

ROGÉRIO CALVI

**UM ALGORITMO PARA O PROBLEMA DE
ESCALONAMENTO DE TRIPULAÇÃO EM EMPRESAS DE
ÔNIBUS**

MARINGÁ

2005

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

ROGÉRIO CALVI

**UM ALGORITMO PARA O PROBLEMA DE
ESCALONAMENTO DE TRIPULAÇÃO EM EMPRESAS DE
ÔNIBUS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Ademir Aparecido Constantino

MARINGÁ

2005

ROGÉRIO CALVI

**UM ALGORITMO PARA O PROBLEMA DE
ESCALONAMENTO DE TRIPULAÇÃO EM EMPRESAS DE
ÔNIBUS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Aprovado em 29/08/2005.

BANCA EXAMINADORA

Prof. Dr. Ademir Aparecido Constantino
Universidade Estadual de Maringá – UEM

Prof. Dr. Maurício Fernandes Figueiredo
Universidade Estadual de Maringá – UEM

Prof. Dr. Silvio Alexandre de Araujo
Universidade Estadual de Maringá – UEM

Prof. Dr. Celso Carnieri
Universidade Federal do Paraná – UFPR

DEDICATÓRIA

Dedico este trabalho

A minha esposa Leticia.

AGRADECIMENTOS

Agradeço primeiramente a Jeová Deus, pela vida, força e habilidades necessárias para tornar este trabalho realidade.

Ao Prof. Dr. Ademir Aparecido Constantino, pela orientação e oportunidade que tornaram possível o desenvolvimento deste trabalho.

Aos professores do Programa de Pós-Graduação em Ciência da Computação pelos ensinamentos recebidos.

A minha esposa Letícia, pelo apoio e compreensão.

Aos meus pais, que muito fizeram por mim e incentivaram-me a continuar no caminho dos estudos.

À CAPES, pelo apoio financeiro.

RESUMO

O Problema de Escalonamento de Tripulação (PET) consiste em encontrar um conjunto de jornadas que cubra a escala diária de ônibus de uma empresa de transporte público minimizando os custos e satisfazendo um conjunto de leis trabalhistas e regulamentações da empresa. Neste trabalho é proposto um novo algoritmo heurístico para resolução do PET utilizando múltiplos problemas de designação. Testes computacionais são realizados usando dados reais de duas empresas de transporte rodoviário urbano e instâncias aleatórias extraídas de dados reais. Os resultados são comparados com os resultados obtidos por um modelo de cobertura de conjunto e com limites inferiores calculados para as instâncias.

Palavras-chave: problema de escalonamento de tripulação, problema de designação.

ABSTRACT

The Crew Scheduling Problem (CSP) consists in finding a set of duties which covers the bus daily schedule of a public transport company, minimizing costs and satisfying a set of work laws and company regulations. In this work it is proposed a new heuristic algorithm to solve the CSP using multiple assignment problems. Computational tests are realized using real data of two companies of urban bus and randomized instances extracted from real data. Results are compared with results of a resolution using the set covering model and with lower bounds calculated for instances.

Key-words: crew scheduling problem, assignment problem.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Considerações Iniciais	1
1.2	Importância do Trabalho	2
1.3	Objetivos do Trabalho	5
1.4	Estrutura do Trabalho	5
2	REVISÃO DA LITERATURA	7
2.1	Problema de Escalonamento de Tripulação	7
2.2	Modelos e Algoritmos de Escalonamento de Tripulação	8
2.2.1	Métodos Baseados em Cobertura de Conjunto e Partição de Conjunto	9
2.2.2	Métodos Heurísticos que não Utilizam o Modelo de Cobertura de Conjunto	14
2.3	Problema de Escalonamento de Trabalhadores em Local Fixo	15
2.4	Modelos e Algoritmos para Integração de Escalonamento de Veículos e Tripulação	16
3	ALGORITMO PROPOSTO	20
3.1	Considerações Iniciais	20
3.2	Escalonamento Prévio dos Veículos	22
3.3	Descrição do Algoritmo Proposto	22
3.3.1	Construção da Solução Inicial	22
3.3.2	Fase de Melhoramento	26
3.3.3	Análise de Complexidade do Algoritmo Proposto	31
3.4	Resolução do Problema de Escalonamento de Tripulação como um Problema de Cobertura de Conjunto	33

3.5	Um Limite Inferior para o Problema de Escalonamento de Tripulação	34
3.6	Limites Inferiores para o Problema de Cobertura de Conjunto	35
4	RESULTADOS E ANÁLISE DOS RESULTADOS	38
4.1	Características dos Dados	38
4.1.1	Características das Instâncias de Teste	38
4.1.2	Descrição das Regras Trabalhistas Consideradas	38
4.2	Resultados	40
4.2.1	Resultados Obtidos pelo Algoritmo Proposto	40
4.2.2	Comparação entre os Procedimentos Utilizados na Fase de Melhoramento	41
4.2.3	Resultados Obtidos Utilizando-se o Modelo de Cobertura de Conjunto	42
4.2.4	Comparação dos Resultados do PCC com os Resultados do AP	45
4.2.5	Comparação com a Solução da Empresa	46
4.3	Análise dos Resultados	48
5	CONCLUSÕES	50
5.1	Conclusões	50
5.2	Recomendações para Trabalhos Futuros	51
	REFERÊNCIAS BIBLIOGRÁFICAS.....	51

LISTA DE FIGURAS

Figura 1.1 - Atividades do planejamento operacional no transporte coletivo.....	1
Figura 3.1 - Camadas do Grafo G.....	23
Figura 3.2 - Estrutura da matriz de custos C.....	24
Figura 3.3 - Procedimento para construção da solução inicial.....	25
Figura 3.4 - Solução obtida na construção inicial e os possíveis cortes.....	26
Figura 3.5 - Exemplo de execução do procedimento M1.....	27
Figura 3.6 - Estrutura da Matriz D.....	29
Figura 3.7 - Exemplo de execução do procedimento M2.....	30
Figura 3.8 - Procedimento para melhoramento da solução.....	31
Figura 3.9 - Grafo de formação de pedaços.....	33

LISTA DE TABELAS

Tabela 4.1 – Resultados obtidos pelo algoritmo proposto.....	41
Tabela 4.2 – Comparação entre os procedimentos de melhoramento.....	42
Tabela 4.3 – Resultados obtidos para o PCC com $DurMin = 160$ e $DurMax = 280$	43
Tabela 4.4 – Resultados obtidos para o PCC com $DurMin = 60$ e $DurMax = 180$	44
Tabela 4.5 – Comparação dos resultados do AP com os resultados do PCC.....	46
Tabela 4.6 – Comparação das soluções obtidas com a solução da Empresa	47
Tabela 4.7 – Comparação qualitativa das soluções obtidas com a solução da Empresa.....	47

ABREVIATURAS, SIGLAS E SÍMBOLOS UTILIZADOS NO TEXTO

AdHE – Adicional de Hora Extra

AG – Algoritmo Genético

AP – Algoritmo Proposto

ATP – Algoritmo de Treinamento Populacional

BT – Busca Tabu

CTL – Custo da Troca de Linha

ES – Evolução Simulada

LI – Limite Inferior

LIP – Limite Inferior do Problema

NJD – Número de jornadas à direita do corte

NJE – Número de jornadas à esquerda do corte

PCC – Problema de Cobertura de Conjunto

PET – Problema de Escalonamento de Tripulação

PEV - Problema de Escalonamento de Veículos

PL – Programação Linear

PLI – Programação Linear Inteira

PPC – Problema de Partição de Conjunto

PLG – Problema de Relaxação Lagrangeana

SA – *Simulated Annealing*

ZPL – Valor da função objetivo do problema de PL

ZPLG – Valor da função objetivo do PLG

ZPLI – Valor da função objetivo do problema de PLI

1 INTRODUÇÃO

1.1 Considerações Iniciais

O transporte coletivo de passageiros é um setor da economia essencial para a sociedade. Além de empregar uma parcela considerável da população em sua atividade fim, bem como em suas atividades relacionadas, grande parcela da população depende dos serviços de transporte público diariamente para locomoção até seu trabalho.

O planejamento operacional no transporte coletivo envolve várias atividades. A Figura 1.1 mostra a seqüência de atividades geralmente realizadas em tal planejamento.

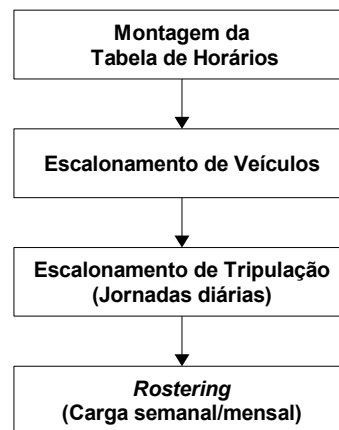


Figura 1.1 - Atividades do planejamento operacional no transporte coletivo.

Em geral o órgão gestor de transporte do município define as linhas de ônibus, faz um levantamento da demanda e monta uma tabela de horários que contém os locais de saída/chegada e horários de início/fim das viagens de cada linha. Esta tabela de horários é

entregue à empresa de transporte coletivo contratada que deverá realizar as viagens constantes na mesma. A empresa primeiro faz um *escalonamento de veículos*, que consiste em alocar seqüências de viagens aos veículos. A seguir realiza o *escalonamento de tripulação* que consiste em formar jornadas diárias de trabalho que serão alocadas às tripulações (motorista e cobrador). O *rostering* consiste em alocar jornadas às tripulações considerando um período mais longo (*e.g.* uma semana). Neste trabalho será considerado apenas o escalonamento de tripulação.

O problema de escalonamento de tripulação (PET) é um problema de difícil resolução. O grande número de escalas possíveis torna inviável resolver problemas de grande porte utilizando-se métodos exatos. Assim, o PET é normalmente abordado através de técnicas heurísticas.

1.2 Importância do Trabalho

Na administração de uma empresa de transporte de passageiros, as duas atividades mais complexas e relevantes a nível operacional é o escalonamento de veículos e o escalonamento de tripulação, ou seja, determinar a seqüência de viagens que será realizada por cada veículo e a jornada de trabalho de cada tripulação.

O escalonamento de tripulação é mais complexo que o escalonamento de veículos, pois as jornadas dos condutores devem obedecer a um grande conjunto de restrições legais, ao passo que as escalas de veículos estão sujeitas a poucas restrições. Dos dois problemas o mais relevante é o escalonamento de tripulação, pois os custos com pessoal nas empresas de transporte coletivo correspondem, aproximadamente, 50% do custo operacional total.

Planejar de forma eficiente o escalonamento de condutores em uma frota de tamanho considerável para a realização das viagens exigidas é uma tarefa difícil devido ao número de combinações possíveis ser muito grande. Assim um sistema computacional utilizando-se das técnicas de otimização apropriadas seria de grande auxílio na elaboração de um plano eficiente de escalonamento de condutores.

As características do problema conduzem naturalmente para um modelo de Otimização Combinatória. Alguns destes modelos já foram provados como sendo da classe NP-difícil, como por exemplo o Problema de Cobertura de Conjuntos (PCC), bastante utilizado para modelar o PET. Visto ser o PCC um problema NP-difícil a resolução de grandes instâncias do PET por este modelo torna-se inviável através de um algoritmo exato, sendo necessário o desenvolvimento de algoritmos heurísticos que produzam boas soluções em um tempo computacional aceitável.

Em geral, o escalonamento de veículos e o escalonamento de tripulação são problemas que se interagem, e portanto, seria desejável resolvê-los de forma integrada. No entanto modelos que incorporam os dois problemas são bastante complexos e, por isso, aplicações práticas tratam primeiro da resolução do escalonamento de veículos e depois resolvem o escalonamento de tripulação.

O escalonamento de tripulação deve ser realizado com o menor custo possível, sendo que horas extras devem ser evitadas pois têm um custo maior em relação à hora de trabalho da jornada normal. Em geral as empresas acreditam que devam reduzir ao mínimo o número de tripulações e para isso utilizam-se ao máximo de horas extras, no entanto, utilizar o número mínimo de condutores nem sempre é a opção de menor custo, pois com a contratação adicional de condutores e uma distribuição melhor de trabalho em cada jornada, pode-se

conseguir um custo total menor por reduzir o número de horas extras necessárias ao mínimo. Além da economia a curto prazo devido à redução do custo operacional, estas empresas podem obter também uma economia a longo prazo, pois um escalonamento de tripulação otimizado respeitando as regras trabalhistas evitaria futuras reclamações trabalhistas.

Muitas empresas resolvem o problema manualmente pelo desconhecimento de técnicas computacionais específicas e/ou pela dificuldade de implementação de tais técnicas. Em virtude da complexidade e da dimensão do problema, a solução manual dificilmente atinge a solução ótima. Além disso, o tempo despendido nesta tarefa é bastante significativo, fazendo com que o encarregado de tal tarefa simplesmente encontre uma solução aceitável, tornando inviável avaliar outras possibilidades de alocação que poderiam ser melhores para a empresa. Sistemas computacionais podem contribuir para a redução do custo operacional. Além disso o tempo economizado na tarefa de efetuar a alocação das tripulações permitirá que sejam elaborados mais de um plano de alocação, considerando cenários alternativos, tais como alterações em acordos coletivos da categoria, fornecendo uma base para a tomada de decisões estratégicas.

Um bom escalonamento de tripulação também traz benefícios para os funcionários. Características indesejáveis normalmente observadas no escalonamento manual como jornadas de trabalho excessivas e intervalos entre jornadas insuficientes seriam eliminadas.

Visto que os custos operacionais sempre tendem a ser repassados para as tarifas, mesmo que estas sejam controladas pelo poder público, um planejamento operacional mais eficiente poderia vir a contribuir para a manutenção ou até mesmo a redução de tarifas.

Com este trabalho pretende-se contribuir para a resolução do PET apresentando-se uma nova abordagem de resolução. O algoritmo apresentado utiliza-se de múltiplos problemas de

designação e é capaz de encontrar soluções de boa qualidade para grandes instâncias em um tempo computacional pequeno.

1.3 Objetivos do Trabalho

O objetivo geral deste trabalho é resolver o problema de escalonamento de tripulação no transporte rodoviário urbano/metropolitano de passageiros.

Os objetivos específicos são:

- a) Propor um algoritmo heurístico para resolução do problema de escalonamento de tripulação;
- b) Testar o algoritmo proposto utilizando instâncias de dados reais e instâncias aleatórias baseadas em dados reais;
- c) Comparar as soluções obtidas pelo algoritmo proposto com soluções obtidas por meio da resolução do problema de escalonamento de tripulação modelado como problema de cobertura de conjunto;

1.4 Estrutura do Trabalho

O presente trabalho está dividido em cinco capítulos. Neste primeiro capítulo, têm-se uma introdução, a importância do trabalho e os objetivos do mesmo. O segundo capítulo traz uma

revisão da literatura sobre o problema de escalonamento de tripulação, incluindo modelos e algoritmos para resolução deste problema. O terceiro capítulo descreve o algoritmo proposto. No quarto capítulo são apresentados e analisados os resultados obtidos. Finalmente, o quinto capítulo apresenta as conclusões do trabalho e recomendações para trabalhos futuros.

2 REVISÃO DA LITERATURA

2.1 Problema de Escalonamento de Tripulação

O Problema de Escalonamento de Tripulação (PET), consiste em gerar uma escala de trabalho, isto é, um conjunto de jornadas que cubra toda uma escala de ônibus predefinida minimizando os custos. Além disso é necessário satisfazer um conjunto de restrições legais estabelecidas em lei e contratos com o sindicato da categoria, bem como regulamentações da companhia, o que aumenta ainda mais a complexidade do problema.

Segue abaixo alguns conceitos básicos relacionados ao PET:

- Um *bloco* é uma seqüência de viagens que um veículo tem que realizar em um dia, geralmente começando e terminando na garagem. O conjunto de blocos forma a *escala de ônibus*;
- Uma *oportunidade de troca* é um local e horário onde é permitido que uma tripulação abandone ou assuma um veículo;
- Uma *tarefa* é a porção de trabalho entre duas oportunidades de trocas consecutivas. Portanto durante a realização de uma tarefa não é possível que haja troca de tripulação. As tarefas são os dados de entrada para resolução do PET;
- Um *pedaço de trabalho*, ou simplesmente *pedaço*, contém um número de tarefas consecutivas;

- O trabalho diário a ser realizado por uma tripulação é chamado *jornada* e compreende um ou mais pedaços de trabalho;
- Geralmente uma *tripulação* envolve um motorista e um cobrador, mas em alguns casos pode referir-se à apenas um motorista;
- Uma *escala de tripulação* ou *escala de trabalho* é o conjunto de jornadas que cobre toda a escala de ônibus. Esta é a saída da resolução do PET.

Algumas restrições legais que devem ser observadas são: o tempo máximo de trabalho permitido, o intervalo mínimo e máximo para descanso dentro da jornada, o intervalo mínimo para descanso entre jornadas consecutivas e a duração máxima do trabalho contínuo. Essas restrições costumam ser diferentes para cada cidade ou região.

2.2 Modelos e Algoritmos de Escalonamento de Tripulação

A maioria dos trabalhos trata da resolução do PET modelando-o como um problema de cobertura de conjunto ou partição de conjunto, porém alguns utilizam-se de métodos heurísticos sem utilizar tais modelos. Na subseção 2.2.1 são apresentados alguns trabalhos que tratam da primeira abordagem e na subseção 2.2.2 são citados trabalhos que não utilizam tal abordagem.

2.2.1 Métodos Baseados em Cobertura de Conjunto e Partição de Conjunto

Entre os modelos mais utilizados para resolução do problema de escalonamento de tripulação estão o Problema de Cobertura de Conjunto (PCC) e o Problema de Partição de Conjunto (PPC) que são expressos como problemas de programação linear inteira (PLI).

Sejam n o número de potenciais jornadas (colunas), m o número de pedaços de jornada (linhas) a serem cobertos e c_j o custo associado à jornada j . Considerando ainda as constantes inteiras a_{ij} assumindo valor 1 se a jornada j cobre o pedaço i e o valor 0 caso contrário e usando a variável binária $x_j = 1$ se a coluna j está na solução, podemos formular o seguinte modelo do PCC:

$$\text{Min } \sum_{j=1}^n c_j x_j \quad (2.1)$$

$$\text{Sujeito a: } \sum_{j=1}^n a_{ij} x_j \geq 1, i \in \{1, 2, \dots, m\} \quad (2.2)$$

$$x_j \in \{0, 1\}, j \in \{1, 2, \dots, n\} \quad (2.3)$$

A função objetivo (2.1) minimiza o custo total. A restrição (2.2) garante que cada linha (pedaço de trabalho) seja coberta ao menos por uma coluna (jornada), e a restrição de integralidade (2.3) garante que somente jornadas completas serão consideradas.

Alterando-se a restrição (2.2) da seguinte maneira, podemos formular o PPC:

$$\text{Sujeito a: } \sum_{j=1}^n a_{ij}x_j=1, i \in \{1, 2, \dots, m\} \quad (2.4)$$

Os dois modelos são bastante semelhantes, exceto pela substituição da desigualdade no modelo de cobertura de conjunto pela igualdade no modelo de partição. No caso do modelo de cobertura, cada pedaço de trabalho deve ser coberto por pelo menos uma jornada, ao passo que no modelo de partição, cada pedaço de trabalho deve ser coberto exatamente por uma jornada. Em teoria, a solução para o PLI é a escala com custo mínimo. Porém, o número total de possíveis jornadas é usualmente muito grande para ser revolido usando o modelo de PLI. Conjuntos de colunas (jornadas) muito grandes têm sido reduzidos por heurísticas, ou o problema tem sido decomposto em vários subproblemas e resolvidos separadamente. Tanto o PCC como o PPC são problemas NP-difíceis (KARP¹, 1975 e KARP², 1976 *apud* GOLDBARG e LUNA, 2000).

Beasley (1987) propôs um algoritmo exato para o PCC que combina testes de redução com subida dual (*dual ascent*), otimização de subgradiente e programação linear. No artigo são apresentados resultados computacionais para problemas envolvendo até 400 linhas e 4000 colunas.

Beasley e Chu (1996) apresentaram um algoritmo genético para resolver o PCC. Os autores propuseram várias modificações para os procedimentos genéticos básicos incluindo um novo operador *crossover* chamado *fusão* baseado no *fitness* dos pais, uma taxa de mutação variável e um operador heurístico para viabilidade desenhado especialmente para o problema de cobertura de conjunto. Como estrutura do cromossomo usaram uma seqüência binária de n

bits, onde n é o número de colunas do PCC e o valor 1 para o i -ésimo *bit* implica que a coluna

¹KARP, R. M. On the Computational Complexity of Combinatorial Problems, *Networks*5, p. 45-68, 1975.

²KARP, R. M. The Probabilistic Analysis of Some Combinatorial Search Algorithms, em: J. F. Traub (ed.). *Algorithms and Complexity: New Directions and Recent Results*, p. 1-19, Academic Press, 1976.

i está na solução. Os resultados computacionais apresentados mostraram que o algoritmo é capaz de produzir soluções de alta qualidade. Os testes foram realizados sobre um conjunto de 65 problemas de diversos tamanhos com até 1000 linhas e 10000 colunas.

Caprara, Fischetti e Toth (1999) propuseram uma heurística baseada em relaxação Lagrangeana para o PCC. As principais características do algoritmo proposto são: 1) um esquema de *pricing* dinâmico para as variáveis, sendo acoplado com otimização de subgradiente e algoritmos *greedy*, e 2) o uso sistemático de fixação de colunas para obter soluções melhoradas. O algoritmo foi testado em problemas de grande escala, envolvendo até 5000 linhas e 10^6 colunas. Em 92 das 94 instâncias de teste, a solução ótima (ou a melhor conhecida) foi encontrada em um tempo computacional pequeno. Além disso, entre 18 instâncias cujo ótimo não é conhecido, 6 soluções são melhores do que qualquer solução encontrada anteriormente por outras técnicas.

Wren e Wren (1995) usaram um algoritmo genético com um novo operador de cruzamento para resolver o escalonamento diário de condutores. Este algoritmo tinha como objetivo substituir o terceiro estágio do sistema IMPACS, que seleciona, à partir do conjunto de potenciais jornadas geradas nos estágios anteriores, um subconjunto que cubra todo o trabalho de maneira eficiente. Em sua representação do cromossomo cada posição do gene (*locus*) corresponde a um pedaço de trabalho e o seu valor (*alelo*) é a jornada que cobre o respectivo pedaço. Os autores comparam os resultados obtidos na solução de um problema usando este algoritmo e usando o sistema IMPACS e concluem dizendo que apesar do IMPACS ter produzido melhores resultados, este possui código que têm sido refinado ao longo de um período de 10 anos, ao passo que o algoritmo genético foi desenvolvido em um período muito curto.

Mayerle (1996) propôs um sistema de apoio à decisão para o planejamento operacional de empresas de transporte rodoviário urbano de passageiros. Neste sistema o problema de escalonamento de tripulação é resolvido através do modelo de cobertura de conjunto, cuja solução é obtida utilizando-se um algoritmo genético. Os pedaços que compõem as colunas são formados por encontrar-se o menor caminho em um grafo de formação de pedaços. As colunas são geradas pela combinação de dois a quatro pedaços.

Fortsyth e Wren (1997) aplicaram a meta-heurística *Ant System* para resolver o escalonamento de tripulação. O algoritmo utiliza as potenciais jornadas geradas pelo sistema TRACS II. Cada vértice representa uma oportunidade de troca. Em cada vértice a formiga escolhe uma jornada do conjunto de jornadas que têm início na oportunidade de rendição representada pelo vértice. O processo é repetido até que todo o trabalho seja coberto. As soluções obtidas com este algoritmo não foram muito boas quando comparadas com as do TRACS II.

Ohlsson, Peterson e Söderberg (2001) desenvolveram uma Rede Neural Artificial com realimentação de campo médio para resolver o PCC. Os autores usaram uma função penalidade multilinear para obter uma codificação conveniente das restrições de desigualdade. Uma energia mínima aproximada é obtida por iterar um conjunto de equações de campo médio em combinação com resfriamento. O algoritmo foi testado para um conjunto de problemas de grande escala (até 5000 linhas e 10^6 colunas) e os resultados mostraram que esta abordagem produz resultados a um percentual pequeno das soluções ótimas com tempo de execução bastante eficiente.

Li e Kwan (2000) apresentaram um algoritmo *Evolução Simulada Fuzzy* para o problema de escalonamento de tripulação, que envolve resolver um modelo do PCC. Este algoritmo incorpora a idéia de avaliação compreensiva *fuzzy* dentro de Evolução Simulada, combinando

as características de melhoria iterativa e perturbação construtiva para explorar o espaço de solução efetivamente e obter escalas superiores. Evolução Simulada (ES) é similar ao Algoritmo Genético (AG) no aspecto de usar evolução de uma geração para a próxima. No entanto, ao passo que AG mantém um número de soluções (população) durante cada geração como pais, gerando filhos pelo uso de operadores de cruzamento, ES somente gera um filho a partir de um único pai durante cada geração usando algoritmos heurísticos de domínio específico para derivar uma nova solução de partes da solução anterior. Os passos de *Avaliação* e *Construção* do algoritmo ES foram fuzificados. No passo *Avaliação* cada jornada na solução é avaliada por uma função de avaliação, que é baseada em seu estado de cobertura e cinco critérios fuzificados. No passo *Construção* aplicou-se uma heurística *greedy* com uma função de avaliação fuzzy para formar uma solução completa a partir de uma solução parcial. Os resultados do algoritmo foram comparados com a melhor solução conhecida de alguns problemas e apresentou em média 1,99% jornadas a mais em termos do número total de jornadas e 1,84% mais caro em termos de custo total.

Mauri e Lorena (2004) descreveram um metodologia interativa baseada na aplicação do Algoritmo de Treinamento Populacional (ATP) juntamente com programação linear (PL) para a geração de escalas de tripulações de um sistema de transporte coletivo. O ATP através de informações do problema de PL tem a responsabilidade de gerar boas colunas que formarão um problema de partição de conjuntos resolvido por uma ferramenta comercial para resolução de PLI. Os testes foram realizados em instâncias geradas aleatoriamente, porém baseadas em problemas reais, que continham de 25 a 500 tarefas. Os resultados do ATP foram melhores quando comparados com uma abordagem usando a metaheurística *Simulated Annealing*.

2.2.2 Métodos Heurísticos que não Utilizam o Modelo de Cobertura de Conjunto

Apesar da maioria dos trabalhos encontrados na literatura tratarem de métodos que utilizam os modelos de cobertura ou partição de conjunto para resolver o problema de escalonamento de tripulação (PET), existem alguns trabalhos que utilizam métodos heurísticos para resolução do PET sem utilizar tais modelos.

Shen e Kwan (2000) desenvolveram o algoritmo HACS que utiliza uma Busca Tabu para resolver o problema de escalonamento de tripulação. A abordagem HACS baseia-se na representação do problema envolvendo seqüências de *links*. Os *links* e as oportunidades de rendição associadas compõem o espaço de solução. Os autores desenvolveram quatro estruturas de vizinhança aplicadas em seqüência para atingir uma maior cobertura do espaço de busca. Comparado com os resultados do sistema TRACS II sua qualidade das soluções é um pouco inferior.

Souza, Cardoso e Silva (2003) abordaram o problema de escalonamento de tripulação utilizando as meta-heurísticas *Simulated Annealing* (SA), Busca Tabu (BT) e uma hibridização dessas duas técnicas. Foi utilizado um único movimento para explorar a vizinhança, que consiste em realocar uma tarefa de um determinado tripulante i a um outro tripulante j . Os algoritmos foram testados com dados reais de uma empresa que opera na cidade de Belo Horizonte responsável por 11 linhas de ônibus, com frota empenhada de 111 veículos e que utiliza atualmente 219 tripulações para operar esta frota. O melhor resultado encontrado foi obtido aplicando-se BT, sendo que a aplicação de SA e a hibridização das duas técnicas também apresentaram economias em relação à solução atual da empresa.

Siqueira (1999) resolve o PET através do problema de emparelhamento em grafo não bipartido, sendo que as jornadas são formadas por no máximo dois pedaços. Os vértices do grafo representam os pedaços ao passo que as arestas estão associadas a pesos de acordo com a qualidade da combinação entre pares de pedaços. Esses pedaços são formados previamente por um sistema não abordado no trabalho. A resolução proposta foi testada usando dados de três empresas do transporte coletivo da cidade de Curitiba que usavam algoritmos heurísticos. Os resultados obtidos foram melhores do que os de tais algoritmos heurísticos.

2.3 Problema de Escalonamento de Trabalhadores em Local Fixo

Segundo Bodin *et al.* (1983) o problema de escalonamento de trabalhadores em local fixo pode ser usado para obter uma estimativa de tripulantes requeridos para o PET. Neste caso as considerações espaciais do PET são desconsideradas para obter-se uma figura clara das considerações temporais.

Para este problema considere a divisão de um dia de trabalho em T intervalos de tempo e uma demanda especificada de trabalhadores d_t associada com cada intervalo de tempo $t = 1, 2, \dots, T$. O problema de escalonamento de trabalhadores consiste em encontrar um conjunto de jornadas diárias que cubra todo o trabalho requerido.

Sejam n o número de jornadas diárias possíveis, c_j o custo associado à jornada j , e as constantes inteiras a_{ij} assumindo valor 1 se a jornada j cobre o período t e o valor 0 caso contrário, então podemos formular o seguinte modelo para o problema de escalonamento de trabalhadores em local fixo:

$$\text{Min } \sum_{j=1}^n c_j x_j \quad (2.5)$$

$$\text{Sujeito a: } \sum_{j=1}^n a_{tj} x_j \geq d_t, \quad t=1, 2, \dots, T \quad (2.6)$$

$$x_j \geq 0 \text{ e inteiro}, \quad j=1, 2, \dots, n \quad (2.7)$$

Esta formulação é uma generalização do problema de cobertura de conjunto.

Para usar este modelo no cálculo de um limite inferior para o PET deve-se definir d_t como sendo o número de veículos que operam durante todo o intervalo de tempo t . Assim, d_t é um limite inferior no número de jornadas de tripulações requeridas durante o período de tempo t .

2.4 Modelos e Algoritmos para Integração de Escalonamento de Veículos e Tripulação

Conforme já comentado na introdução, seria desejável resolver o escalonamento de veículos e tripulação de maneira integrada, visto que a resolução de um traz implicações para a resolução do outro, porém devido a complexidade e tamanho do modelo integrado costuma-se usar uma abordagem seqüencial, ou seja, primeiro resolve-se o escalonamento de veículos e depois o escalonamento de tripulação. Apesar da maioria dos trabalhos tratar da abordagem seqüencial, alguns tratam da resolução integrada.

Freling, Huismann e Wagelmans (2003) propuseram novas formulações matemáticas para integração dos problemas de escalonamento de veículo e tripulação considerando uma única

garagem e discutiram as correspondentes relaxações Lagrangeanas e heurísticas Lagrangeanas. Para resolver as relaxações Lagrangeanas, eles usaram a técnica de geração de colunas aplicada a modelos do tipo partição de conjuntos. Os autores trataram do caso em que é permitido que uma tripulação troque de veículo ao longo da sua jornada e do caso em que tal troca é proibida. Os testes computacionais usaram dados de problemas reais de tamanho entre 24 e 238 viagens. Os testes confirmaram os benefícios da utilização de integração ao invés da abordagem seqüencial, sendo que para o caso em que as trocas de veículos não eram permitidas os ganhos foram maiores.

Huisman, Freling e Wagelmans (2003) apresentam dois modelos e algoritmos para o escalonamento integrado de veículo e tripulação considerando várias garagens. Ambos os algoritmos baseiam-se na combinação de geração de coluna e relaxação Lagrangeana. Neste trabalho eles estenderam o modelo matemático e a abordagem de solução desenvolvidas para o caso envolvendo apenas uma garagem abordado em (FRELING; HUISMAN; WAGELMANS, 2003). Os autores compararam as duas abordagens integradas propostas com a abordagem seqüencial tradicional usando instâncias de dados gerados aleatoriamente, bem como, dados do mundo real. Para teste foram usadas 8 instâncias com número de viagens variando entre 194 e 653, correspondendo a subconjuntos de um conjunto maior com 1104 viagens e 4 garagens da maior companhia de ônibus dos Países Baixos, e 10 instâncias geradas aleatoriamente com número de viagens variando de 80 a 400. Os dois algoritmos propostos apresentaram ganhos significativos em relação à abordagem seqüencial tradicional.

Groot e Huisman (2004) discutiram vários métodos para resolver grandes instâncias do mundo real do problema integrado de escalonamento de veículo e tripulação. Neste trabalho foram estudadas diversas maneiras de dividir uma grande instância do problema envolvendo várias garagens em instâncias menores para que seja resolvido de maneira integrada ou seqüencial. O

algoritmo utilizado foi um dos propostos em (HUISMAN; FRELING; WAGELMANS, 2003) e os testes foram realizados utilizando-se dois grandes conjuntos de dados, o primeiro contendo 1104 viagens e 4 garagens e o outro contendo 1372 viagens e 6 garagens. Primeiro testaram os métodos de divisão sobre oito subconjuntos do primeiro conjunto e depois escolheram o melhor método para aplicar sobre o conjunto total. Os testes mostraram que a economia obtida em relação à abordagem seqüencial é grande.

Ball, Lawrence e Dial (1983) propuseram uma heurística baseada em problemas de emparelhamento em grafo não bipartido para resolução integrada do problema de escalonamento de veículo e tripulação, porém enfatizando o escalonamento de tripulação. Neste modelo são considerados dois grafos, um representando o problema de escalonamento de veículo e outro o problema de escalonamento de tripulação. Um conjunto de restrições assegura que a solução obtida seja compatível em ambos os grafos. Os vértices representam partes de viagens denominadas *d-trips* que devem ser executadas por um veículo e tripulação, e dois vértices *s* e *t* representam a garagem. Os arcos destes grafos podem ser agrupados em dois tipos, aqueles que indicam que uma tripulação e um veículo seguem de uma *d-trip* para outra e aqueles que indicam que somente a tripulação segue de uma *d-trip* para a outra. O escalonamento de tripulação é resolvido através de um procedimento que têm três fases: 1) construção de pedaços; 2) melhoramento dos pedaços; e 3) construção das jornadas. A rotina de construção de pedaços forma pedaços cuja duração seja menor que uma constante *T*. Isto é obtido pela resolução de problemas de emparelhamento em grafo não bipartido. A rotina de melhoramento de pedaços combina pares de pedaços para formar um novo ou redividí-los em seguida formando dois novos pedaços usando uma heurística de trocas baseada em emparelhamento em grafo não bipartido. Na terceira fase são construídas jornadas contendo até quatro pedaços também resolvendo-se o problema de emparelhamento. A solução para o escalonamento de veículo é uma consequência da resolução do escalonamento de tripulação,

visto que as restrições asseguram que a solução obtida seja compatível em ambos os grafos. Um teste computacional usando uma instância real com 1602 *d-trips* apresentou soluções melhores que a atualmente utilizada.

3 ALGORITMO PROPOSTO

3.1 Considerações Iniciais

Muitos trabalhos encontrados na literatura tratam da resolução do Problema de Cobertura de Conjuntos (PCC) que é o modelo mais usado para representar o Problema de Escalonamento de Tripulação (PET), sendo que a maioria utiliza-se de heurísticas para resolvê-lo. Poucos trabalhos tentam resolver o PET sem usar o modelo do PCC, geralmente fazendo-o pelo uso de metaheurísticas, tais como *Simulated Annealing*, *GRASP*, Busca Tabu, entre outras.

O trabalho de Ball *et al.* (1983) resolve o PET e o PEV de maneira integrada através de problemas de emparelhamento em grafo não bipartido. O algoritmo proposto por Ball *et al.* (1983) consiste em três fases: 1) formação de pedaços com duração menor que um parâmetro T ; 2) melhoramento dos pedaços por combinar pares de pedaços formando um novo ou redividindo-os logo em seguida para formar dois novos pedaços; e 3) formação de jornadas contendo até quatro pedaços.

Siqueira (1999) também resolve o PET através do problema de emparelhamento em grafo não bipartido, sendo que as jornadas são compostas pela combinação de pares de pedaços, que são os vértices do grafo que representa o problema de emparelhamento ao passo que as arestas representam a possibilidade de combinação entre pares de pedaços.

O algoritmo proposto neste trabalho baseia-se na resolução de problemas de designação. O problema de designação é equivalente ao emparelhamento de custo mínimo em um grafo bipartido. Dada uma matriz de custos $[c_{ij}]$ de dimensão $n \times n$, o problema de designação

consiste em associar a cada linha uma coluna diferente de tal maneira que o soma dos custos seja mínima. Usando uma variável binária $x_{ij} = 1$ se a linha i está associada a coluna j , o problema pode ser formulado conforme abaixo:

$$\text{Min } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (3.1)$$

Sujeito a:

$$\sum_{j=1}^n x_{ij} = 1 \quad (i=1, 2, \dots, n) \quad (3.2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad (j=1, 2, \dots, n) \quad (3.3)$$

$$x_{ij} \in \{0, 1\} \quad (i, j=1, 2, \dots, n) \quad (3.4)$$

A principal diferença em relação aos trabalhos anteriores é que neste trabalho não é realizada uma formação prévia de pedaços. As jornadas são construídas utilizando-se tarefas e não pedaços, portanto não estão limitadas apenas às combinações formadas por pedaços formados previamente. Além disso, uma jornada pode conter qualquer quantidade de tarefas ao invés de estarem limitadas à uma quantidade específica de pedaços. A seguir são aplicadas heurísticas de melhoramento, não sobre pedaços, mas sim sobre as jornadas formadas na solução inicial. Portanto, o algoritmo proposto está dividido em duas fases: construtiva e melhoramento.

3.2 Escalonamento Prévio dos Veículos

O algoritmo para resolução do escalonamento de tripulação recebe como entrada um conjunto de tarefas. Estas são obtidas à partir dos blocos que compõem a escala de ônibus, sendo que cada tarefa corresponde à porção de trabalho existente entre duas oportunidades de trocas consecutivas.

Embora exista algoritmo de tempo polinomial para resolver o problema de escalonamento de veículos (PEV) quando há apenas uma garagem e todos os veículos são do mesmo tipo (BODIN *et al.*, 1983), neste trabalho o PEV não é resolvido, visto que o foco é o PET. Os blocos utilizados para obter-se as tarefas são os blocos da atual escala de ônibus das empresas consideradas.

3.3 Descrição do Algoritmo Proposto

O algoritmo proposto para resolução do PET divide-se em duas fases: 1) construção de uma solução inicial; e 2) melhoramento da solução. Tanto na primeira como na segunda fase são resolvidos problemas de designação. A seção 3.3.1 descreve o procedimento para construção da solução inicial e a seção 3.3.2 descreve os procedimentos usados na fase de melhoramento.

3.3.1 Construção da Solução Inicial

Considere o grafo $G(V, A)$, onde V é o conjunto de vértices que representam as tarefas que deverão ser alocadas às tripulações e A é o conjunto de arcos a_{ij} que indicam a possibilidade de uma mesma tripulação executar a tarefa j após a tarefa i . Considere ainda que os vértices estão

dispostos em camadas e que cada uma destas camadas é composta por vértices que representam tarefas que podem ser realizadas como seqüência de alguma tarefa na camada imediatamente anterior, e que não podem ser realizadas após tarefas localizadas na mesma camada ou camadas posteriores, conforme mostrado na Figura 3.1.

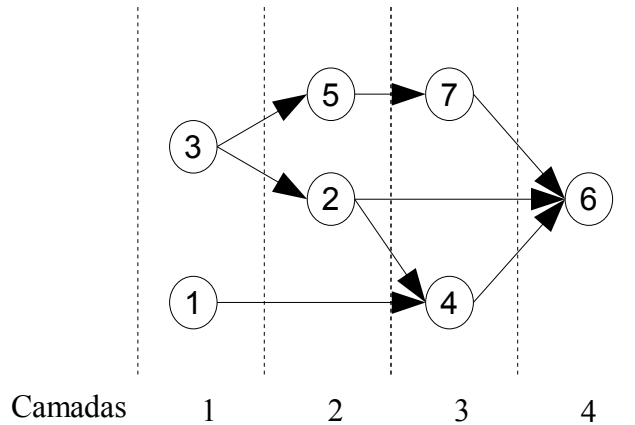


Figura 3.1 - Camadas do Grafo G .

A construção da solução inicial consiste em resolver um problema de designação para cada camada. O problema de designação em cada camada é representado pela matriz quadrada $C = [c_{ij}]$, de ordem $|T_k| + |J|$, onde T_k é o conjunto de tarefas da camada k e J é o conjunto de jornadas. O conjunto de jornadas J é inicializado com um número de jornadas suficiente para que nenhuma tarefa fique sem ser alocada, podendo este número ser estimado ou ser igual ao número de tarefas. Inicialmente todas as jornadas estão vazias e tarefas lhe são alocadas em cada iteração de acordo com a resolução do problema de designação da camada corrente. Ao final do procedimento as jornadas que permanecerem vazias são eliminadas da solução. A matriz $C = [c_{ij}]$ pode ser dividida em quatro blocos conforme mostrado na Figura 3.2 e seus elementos são definidos da seguinte forma:

- Bloco I. Para $i = 1, \dots, |J|$ e $j = 1, \dots, |T_k|$, se for possível alocar a tarefa j à jornada i , temos $c_{ij} = f(i, j) + CTL$, onde $j \in T_k$, a função f define o custo de alocar-se a tarefa j à jornada i , e CTL assume um valor que penaliza a troca de linha de ônibus na jornada se a linha de ônibus da tarefa j for diferente da linha de ônibus da última tarefa da jornada i e valor zero caso contrário. Se não for possível alocar a tarefa j à jornada i então $c_{ij} = \infty$;
- Bloco II. Para $i = 1, \dots, |J|$ e $j = |T_k| + 1, \dots, |T_k| + |J|$, c_{ij} é o custo da jornada i sem receber tarefa adicional;
- Bloco III. Para $i = |J| + 1, \dots, |T_k| + |J|$ e $j = 1, 2, \dots, |T_k|$, c_{ij} corresponde a um valor suficientemente alto para que nenhuma tarefa fique sem alocação;
- Bloco IV. Para $i = |J| + 1, \dots, |T_k| + |J|$ e $j = |T_k| + 1, \dots, |T_k| + |J|$, $c_{ij} = 0$;

	Tarefas	Tarefas Fictícias
Jornadas	I $c_{ij} = f(i, j) + CTL$ ou $c_{ij} = \infty$	II $c_{ij} =$ custo da jornada i sem tarefa adicional
Jornadas Fictícias	III $c_{ij} =$ custo por não alocar a tarefa j	IV $c_{ij} = 0$

Figura 3.2 - Estrutura da matriz de custos C .

Para formação do Bloco I pode-se utilizar uma das três funções abaixo para $f(i, j)$:

- i) $f_1(i, j)$ = custo da jornada i em minutos pagos considerando a inclusão da tarefa j na jornada i ;
- ii) ou, $f_2(i, j)$ = minutos ociosos na jornada i considerando a inclusão da tarefa j na jornada i ;
- iii) ou, $f_3(i, j)$ = minutos ociosos na jornada i + minutos extras na jornada i * $AdHE$ considerando a inclusão da tarefa j na jornada i , onde $AdHE$ é o adicional sobre horas extras definido pela legislação como sendo de 50%.

A Figura 3.3 resume o procedimento para construção da solução inicial.

<p>Entrada: Conjunto de tarefas;</p> <p>Passo 1: Inicialize os dados e faça $k = 1$;</p> <p>Passo 2: Forme a camada k contendo as tarefas que não podem ser realizadas na sequência de tarefas da mesma camada ou camadas posteriores;</p> <p>Passo 3: Monte a matriz de custos C;</p> <p>Passo 4: Resolva o problema de designação descrito pela matriz C e aloque as tarefas às jornadas de acordo com o resultado obtido.</p> <p>Passo 5: Faça $k = k + 1$ e retorne ao Passo 2 até que todas as tarefas estejam alocadas.</p>
--

Figura 3.3 - Procedimento para construção da solução inicial.

3.3.2 Fase de Melhoramento

Na fase de melhoramento são usados dois procedimentos. O primeiro, denominado M1, é similar ao encontrado em (MAYERLE, 1996) originalmente utilizado para melhorar a solução do escalonamento de veículos e neste trabalho adaptado para o escalonamento de tripulação. O segundo, denominado M2, foi desenvolvido neste trabalho especialmente para o escalonamento de tripulação com o objetivo de ser utilizado para melhorar a solução quando M1 não conseguir melhorar mais a solução.

A idéia do procedimento M1 é dividir as jornadas construídas em duas partes formando jornadas parciais. Cada divisão ou corte separa as tarefas de uma jornada formando duas jornadas parciais, uma contendo as tarefas à esquerda do corte e outra contendo as tarefas à direita do corte. Em seguida, estas jornadas parciais são recombinadas resolvendo-se um problema de designação. A Figura 3.4 ilustra uma solução obtida após a fase de construção e seu possíveis cortes. Os retângulos representam as tarefas e as setas o seqüenciamento das mesmas em uma jornada. Os cortes possíveis estão representados pelas linhas tracejadas.

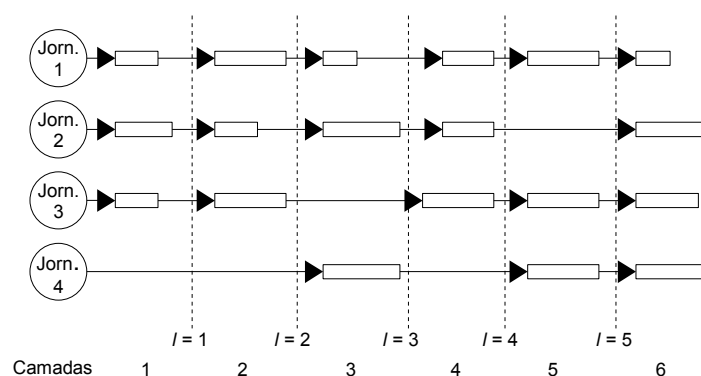


Figura 3.4 - Solução obtida na construção inicial e os possíveis cortes.

Uma iteração completa do procedimento M1 consiste em realizar $K-1$ cortes, onde K é o número de camadas obtido na fase de construção inicial, e resolver um problema de designação para cada corte. Após cada corte calcula-se os custos para recombinar as jornadas parciais à esquerda do corte com jornadas parciais à direita do corte. Estes custos formam a matriz quadrada $D = [d_{ij}]$ que representa um problema de designação. A resolução deste problema determina como as jornadas parciais devem ser recombinadas. A Figura 3.5 exemplifica este procedimento.

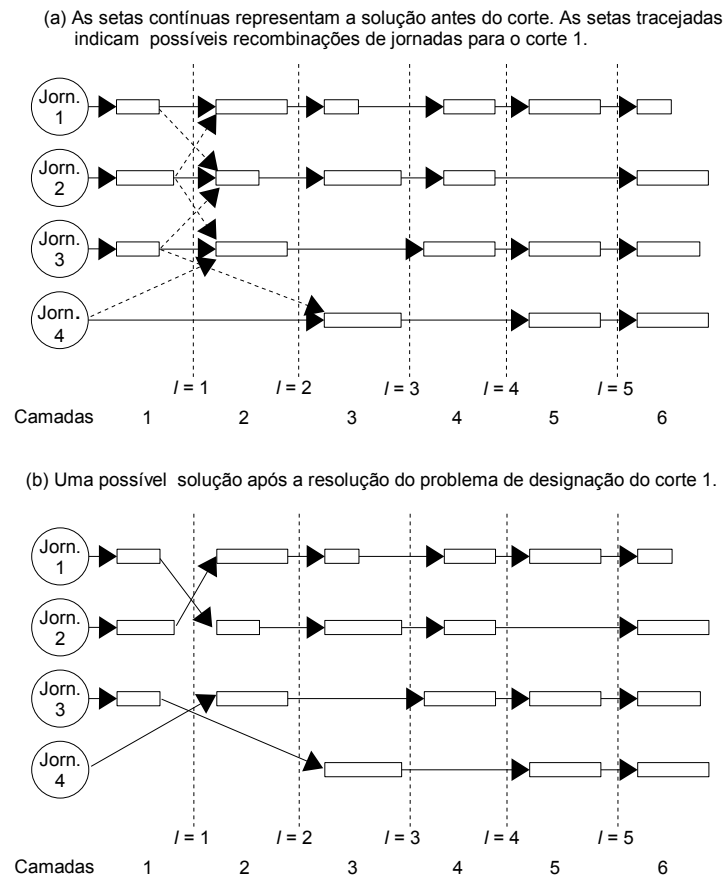


Figura 3.5 - Exemplo de execução do procedimento M1.

O problema de designação exige que a matriz D seja quadrada. Se o número de jornadas parciais à esquerda do corte (NJE) e o número de jornadas parciais à direita do corte (NJD) forem iguais esta exigência estará satisfeita e os elementos d_{ij} corresponderão ao custo em minutos pagos da jornada que combina as jornadas i e j (se for possível combiná-las) + CTL , onde CTL assume um valor pequeno que penaliza a troca de linha de ônibus na jornada se a linha da última tarefa da jornada i for diferente da linha da primeira tarefa da jornada j e valor zero caso contrário. Se não for possível combinar as jornadas i e j , d_{ij} corresponderá à soma dos custos das jornadas i e j .

Se NJE for diferente de NJD , para tornar a matriz D quadrada os elementos serão definidos conforme abaixo:

- Caso 1. Se $NJE > NJD$:

- i) Bloco I. Para $i = 1, \dots, NJE$, e $j = 1, \dots, NJD$, se for possível combinar as jornadas parciais i e j , d_{ij} será o custo em minutos pagos da jornada resultante da combinação das jornadas parciais i e j + CTL , onde CTL assume um valor pequeno que penaliza a troca de linha de ônibus na jornada se a linha da última tarefa da jornada i for diferente da linha da primeira tarefa da jornada j e valor zero caso contrário; Se não for possível combinar as jornadas parciais i e j , d_{ij} corresponderá à soma dos custos das jornadas i e j ;
- ii) Bloco II. Para $i = 1, \dots, NJE$, e $j = NJD + 1, \dots, NJE$, d_{ij} = custo em minutos pagos da jornada i ;

● Caso 2. Se $NJD > NJE$:

- i) Bloco I. Para $i = 1, \dots, NJD$, e $j = 1, \dots, NJE$, se for possível combinar as jornadas parciais i e j , d_{ij} será o custo em minutos pagos da jornada resultante da combinação das jornadas parciais i e j + CTL , onde CTL assume um valor pequeno que penaliza a troca de linha de ônibus na jornada se a linha da última tarefa da jornada i for diferente da linha da primeira tarefa da jornada j e valor zero caso contrário. Se não for possível combinar as jornadas parciais i e j , d_{ij} corresponderá à soma dos custos das jornadas i e j ;
- ii) Bloco II. Para $i = NJD + 1, \dots, NJE$, e $j = 1, \dots, NJE$, d_{ij} = custo em minutos pagos da jornada j ;

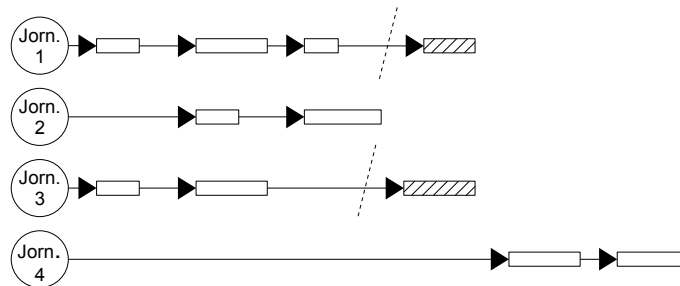
A Figura 3.6 apresenta a estrutura da matriz D para os dois casos descritos.

		Caso 1		Caso 2	
		Jornadas j	Jornadas fictícias	Jornadas j	Jornadas fictícias
Jornadas i	Jornadas i	I d_{ij} = custo da jornada que combina i e j + CTL ou d_{ij} = soma dos custos de i e j	II d_{ij} = custo da jornada i	I d_{ij} = custo da jornada que combina i e j + CTL ou d_{ij} = soma dos custos de i e j	
	Jornadas fictícias			II d_{ij} = custo da jornada j	

Figura 3.6 - Estrutura da Matriz D .

O procedimento M2 realiza um corte na solução atual que consiste em separar a última tarefa das jornadas com horas extras e resolver o problema de designação para este corte. A idéia deste procedimento é diminuir o custo da solução por retirar tarefas das jornadas com horas extras e realocá-las em jornadas que estão com tempo ocioso. Após o corte calcula-se os custos para recombinar as jornadas parciais à esquerda do corte com jornadas parciais à direita do corte. Estes custos formam a matriz quadrada $D = [d_{ij}]$ que representa um problema de designação e que tem a mesma estrutura já definida para o procedimento M1. A resolução deste problema determina como as jornadas parciais devem ser recombinadas. A Figura 3.7 exemplifica este procedimento.

(a) O corte separa a última tarefa das jornadas 1 e 3, pois possuem horas extras.



(b) Após a resolução do problema de designação a tarefa cortada da jornada 1 foi realocada no final da jornada 2 e a tarefa cortada da jornada 3 foi realocada no início da jornada 4.

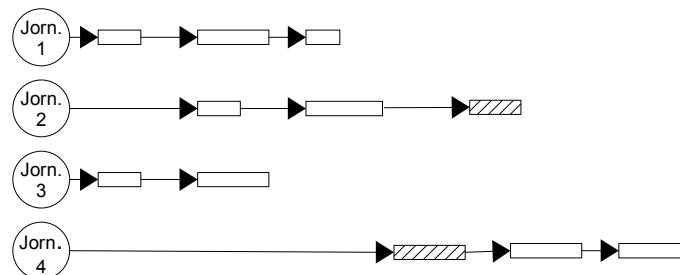


Figura 3.7 - Exemplo de execução do procedimento M2.

Os procedimentos M1 e M2 são executados repetidas vezes até que não haja melhora no custo da solução por *NumIter* iterações. A aplicação dos dois procedimentos em seqüência permite atingir uma maior cobertura do espaço de busca do que o uso de apenas um deles. A Figura 3.8 apresenta o esquema geral da fase de melhoramento.

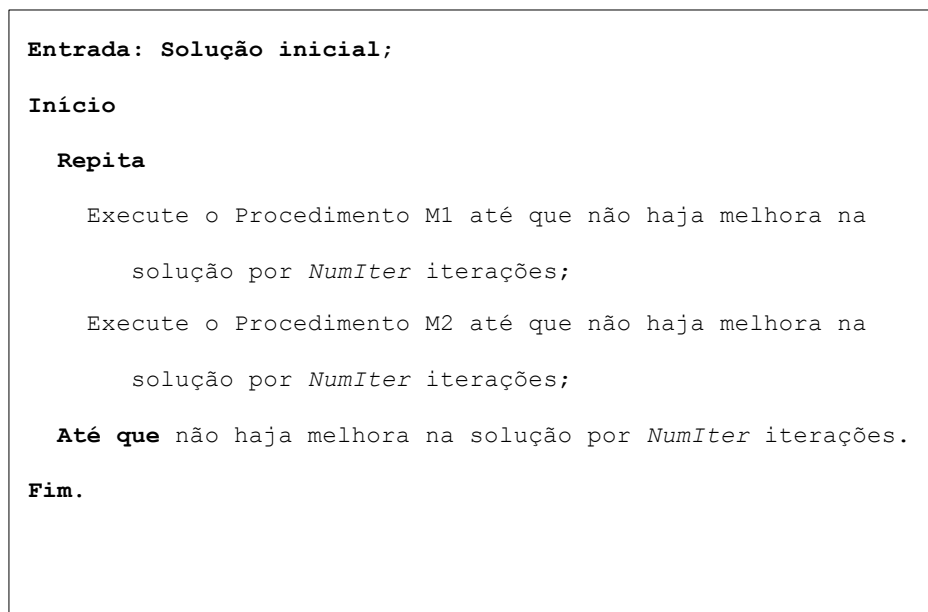


Figura 3.8 - Procedimento para melhoramento da solução.

3.3.3 Análise de Complexidade do Algoritmo Proposto

O algoritmo utilizado para resolução do problema de designação foi proposto por Carpaneto e Toth (1987) e tem complexidade $O(n^3)$, considerando que a matriz de custos que representa o problema tem dimensão $n \times n$.

Na fase de construção, em cada iteração a matriz de custos que representa o problema têm a dimensão $|J| + |T_k| \times |J| + |T_k|$, onde $|J|$ é o número total de jornadas e $|T_k|$ é o número de

tarefas na camada k . Assim a complexidade do problema de designação em cada iteração é $O((|J| + |T_k|)^3)$.

Seja n o número de tarefas do problema e considerando que $|J|$ não será maior que n , visto que não é necessário haver mais jornadas do que tarefas e que $|T_k|$ seria no máximo n caso existisse apenas uma camada, podemos determinar o limite superior em cada iteração como sendo $O((n+n)^3) = O(n^3)$. Considerando ainda que no pior caso haveriam n camadas contendo uma tarefa cada e portanto seriam executadas n iterações obtemos $O(n^4)$ como limite superior para a fase de construção.

Na fase de melhoramento cada problema de designação resolvido também terá limite superior igual a $O(n^3)$, visto que em cada iteração de M1 ou M2 a dimensão da matriz de custos será sempre menor que $n \times n$. O número de iterações irá depender de cada instância, visto que o critério de parada é não ter havido melhora após certo número de iterações. Chamando o total de vezes em que o problema de designação é resolvido na fase de melhoramento de m , podemos definir $O(mn^3)$ como um limite superior desta fase.

O limite superior do algoritmo proposto será portanto $O(n^4) + O(mn^3)$. Embora na prática m seja muito menor que n , o que permitiria estipular como limite superior $O(n^4)$, não é possível efetuar tal demonstração.

3.4 Resolução do Problema de Escalonamento de Tripulação como um Problema de Cobertura de Conjunto

Para resolução do PET como um Problema de Cobertura de Conjunto (PCC) conforme descrito na seção 2.2.1 é necessário formar os pedaços antes de gerar as colunas (jornadas) que são combinações de um número limitado de pedaços. Para gerar tais pedaços utilizou-se o método apresentado em Mayerle (1996) descrito a seguir.

Mayerle (1996) considera para cada *bloco* um grafo $H(T, P)$, onde $T = \{t_1, t_2, \dots, t_k\}$ é o conjunto de vértices correspondentes às oportunidades de troca de tripulação e P é o conjunto de arestas p_{ij} que ligam os vértices t_i e t_j . Cada aresta corresponde a um *pedaço* em potencial a ser formado pelas tarefas contidas entre t_i e t_j , como mostra a Figura 3.9.

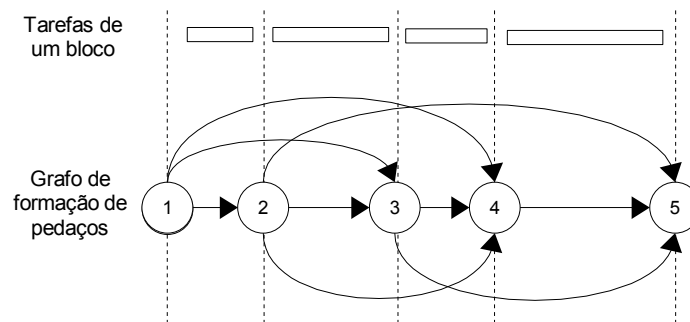


Figura 3.9 - Grafo de formação de pedaços.

A cada uma das arestas p_{ij} associa-se um custo c_{ij} dado pelo tempo de trabalho $Dur(t_i, t_j)$ compreendido entre o início e o fim do *pedaço*. Considera-se ainda os parâmetros $DurMax$ e $DurMin$, correspondendo a duração máxima e mínima de um pedaço. O custo de uma aresta do grafo $H(T, P)$ será dado por:

i) $c_{ij} = DurMin$, se $Dur(t_i, t_j) < DurMin$;

ii) $c_{ij} = +\infty$, se $Dur(t_i, t_j) > DurMax$;

iii) $c_{ij} = Dur(t_i, t_j)$ nos demais casos.

Os pedaços serão formados por encontrar-se o caminho de menor custo no grafo $H(T, P)$, entre os vértices t_1 e t_k .

Uma vez gerados os pedaços deve-se gerar as colunas combinando-se dois ou mais pedaços por coluna. Em Mayerle (1996) as colunas contém de dois a quatro pedaços.

3.5 Um Limite Inferior para o Problema de Escalonamento de Tripulação

Para obter-se um limite inferior para o PET é resolvido o modelo descrito por Bodin *et al.* (1983) para o escalonamento de trabalhadores em local fixo (veja seção 2.3). Considerou-se que existem $T = 1590$ intervalos, sendo cada intervalo com duração de um minuto. Apesar de um dia ser composto por apenas 1440 minutos considera-se a existência de 1590 intervalos devido a escala de alguns veículos começar no final de um dia e terminar no início do dia seguinte. Supor que os trabalhadores estejam em um local fixo serve de limite inferior para o PET que leva em consideração o fato das tripulações começarem/terminarem suas jornadas em diversos locais.

Visto que o número de colunas geradas para resolver o modelo de trabalhadores em local fixo seria proibitivo, foi realizada mais uma relaxação das restrições além da espacial. A fim de gerar um número de colunas pequeno considerou-se que uma jornada é composta por um período contínuo de trabalho de duração igual a 13 horas, visto que deve existir um intervalo de descanso entre um dia de trabalho e outro de 11 horas. As restrições com respeito a intervalo mínimo e máximo para descanso dentro da jornada, duração máxima do trabalho contínuo, duração máxima da jornada, número máximo de horas extras e contagem de horas de 52,5 minutos no período noturno foram simplesmente desconsideradas. Além disso cada coluna gerada com duração de 13 horas tem um custo associado igual a 440, que é o menor custo possível para uma jornada, pois mesmo que um funcionário tenha uma jornada com tempo inferior a jornada normal que é de 440 minutos irá receber o valor da jornada normal. Ao associar-se o custo de 440 para cada coluna foram desconsiderados os custos relacionados ao adicional de horas extras e adicional noturno.

Após resolver-se o modelo acima é calculado o valor do adicional noturno de 20% sobre as tarefas do horário noturno e adicionado ao custo da solução obtida, visto que não foi considerado no custo das jornadas. Este valor sempre existirá independente da solução obtida, pois todas as tarefas noturnas devem ser cobertas.

3.6 Limites Inferiores para o Problema de Cobertura de Conjunto

Para pequenas instâncias é possível obter a solução ótima do Problema de Cobertura de Conjunto (PCC) por resolver-se um modelo de Programação Linear Inteira (PLI), no entanto para instâncias maiores isto não é praticável. Porém, para estas instâncias é possível calcular um Limite Inferior (LI) para o valor ótimo. Uma maneira de obter um LI para o PCC é

resolver um problema de programação linear relaxado (PL), onde a restrição de integralidade 2.3 é substituída pela sua relaxação linear $0 \leq x_j \leq 1$, porém para problemas grandes resolver o problema de PL continua sendo impraticável devido ao grande número de variáveis e restrições.

Uma segunda alternativa de obter-se um LI para instâncias grandes do PCC é usar relaxação Lagrangeana. As restrições do PCC representadas pela equação 2.2 podem ser relaxadas e associadas a multiplicadores Lagrangeanos $\lambda_i \geq 0 (i=1, \dots, m)$, obtendo-se o problema de relaxação Lagrangeana (PLG) abaixo:

$$\text{Min} \quad \sum_{j=1}^n C_j x_j + \sum_{i=1}^m \lambda_i \quad (3.1)$$

$$\text{Sujeito a:} \quad x_j \in \{0, 1\}, j=1, 2, \dots, n \quad (3.2)$$

onde $C_j = c_j - \sum_{i=1}^m \lambda_i a_{ij}$.

O vetor (x_j) correspondendo a solução do PLG pode ser obtido por inspeção, fazendo-se,

i) $x_j = 1$, se $C_j \leq 0$;

ii) $x_j = 0$, caso contrário,

e o valor da solução do PLG será:

$$ZPLG = \sum_{j=1}^n C_j x_j + \sum_{i=1}^m \lambda_i \quad , \quad (3.3)$$

onde $ZPLG$ é um limite inferior para a solução ótima do PCC original.

O valor inicial de cada multiplicador Lagrangeano é igual ao valor de cada variável dual do problema dual do PCC, que é resolvido através de um procedimento de subida dual (*dual ascent*). O valor $ZPLG$ é calculado para esse conjunto inicial e então são executadas iterações de otimização de subgradiente para obter multiplicadores Lagrangeanos que resultem em valores melhores para $ZPLG$.

Deve-se ressaltar que o valor obtido para $ZPLG$ será no máximo igual ao valor obtido pelo modelo de PL, portanto $ZPLG$ deve ser usado apenas se a resolução do modelo de PL for impraticável.

O modelo PLG acima, bem como os procedimentos de subida dual e otimização de subgradiente são descritos em (BEASLEY, 1987) e (REEVES, 1993).

4 RESULTADOS E ANÁLISE DOS RESULTADOS

4.1 Características dos Dados

4.1.1 Características das Instâncias de Teste

Duas das instâncias utilizadas nos testes são instâncias reais e as demais são instâncias aleatórias baseadas em uma instância real. As instâncias são descritas abaixo, sendo que a parte numérica do nome indica o número de tarefas (*e.g.* RE412 contém 412 tarefas):

- RE412 e RE2313: correspondem aos dados reais de duas empresas de transporte urbano de passageiros do Estado do Paraná;
- AL130, AL251, AL512, AL761, AL1000, AL1253, AL1517 e AL2010: são instâncias aleatórias obtidas à partir da instância RE2313 sorteando-se *blocos* da mesma.

4.1.2 Descrição das Regras Trabalhistas Consideradas

A resolução do PET deve obedecer a um conjunto de restrições trabalhistas que pode variar conforme a localidade. Abaixo encontram-se listadas as regras trabalhistas impostas pela CLT (Consolidação das Leis do Trabalho), pelo acordo coletivo com o sindicato da categoria e pela política da empresa que devem ser obedecidas na resolução das instâncias consideradas neste trabalho:

- i) A jornada normal de trabalho tem duração de 7 horas e 20 minutos. Mesmo que a jornada de trabalho seja inferior a este tempo, o funcionário irá receber o valor da jornada normal, ou seja, o tempo mínimo pago por uma jornada será 7 horas e 20 minutos;

- ii) A jornada diária poderá ter duração maior que 7 horas e 20 minutos, desde que as horas extras não excedam 2 horas, ou seja, a jornada poderá ter duração de até 9 horas e 20 minutos;

- iii) As horas extras serão remuneradas com um adicional de 50% sobre o valor da hora da jornada normal;

- iv) A duração máxima do trabalho contínuo é 6 horas. Quando a duração do trabalho ultrapassar 6 horas deverá haver um intervalo para descanso de no mínimo 1 hora e 30 minutos e de no máximo 5 horas;

- v) Entre o fim da jornada em um dia e o início da jornada no dia seguinte deve haver um intervalo de no mínimo 11 horas, o que implica que a jornada terá uma extensão máxima de 13 horas considerando que o tripulante executará a mesma jornada no dia seguinte;

- vi) As horas trabalhadas no horário noturno compreendido entre as 22:00 de um dia e 05:00 do dia seguinte terão duração de 52,5 minutos;

- vii) Sobre as horas trabalhadas no horário noturno haverá um adicional de 20% sobre o valor da hora da jornada normal.

4.2 Resultados

O algoritmo proposto, os procedimentos para cálculo do limite inferior Lagrangeano (ZPLG) e para geração dos modelos de PLI ou PL foram implementados utilizando-se a ferramenta Delphi 4. O problema de designação é resolvido otimamente em tempo polinomial através do algoritmo proposto por Carpaneto e Toth (1987) que combina o método Húngaro e o método do Menor Caminho Aumentante. Para resolução dos modelos de PLI ou PL foi utilizado o resolvidor *lp-solver* de Markus Weidenauer (2003), que utiliza o algoritmo *simplex revisado*. Todos os testes computacionais foram realizados em um Pentium III de 900 Mhz com 512 MB de memória RAM e rodando o sistema operacional Windows 2000.

4.2.1 Resultados Obtidos pelo Algoritmo Proposto

A Tabela 4.1 apresenta os resultados obtidos pelo algoritmo proposto (AP) usando na construção da solução inicial as funções f_1 , f_2 e f_3 , sendo que para o parâmetro *CTL* que penaliza as trocas de linhas foi utilizado o valor 1 e para o parâmetro *NumIter* que define o número de iterações sem melhora da solução foi utilizado o valor 4. Para cada instância é mostrado o número de jornadas (NJor) na solução, o custo da solução em minutos pagos e o tempo de execução no formato minutos:segundos. A coluna LIP mostra o valor do limite inferior para o PET calculado de acordo com a seção 3.5 e Gap% é o percentual da melhor solução do AP (destacada em negrito) para cada instância em relação a LIP, ou seja,

$$Gap = (\text{Custo} - LIP) / LIP \cdot 100 \quad .$$

Tabela 4.1 – Resultados obtidos pelo algoritmo proposto

Instância	AP usando f_1			AP usando f_2			AP usando f_3			LIP	Gap%
	NJor	Solução	Tempo	NJor	Solução	Tempo	NJor	Solução	Tempo		
AL130	19	8451,3	00:38	19	8472,5	01:09	19	8389,4	00:45	8057	4,13
AL251	40	17667,5	00:36	40	17690,0	00:46	40	17600,0	00:35	15655	12,42
RE412	69	30810,0	01:05	69	30795,0	00:36	66	29512,5	00:48	25427	16,07
AL512	80	35612,5	01:08	79	35105,0	00:40	80	35312,5	01:00	30858	13,76
AL761	109	48395,0	02:22	110	48632,5	01:47	107	47532,9	01:50	43010	10,52
AL1000	152	67090,0	01:39	147	65019,4	02:10	146	64873,6	05:00	57000	13,81
AL1253	191	84580,0	02:29	188	83261,1	02:40	187	82842,9	03:36	72261	14,64
AL1517	232	102729,5	02:02	225	99852,8	05:29	227	100507,0	02:45	89191	11,95
AL2010	297	131482,5	05:21	290	128964,2	07:00	292	129637,8	03:43	116019	11,16
RE2313	339	150649,7	08:08	331	147215,0	04:37	340	150522,5	04:04	131800	11,70

O tempo computacional foi pequeno, sendo o maior tempo igual a 8min8s para a instância RE2313 utilizando-se a função f_1 . Os melhores resultados foram obtidos utilizando-se as funções f_2 e f_3 , que obtiveram o menor custo em 4 e 6 instâncias respectivamente. Os percentuais de Gap parecem elevados, mas deve-se ter em mente que o valor de LIP é muito baixo, pois para calculá-lo o problema foi bastante simplificado desconsiderando-se muitas restrições.

4.2.2 Comparação entre os Procedimentos Utilizados na Fase de Melhoramento

Conforme descrito na seção 3.3.2 foram utilizados os procedimentos M1 e M2 na fase de melhoramento do algoritmo proposto. A Tabela 4.2 mostra o quanto o custo da solução inicial pode ser reduzido utilizando-se isoladamente cada um dos procedimentos e utilizando-se os dois em conjunto. Para este teste utilizou-se os mesmos parâmetros descritos na seção 4.2.1, sendo que na fase de construção da solução inicial foi utilizada a função f_3 .

Tabela 4.2 – Comparação entre os procedimentos de melhoramento

Instância	Inicial	M1	Redução	M2	Redução	M1 + M2	Redução
RE412	33394,8	29512,5	11,63%	30897,3	7,48%	29512,5	11,63%
AL1000	74637,0	65442,0	12,32%	68550,5	8,15%	64873,6	13,08%
AL1253	95622,0	82955,4	13,25%	88486,9	7,46%	82842,9	13,36%
RE2313	172660,4	150635,0	12,76%	158088,0	8,44%	150522,5	12,82%

Na Tabela 4.2 nota-se que as soluções obtidas utilizando-se apenas M1 foram melhores que utilizando-se apenas M2. Ao usar-se os dois procedimentos em conjunto a solução foi melhor para as instâncias AL1000, AL1253 e RE2313. No caso da instância RE412 a solução foi a mesma utilizando-se M1 isoladamente ou em conjunto com M2. O maior ganho no uso conjunto de M1 e M2 foi para a instância AL1000 que obteve uma redução adicional de 0,76% em relação a redução obtida usando-se apenas M1. Embora este ganho seja percentualmente pequeno deve-se lembrar que seu valor econômico pode ser significativo.

4.2.3 Resultados Obtidos Utilizando-se o Modelo de Cobertura de Conjunto

Nesta seção são apresentados resultados para o PET modelado como um PCC. Para as instâncias menores foi possível obter o valor ótimo usando programação linear inteira, ao passo que para as instâncias maiores foram obtidos limites inferiores usando relaxação linear ou relaxação lagrangeana dependendo do tamanho da instância. Além disso, tais valores são comparados a um limite inferior do PET que independe do modelo utilizado.

A Tabela 4.3 apresenta resultados obtidos resolvendo-se o PET usando o modelo do PCC conforme descrito na seção 3.6. Os parâmetros *DurMin* e *DurMax* necessários para a formação de pedaços descrita na seção 3.4 tiveram valores iguais a 160 e 280 minutos, respectivamente. A coluna *MaxPed* mostra o número máximo de pedaços por coluna, sendo que um asterisco antes do número indica que todas as colunas possíveis foram geradas. Os

valores sem asterisco indicam que a instância poderia ter colunas com mais de *MaxPed* pedaços que não foram geradas a fim de limitar o tamanho do problema numa tentativa de tornar possível sua resolução.

Também são mostrados na Tabela 4.3 o número de pedaços formados (Linhas) e o número de Colunas obtidas pela combinação dos pedaços. A coluna ZPL traz o valor do problema de programação linear relaxado (PL), ZPLG é o valor do limite inferior Lagrangeano e ZPLI é o custo da solução inteira obtida resolvendo-se o modelo de PLI. Para obtenção de ZPL/ZPLI adotou-se um limite de tempo igual a 24 horas. Para o cálculo de ZPLG não houve limite de tempo. O valor Gap% é obtido pela fórmula $Gap\% = (Custo - LIP) / LIP \cdot 100$, onde *Custo* é o valor de ZPL para as instâncias em que foi possível obtê-lo e para as demais é o valor de ZPLG.

Tabela 4.3 – Resultados obtidos para o PCC com $DurMin = 160$ e $DurMax = 280$

Instância	MaxPed	Linhas	Colunas	ZPLI	ZPL	ZPLG	LIP	Gap%
AL130	*3	36	172	9530,60	9530,60	9528,90	8057	18,29
AL251	*4	77	1483	18691,20	18691,20	18587,40	15655	19,39
RE412	*4	136	6730	30455,00	30455,00	30228,10	25427	19,77
AL512	*5	162	16875	37112,50	37112,50	36468,69	30858	20,27
AL761	*5	234	47552	51752,50	51752,50	50242,08	43010	20,33
AL1000	*6	315	176734	67971,50	67971,50	65821,81	57000	19,25
AL1253	*6	398	314149	86349,00	86349,00	84052,23	72261	19,50
AL1517	*6	480	579666	105178,50	105178,50	102315,24	89191	17,93
AL2010	4	623	1206504	138035,40	138035,40	133055,62	116019	18,98
RE2313	4	715	1610242	-	-	149648,34	131800	13,54

Conforme apresentado na Tabela 4.3, foi possível obter o valor de ZPLI para quase todas instâncias, a exceção foi a instância RE2313 devido ao elevado número de colunas. Os valores da solução inteira (ZPLI) para estas instâncias são ótimos, pois são iguais ao valor do custo da solução do problema relaxado (ZPL).

Deve-se ressaltar que os valores de ZPLI apresentados na Tabela 4.3 são valores ótimos para o modelo de PCC gerado e não para o PET, visto que não foram geradas todas as colunas possíveis e seria inviável fazê-lo.

Os valores obtidos pela resolução do PCC (Tabela 4.2) tiveram um Gap% maior que o Gap% dos resultados do AP (Tabela 4.1) em todas as instâncias.

Com o objetivo de obter-se resultados melhores para o PCC os parâmetros *DurMin* e *DurMax* tiveram seus valores alterados para 60 e 180 minutos, respectivamente. Assim, foram gerados modelos com um número maior de colunas. Os resultados são apresentados na Tabela 4.4, cujas colunas são as mesmas da Tabela 4.3.

Para obtenção de ZPL/ZPLI adotou-se um limite de tempo igual a 24 horas. Para o cálculo de ZPLG não houve limite de tempo, sendo a única restrição a capacidade de memória do equipamento que não foi suficiente para calcular o ZPLG das instâncias maiores devido ao grande número de linhas e colunas.

Tabela 4.4 – Resultados obtidos para o PCC com *DurMin* = 60 e *DurMax* = 180

Instância	MaxPed	Linhas	Colunas	ZPLI	ZPL	ZPLG	LIP	Gap%
AL130	*4	52	2153	8807,20	8807,20	8678,13	8057	9,31
AL251	*5	104	18438	16453,30	16452,77	16216,20	15655	5,10
RE412	*5	183	113121	27521,30	27298,60	26889,98	25427	7,36
AL512	*6	218	284456	32005,00	31983,80	31669,96	30858	3,65
AL761	4	316	900914	45587,20	45583,05	44104,99	43010	2,55
AL1000	4	421	2915417	-	-	55683,23	57000	-
AL1253	4	535	6791350	-	-	71015,00	72261	-
AL1517	4	649	> 15·10 ⁶	-	-	-	89191	-
AL2010	4	842	> 38·10 ⁶	-	-	-	116019	-
RE2313	4	965	> 65·10 ⁶	-	-	-	131800	-

Conforme apresentado na Tabela 4.4, foi possível obter o valor de ZPLI apenas para as cinco instâncias menores. O valor da solução inteira (ZPLI) da instância AL130 é ótimo e das outras quatro instâncias ficou bastante próximo do ótimo, estando no pior caso 0,82% acima de ZPL (instância RE412). As instâncias maiores não puderam ser resolvidas devido ao elevado número de colunas.

4.2.4 Comparação dos Resultados do PCC com os Resultados do AP

Para uma comparação direta dos resultados obtidos pelo algoritmo proposto (AP) e pela resolução do PET usando o modelo do PCC, na Tabela 4.5 é mostrado o Afastamento Percentual Relativo (APR) do custo da melhor solução obtida por AP retirado da Tabela 4.1 em relação aos valores de ZPL (quando disponível) obtidos pelo PCC conforme Tabelas 4.3 e 4.4. Além disso, para todas as instâncias o valor obtido por AP é comparado com LIP. O APR é dado pela fórmula $APR = (Custo1 - Custo2) / Custo2 \cdot 100$, onde *Custo1* é o custo da solução de AP e *Custo2* é o valor de ZPL ou LIP, conforme a coluna da tabela.

Na Tabela 4.5 para o modelo do PCC da instância RE2313 contendo pedaços com *DurMin* e *DurMax* iguais a 160 e 280 foi utilizado o limite inferior lagrangeano ZPLG, visto que não foi possível calcular ZPL. Para o modelo do PCC das instâncias AL1000 e AL1253 contendo pedaços com *DurMin* e *DurMax* iguais a 60 e 180 o valor de ZPLG ficou abaixo do valor de LIP e portanto não foi utilizado.

Tabela 4.5 – Comparação dos resultados do AP com os resultados do PCC

Instância	ZPL	ZPL	LIP
	<i>DurMin</i> = 60 <i>DurMax</i> = 180	<i>DurMin</i> = 160 <i>DurMax</i> = 280	
AL130	-4,74%	-11,97%	4,13%
AL251	6,97%	-5,84%	12,42%
RE412	8,11%	-3,09%	16,07%
AL512	9,76%	-5,41%	13,76%
AL761	4,28%	-8,15%	10,52%
AL1000	-	-4,56%	13,81%
AL1253	-	-4,06%	14,64%
AL1517	-	-5,06%	11,95%
AL2010	-	-6,57%	11,16%
RE2313	-	-1,63%	11,70%

A Tabela 4.5 mostra que os custos das soluções AP em todas as instâncias são menores do que os custos das soluções do PCC com pedaços entre 160 e 280 minutos. Já o PCC com pedaços entre 60 e 180 minutos teve resultados melhores que o AP nas instâncias menores (exceto a AL130), porém tornou-se inviável resolver o modelo do PCC para as instâncias maiores devido ao grande número de colunas geradas.

As soluções do AP ficaram acima do limite inferior do problema (LIP) entre 4,13% e 16,07%, porém deve-se ressaltar que LIP é um limite inferior bastante baixo e que não se conhece a verdadeira distância da solução ótima.

4.2.5 Comparação com a Solução da Empresa

Uma das empresas forneceu a solução atualmente utilizada pela mesma obtida manualmente. Assim, a Tabela 4.6 compara a solução obtida por AP com a solução da empresa para a instância RE412 apresentando o número de jornadas (NJor), custo da solução em minutos pagos e o afastamento percentual em relação à solução da empresa. Na tabela a coluna AP apresenta a solução obtida por AP seguindo as regras descritas na seção 4.1.2 conforme

mostrado na Tabela 4.1 e a coluna AP2 apresenta a solução obtida utilizando-se regras similares às da empresa.

Tabela 4.6 – Comparação das soluções obtidas com a solução da Empresa

Empresa			AP			AP2		
NJor	Custo da Solução	Afastamento %	NJor	Custo da Solução	Afastamento %	NJor	Custo da Solução	Afastamento %
49	27443,0	0,00	66	29512,5	7,54	57	26984,1	-1,67

A solução da empresa foi melhor do que a solução de AP considerando as regras descritas na seção 4.1.2, pois a solução da empresa não respeita totalmente estas regras. A solução na coluna AP2 mostra que o algoritmo é capaz de produzir uma solução melhor que a da empresa caso sejam consideradas regras similares.

A Tabela 4.7 apresenta uma análise qualitativa mostrando que as soluções obtidas têm qualidade superior à da empresa, visto possuírem menos violações das regras.

Tabela 4.7 – Comparação qualitativa das soluções obtidas com a solução da Empresa

Situações observadas	Empresa	AP	AP2
Número de jornadas com trabalho contínuo > 6h	18	0	23
Número de jornadas com intervalo de descanso > 5h	14	0	0
Número de jornadas com total de horas extras > 2h	6	0	0
Número de jornadas com extensão > 13h	25	0	12
Total de violações das regras	63	0	35
Máximo de horas extras em uma jornada	2h27min	2h	2h
Intervalo mínimo dentro de uma jornada	0h	1h30min	1h30min
Intervalo máximo dentro de uma jornada	10h15min	5h	5h
Maior extensão de uma jornada	18h05min	13h	15h
Maior duração de trabalho contínuo	8h39min	6h	8h

Embora RE2313 também seja um problema real não foi possível uma comparação com a solução da empresa, visto que a mesma não forneceu sua solução.

4.3 Análise dos Resultados

Em termos de desempenho computacional o algoritmo proposto (AP) foi bastante eficiente. Quanto a qualidade das soluções obtidas, os valores variaram entre 4,13% e 16,07% acima do Limite Inferior para o PET (LIP). Apesar de alguns percentuais parecerem elevados deve-se levar em consideração que não se conhece a verdadeira distância do limite inferior à solução ótima e que provavelmente o limite inferior está bastante abaixo da solução ótima, visto que foram relaxadas muitas restrições do problema.

Gerar instâncias do PCC com pedaços mais longos ($DurMin=160$ e $DurMax=280$) resultou em um número menor de pedaços e conseqüentemente colunas, tornando possível obter uma solução inteira ótima para problemas com até 2010 tarefas, no entanto os resultados obtidos foram piores que os obtidos por AP. Para o problema maior com 2313 tarefas, mesmo gerando-se menos colunas não foi possível obter uma solução inteira, no entanto o limite inferior calculado mostra que se fosse obtida a solução ótima esta seria pior que o valor obtido por AP. A Tabela 4.4 mostra que AP obteve valores entre 1,63% e 11,97% abaixo dos resultados destas instâncias do PCC.

Gerar instâncias do PCC com pedaços mais curtos ($DurMin=60$ e $DurMax=180$) resulta em um número maior de pedaços e conseqüentemente colunas obtendo-se valores ótimos ou próximos do ótimo para as instâncias menores com até 761 tarefas, no entanto este aumento de colunas tornou impraticável a resolução dos problemas maiores. A Tabela 4.4 mostra que as soluções do AP para as instâncias AL251, CV 412, AL512 e AL761, ficaram 6,97%, 8,11%, 9,76% e 4,28% acima das soluções do PCC, respectivamente. Para as instâncias

maiores do PCC não foi possível calcular limites inferiores devido ao grande número de colunas.

A solução obtida pelo AP para AL130 ficou 4,74% abaixo da solução ótima do PCC. Isto pode ser explicado pelo fato de que as instâncias do PCC geradas não são um modelo exato do PET, assim a solução ótima do PCC não é necessariamente a solução ótima do PET. Para que o PCC fosse um modelo exato garantindo a solução ótima para o PET seria necessário gerar todas as colunas possíveis para o conjunto de tarefas sem gerar pedaços previamente, visto que estes limitam as possibilidades de colunas, o que seria impraticável. Assim, pode-se dizer que o AP produziu uma solução melhor para esta instância porque sua solução poderia conter jornadas que não estariam no PCC, visto que o AP não forma pedaços previamente.

5 CONCLUSÕES

5.1 Conclusões

Limitar o número de pedaços e conseqüentemente colunas a fim de tornar possível a resolução do PET formulado como um PCC compromete a qualidade da solução, conforme observado nos testes com pedaços mais longos. Aumentar o número de colunas para conseguir um resultado melhor só é viável para os problemas com poucas tarefas, ao passo que fazer isso para os grandes problemas torna a resolução do PCC impraticável.

O algoritmo proposto resolve tanto problemas pequenos quanto grandes em um tempo bastante curto e com uma qualidade aceitável para as grandes instâncias, visto que seria difícil alcançar soluções melhores para tais.

Assim, o algoritmo proposto pode ser considerado uma boa opção para resolver grandes instâncias do PET, visto que para tais o grande número de colunas torna a resolução do PCC impraticável e a limitação do número de colunas para tornar sua resolução possível compromete a qualidade da solução, ao passo que para as pequenas instâncias a resolução exata do PCC é recomendada visto que pode-se obter uma economia significativa.

5.2 Recomendações para Trabalhos Futuros

Recomenda-se para trabalhos futuros:

- a) a criação de novas heurísticas para a fase de melhoramento;
- b) e a modificação do algoritmo proposto de modo a permitir a resolução integrada dos problemas de escalonamento de veículos e escalonamento de tripulação.

REFERÊNCIAS BIBLIOGRÁFICAS

- BALL, M.; BODIN, L.; DIAL, R. A matching based heuristic for scheduling mass transit crews and vehicles. *Transportation Science*, v. 17, n. 1, p. 4-31, 1983.
- BEASLEY, J. E. An algorithm for set covering problem. *European Journal of Operational Research*, v. 31, n. 1, p. 85-93, jul. 1987.
- BEASLEY, J. E; CHU, P. C. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, v. 94, n. 2, p. 392-404, out. 1996.
- BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M.. Routing and scheduling of vehicles and crews - The state of the art. *International Journal of Computers and Operations Research*, v. 10, n. 2, p. 63-211, 1983.
- CAPRARA, A.; FISCHETTI, M.; TOTH P. A heuristic method for the set covering problem. *Operations Research*, v. 47, n. 5, p. 730-743, set./out. 1999.
- CARPANETO, G.; TOTH, P. Primal-dual algorithms for the assignment problem. *Discrete Applied Mathematics*, n. 18, p. 137-153, North-Holland, 1987.
- FORSYTH, P.; WREN, A. *An ant system for bus driver scheduling*, Report 97.25, Research Report Series, School of Computer Studies, University of Leeds, 1997. Disponível em: <http://www.comp.leeds.ac.uk/research/pubs/reports/1997/97_25.ps.Z>. Acesso em: 01 jul. 2004.
- FRELING, R.; HUISMANN, D.; WAGELMANS, A. P. M. Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, v. 6, n. 1, p. 63-85, jan./fev. 2003.
- FRELING, R.; PAIXÃO, J. M. P.; WAGELMANS, A. P. M.. *Models and algorithms for single depot vehicle scheduling*, 1999. Disponível em: <<http://www.few.eur.nl/few/people/wagelmans/publicat.htm>>. Acesso em: 01 abr. 2003.
- GOLDBARG, M. C.; LUNA, H. P. L. *Otimização combinatória e programação linear: modelos e algoritmos*, 4.ed. Rio de Janeiro: Campus, 2000.
- GROOT S. W.; HUISMAN, D. *Vehicle and crew Scheduling: solving large real world instances with an integrated approach*, Econometric Institute Report EI2004-13, 2004. Disponível em: <<https://ep.eur.nl/retrieve/2760/ei200413.pdf>>. Acesso em: 14 jun. 2004.
- HUISMAN, D.; FRELING, R.; WAGELMANS, A. P. M. *Multiple-depot integrated vehicle and crew scheduling*, Econometric Institute Report EI2003-02, 2003. Disponível em: <<http://www.eur.nl/WebDOC/doc/econometrie/feweco20030217143604.pdf>>. Acesso em: 14 jun. 2004.
- LI, J.; KWAN, R. S. K. *A fuzzy simulated evolution algorithm for the driver scheduling problem*, Report 2000.27, Research Report Series, School of Computing, University of Leeds,

2000. Disponível em:

<http://www.comp.leeds.ac.uk/research/pubs/reports/2000/2000_27.ps.gz>. Acesso em: 14 jun. 2004.

MAURI, G. R.; LORENA, L. A. N. Método iterativo para resolução do problema de escalonamento de programação de tripulações. In: SBPO, n. 36, 2004, São João Del Rei. *Anais...* São João Del Rei: UFSJ, 2004. p. 30-41.

MAYERLE, S. F. *Um sistema de apoio à decisão para o planejamento operacional de empresas de transporte rodoviário urbano de passageiros*. 1996. Tese (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis.

OHLSSON, M.; PETERSON, C.; SÖDERBERG, B. An efficient mean field approach to the set covering problem. *European Journal of Operational Research*, v. 133, n. 3, p. 583-595, set. 2001.

REEVES, C. R. *Modern heuristic techniques for combinatorial problems*, Oxford: Blackwell Scientific Publications, 1993.

SHEN, Y.; KWAN, R. S. K. *Tabu search for driver scheduling*, Report 2000.10, Research Report Series, School of Computer Studies, University of Leeds, 2000. Disponível em: <http://www.comp.leeds.ac.uk/research/pubs/reports/2000/2000_10.ps.gz>. Acesso em: 14 jun. 2004.

SIQUEIRA, P. H. *Aplicação do algoritmo do matching no problema da construção de escalas de motoristas e cobradores de ônibus*. 1999. Dissertação (Mestrado em Ciências) – Universidade Federal do Paraná.

SOUZA, M. J. F.; CARDOSO, L. X. T.; SILVA, G. P. Programação de tripulações de ônibus urbano: uma abordagem heurística. In: SBPO, n. 35, 2003, Natal. *Anais...* Natal: UFRN, 2003. p. 1285-1294.

WAIDENAUER, M. *Lp-solver*, 2003. Disponível em: <<http://www.netcologne.de/~nc-weidenma/index.htm>>. Acesso em: 1 mar. 2003.

WREN, A.; WREN, D. A genetic algorithm for public transport driver scheduling. *Computers Ops Res.*, v. 22, n. 1, p. 101-110, 1995.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)