibilitando aos gestores tomar decisões mais eficazes.

O trabalho apresenta dois algoritmos com o intuito de otimizar a solução do problema de escalonamento de motoristas de ônibus, uma das etapas mais importantes do processo de planejamento do transporte público, devido ao fato dos custos com pessoal representarem a maior parte dos custos operacionais nas empresas de transporte coletivo (Santos, 2008). Para os algoritmos propostos, são realizados testes com diferentes configurações, variando a função de custo utilizada, a ordem de execução dos procedimentos de melhoramento e a consideração ou não da estratégia VND na resolução da fase de melhoramento.

Os algoritmos propostos utilizam procedimentos exatos, baseados na resolução de Problemas de Atribuição, para a resolução dos subproblemas. Sendo que para a resolução dos PAs foi implementado o algoritmo de Carpaneto e Toth (1987). A estrutura principal do algoritmo foi adaptada com sucesso do método apresentado em Rizzato *et al.* (2010) para a resolução do problema de escalonamento de enfermeiros e, além disso, novos procedimentos foram desenvolvidos e acrescentados à abordagem, como o *K-Swap*.

As soluções obtidas alcançaram, para a maioria dos casos, resultados computacionais melhores que os utilizados para comparação, de Calvi (2005). Conforme resultados apresentados, os algoritmos propostos conseguiram produzir boas soluções tanto para as instâncias de pequeno porte quanto para as maiores. Para ilustrar tal redução, vale ressaltar que para a instância maior, que possui 2313 viagens, foi obtida uma redução de 331 para 327 motoristas e uma redução de custos de 1513 minutos pagos, ou aproximadamente 25 horas pagas por dia. Para a segunda maior instância, que apresenta 2010 viagens foi possível reduzir o número de jornadas de 290 para 285, causando uma redução de custos de 2216 minutos pagos, o equivalente a aproximadamente 37 horas de trabalho.

Estas reduções de custo obtidas correspondem ao período de apenas um dia de planejamento, sendo assim, isso pode representar uma redução de mais de 700 horas mensais, no caso com 2313 viagens. Outra vantagem, já apresentada no algoritmo de Calvi (2005), é a construção de escalas que atendam às regras operacionais e restrições trabalhistas, o que evita insatisfações e ações trabalhistas relacionadas a estes aspectos. Desta forma, fica evidenciada a importância da PO na resolução deste tipo de problema, por permitir a resolução computacional do problema com maior eficácia e eficiência.

Comparando com a metodologia proposta por Calvi (2005), a principal diferença com o método proposto refere-se ao procedimento de melhoria *K-Swap*, que permite realizar trocas entre pedaços de jornadas, em qualquer parte das jornadas e considerando diferentes tamanhos para os pedaços. Com isso, é possível explorar um espaço de busca maior, com diferentes combinações entre jornadas, o que possibilitou encontrar soluções que apresentassem custos menores.

Neste tipo de problema, que trata de automação do planejamento e não necessita ser resolvido a todo o momento, o fator tempo assume importância secundária, sendo mais importante para o gestor uma solução de qualidade, que proporcione redução de custos do que uma solução rápida. Ainda assim, apesar do tempo computacional necessário para atingir os melhores resultados ter sido elevado quando comparado com os tempos de Calvi (2005), foram obtidas soluções com custos ligeiramente inferiores em tempo similar.

Para trabalhos futuros recomenda-se um estudo mais detalhado dos valores do índice K para o procedimento *K-Swap*, já que de acordo com o gráfico que compara a redução obtida pelos procedimentos de melhoramento quando utilizados isoladamente, as maiores reduções ocorreram justamente quando K assumiu os maiores valores. Com relação às instâncias utilizadas, se propõe que sejam realizados testes com outras instâncias disponíveis na literatura, para que comparação com outros métodos sejam realizadas. Por fim, com relação à abordagem, seria interessante adaptar o algoritmo para investigar a resolução do problema considerando uma abordagem integrada, na qual o escalonamento de veículos e de motoristas seja realizado de forma simultânea.

REFERÊNCIAS

- BAPTISTE, P. Combining Operations Research and Constraint Programming to Solve Real-Life Scheduling Problems. ERCIM (European Community in Information Technology) News, n. 44, jan. 2001. Disponível em: <http://www.ercim.eu/publication/Ercim_News/enw44/baptiste.html>. Acesso em: 14 mai. 2011.
- BLUM, C.; ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys, 35 (3), p. 268-308, 2003. Disponível em: <<http://dl.acm.org/citation.cfm?id=937505>>. Acesso em: 20 nov. 2011.
- CALVI, R. Um Algoritmo para o Problema de Escalonamento de Tripulação em Empresas de Ônibus. 2005. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Maringá, Maringá. Disponível em: <<http://www.din.uem.br/pos-graduacao/mestrado-em-ciencia-da-computacao/dissertacoes>>. Acesso em: 14 mai. 2011.
- CARPANETO, G.; TOTH, P. Primal-dual algorithms for the assignment problem. Discrete Applied Mathematics, 18, p. 137–153, 1987.
- CARRARESI, P.; GALLO, G. A multi-level bottleneck assignment approach to the bus drivers' rostering problem. In: European Journal of Operational Research, 16, p. 163-173, 1984.
- CHAVES, A. A. Modelagens exata e heurística para resolução do problema do caixeiro viajante com coleta de prêmios. 2003. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal de Ouro Preto, Minas Gerais. Disponível em: <<http://www.decom.ufop.br/prof/marcone/Orientacoes/PCVCP-Exato-VNS.pdf>>. Acesso em: 20 nov. 2011.
- CONSTANTINO, A. A.; NETTO, C. A. S.; ARAÚJO, S. A. Problema de Escalonamento de Pessoal em Centrais de Atendimento Telefônico. In: XXXVIII SBPO, 2006, Goiânia. Anais do XXXVIII SBPO. Disponível em: <<http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/EscalonamentoCentraisTelefonicas.pdf>>. Acesso em: 14 mai. 2011.
- CORMEN, T. H. *et al.* Algoritmos: Teoria e Prática. 2. ed. Rio de Janeiro: Elsevier, 2002. 916p.
- DIAS, T. G.; SOUSA, J. P.; CUNHA, J. F. Genetic algorithms for the bus driver scheduling problem: a case study. Journal of the Operational Research Society, 53, p. 1-12, 2002. Disponível em: <http://www.fe.up.pt/si/publs_pesquisa.formview?p_id=5225>. Acesso em: 14 mai. 2011.
- ERNST, A. T.; JIANG, H.; KRISHNAMOORTHY, M.; SIER, D. Staff Scheduling and Rostering: A review of applications, methods and models. In: European Journal of Operational Research, 153, p. 3-27, 2004. Disponível em: <<http://ideas.repec.org/a/eee/ejores/v153y2004i1p3-27.html>>. Acesso em: 14 mai. 2011.

GONÇALVES, T. L. Meta-heurísticas para o Problema de Programação de Tripulações. 2010. Dissertação (Mestrado em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro. Disponível em: <<http://www.cos.ufrj.br/uploadfiles/1269521056.pdf>>. Acesso em: 14 mai. 2011.

HANSEN, P.; MLADENOVIC, N. Variable Neighborhood Search: Principles and applications. In: *European Journal of Operational Research*, v. 130, p. 449-467, 2001. Disponível em: <http://www.elsevier.com/authoried_subject_sections/S03/Anniversary/EJOR_free30.pdf>. Acesso em 25 jul. 2011.

HILLIER, F. S.; LIEBERMAN, G. J. *Introdução à Pesquisa Operacional*. 8. ed. Porto Alegre: AMGH, 2010. 852 p.

HUISMAN, D.; FRELING, R.; WAGELMANS, A. P. M. Multiple-Depot Integrated Vehicle and Crew Scheduling. *Transportation Science*, v. 39, n. 4, p. 491–502, 2005. Disponível em: <<http://transci.journal.informs.org/cgi/content/abstract/39/4/491>>. Acesso em: 25 mar. 2011.

LEONE, R.; FESTA, P.; MARCHITTO, E. A Bus Driver Scheduling Problem: a new mathematical model and a GRASP approximate solution. *Journal of Heuristics*, p. 1-26, 2010. Disponível em: <www.springerlink.com/index/pdf/10.1007/s10732-010-9141-3>. Acesso em: 14 mai. 2011.

LOURENÇO, H. R.; PAIXÃO, J. P.; PORTUGAL, R. Multiobjective Metaheuristics for the Bus Driver Scheduling Problem. *Transportation Science*, v. 35, n. 3, p. 331-343, 2001. Disponível em: <<http://portal.acm.org/citation.cfm?id=769134>>. Acesso em: 14 mai. 2011.

MARINHO, E. H. *et al.* Busca Tabu Aplicada ao Problema de Programação de Tripulações de Ônibus Urbano. In: XXXVI SBPO, 2004, São João del Rei. Anais do XXXVI SBPO. Disponível em: <<http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/PPTviaBT.pdf>>. Acesso em: 24 mar. 2011.

MAURI, G. R.; LORENA, L. A. N. Método Iterativo para Resolução do Problema de Escalonamento de Tripulações. In: XXXVI SBPO, 2004, São João del Rei. Anais do XXXVI SBPO. Disponível em: <<http://www.lac.inpe.br/~lorena/mauri/mauri-lorena-sbpo.pdf>>. Acesso em: 24 mar. 2011.

PENTICO, D. W. Assignment Problems: A golden anniversary survey. In: *European Journal of Operational Research*, 176, p. 774 – 793, 2007. Disponível em: <www.sciencedirect.com>. Acesso em: 19 jun. 2011.

PINEDO, M. L. *Theory, Algorithms and Systems*. 3. ed. Nova York: Prentice Hall, 2008. 678p.

PRATA, B. A. Programação Integrada de veículos e motoristas: uma visão geral. *Revista Eletrônicas Sistemas e Gestão*, v. 4, n. 3, p. 182 – 204, set.\ dez. 2010. Disponível em: <<http://www.uff.br/sg/index.php/sg/article/viewFile/V4N3A1/V4N3A1>>. Acesso em: 24 mar. 2011.

RIZZATO, D. B.; CONSTANTINO, A. A.; MELO, E. L.. Automação e Otimização do Escalonamento de Enfermeiros. In: XXX Encontro Nacional de Engenharia de Produção,

2010, São Carlos - SP. Anais do XXX Encontro Nacional de Engenharia de Produção, 2010. v. 1. p. 1-13.

SANTOS, A. G. Método de Geração de Colunas e Meta-heurísticas para Alocação de Tripulação. 2008. Dissertação (Doutorado em Ciência da Computação) – Universidade Federal de Minas Gerais, Belo Horizonte.

SANTOS, A. G.; MATEUS, G. R. Crew Scheduling Urban Problem: an Exact Column Generation Approach Improved by a Genetic Algorithm. In: IEEE Congress on Evolutionary Computation, p. 1725–1731, 2007. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04424681>>. Acesso em 14 mai. 2011.

SILVA, E. L.; MENEZES, E. M. Metodologia da Pesquisa e Elaboração de Dissertação. 4. ed. Florianópolis: UFSC, 2005. 138 p.

SILVA, G. P.; CUNHA, C. B. Uso da técnica de busca em vizinhança de grande porte para a programação da escala de motoristas de ônibus urbano. Transportes, v. 8 n. 2 p. 37-45, 2010. Disponível em: <<http://www.revistatransportes.org.br/index.php/anpet/article/view/422>>. Acesso em: 14 mai. 2011.

SILVA, G. P.; SOUZA, M. J. F.; ALVES, J. M. C. B. *Simulated Annealing* Aplicado à Programação da Tripulação no Sistema de Transporte Público. In: XXII Encontro Nacional de Engenharia de Produção, 2002, Curitiba. Anais do XXII ENEGEP. Disponível em: <http://www.abepro.org.br/biblioteca/ENEGEP2002_TR68_0254.pdf>. Acesso em: 24 mar. 2011.

SILVA, G. P.; SOUZA, M. J. F.; GOMES JR., A. C. Uma Metodologia Baseada em Emparelhamentos Sucessivos Aplicada ao Problema de Programação de Tripulações. In: XXXV Simpósio Brasileiro de Pesquisa Operacional - SBPO, 2005, Gramado. Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional - SBPO. Disponível em: <<http://www.decom.ufop.br/prof/marcone/publicacoes/SBPO-2005-PPT-TC12259.pdf>>. Acesso em: 24 mar. 2011.

SILVA, G. P.; SOUZA, M. J. F.; REIS, J. A. Um método exato para otimizar a escala de motoristas e cobradores do sistema de transporte público. In: XVIII Congresso de Pesquisa e Ensino em Transportes, 2004, Florianópolis. Anais do XVIII Congresso de Pesquisa e Ensino em Transportes. Disponível em: <http://www.cbtu.gov.br/estudos/pesquisa/anpet_xviiiCongrespens/ct/ct_arq41.pdf>. Acesso em: 14 mai. 2011.

SIQUEIRA, P. H. Aplicação do Algoritmo do Matching no Problema da Construção de Escalas de Motoristas e Cobradores de Ônibus. 1999. Dissertação (Mestrado em Métodos Numéricos em Engenharia) – Universidade Federal do Paraná, Curitiba. Disponível em: <http://www.degraf.ufpr.br/docentes/paulo/publicacoes_arquivos/paulohsDiss.pdf>. Acesso em: 14 mai. 2011.

SIQUEIRA, P. H. *et al.* Uma Proposta de Solução para o Problema da Construção de Escalas de Motoristas e Cobradores de Ônibus por meio do Algoritmo do *Matching* de Peso Máximo. Gestão & Produção, v. 11, n. 2, p. 187–196, 2004. Disponível em: <<http://www.scielo.br/pdf/gp/v11n2/a05v11n2.pdf>>. Acesso em: 24 mar. 2011.

SOBRAPO. Pesquisa Operacional. Disponível em: <http://www.sobrapo.org.br/o_que_e_po.php>. Acesso em: 04 out. 2011.

SOUZA, M. J. F.; RODRIGUES, M. M. S.; MAPA, S. M. S.; SILVA, G. P. Um estudo das heurísticas Simulated Annealing e VNS aplicadas ao problema de programação de tripulações. In: XXIII Encontro Nacional de Engenharia de Produção, 2003, Ouro Preto. Anais do XXII ENEGEP. Disponível em: <http://www.abepro.org.br/biblioteca/ENEGEP2003_TR0608_0872.pdf>. Acesso em: 14 mai. 2011.

Universidade Estadual de Maringá
Departamento de Engenharia de Produção
Av. Colombo 5790, Maringá-PR CEP 87020-900
Tel: (044) 3011-4196/3011-5833 Fax: (044) 3011-4196