



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE

NOVAS HEURÍSTICAS PARA O PROBLEMA DE ESCALONAMENTO DE TRIPULAÇÕES

Geraldo Regis Mauri

Dissertação de Mestrado do Curso de Pós-graduação em Computação Aplicada,
orientada pelo Dr. Luiz Antonio Nogueira Lorena, aprovada em 23 de fevereiro de 2005.

INPE

São José dos Campos

2005

...(...)

MAURI, G. R.

Novas Heurísticas para o Problema de Escalonamento de Tripulações / G. R. Mauri. – São José dos Campos: INPE, 2005.

..... – (INPE-.....-.../...).

1. Algoritmo de Treinamento Populacional. 2. Escalonamento de Tripulações. 3. Heurísticas. 4. Geração de Colunas. 5. Programação Linear.

Aprovada pela Banca Examinadora
em cumprimento a requisito exigido
para a obtenção do Título de Mestre
em Computação Aplicada.

Dr. Solon Venâncio de Carvalho

Presidente

Dr. Luiz Antonio Nogueira Lorena

Orientador

Dr. Edson Luiz França Senne

Membro da Banca

Dr. Marcone Jamilson Freitas Souza

Membro da Banca
– convidado –

Candidato: Geraldo Regis Mauri

São José dos Campos, 23 de fevereiro de 2005.

“Nunca ande pelo caminho traçado, pois ele conduz somente até onde os outros foram”.

ALEXANDRE GRAHAM BELL

Aos meus pais,

PEDRO BRAZ MAURI e

MARIA CÂNDIDA RESENDE REGIS MAURI.

AGRADECIMENTOS

A Deus, pela saúde, proteção e oportunidade de chegar até aqui.

A meus pais, por sempre acreditarem na importância do estudo.

À minha família, em especial à minha avó Tereza, pelo apoio e incentivo durante todas as etapas da vida.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo auxílio financeiro que viabilizou o desenvolvimento deste trabalho.

Aos professores do INPE, pelo conhecimento compartilhado.

Ao orientador Professor Doutor Luiz Antonio Nogueira Lorena, pelos conhecimentos transmitidos, e pela orientação e apoio na realização deste trabalho.

Ao Professor Doutor Marcone Jamilson Freitas Souza (UFOP), pelo incentivo à realização do mestrado.

Aos grandes amigos, companheiros e “assaltados”, Kabron, Sidão, Brieba, Tchaka, Apolônio, e High Level. E também aos outros, não “assaltados”, Peixe, Tainha, Sardinha, BB de Itú e Chuck Norris.

Aos tantos amigos que encontrei, pelo convívio, pelos momentos de descontração, conselhos e pela troca de experiências.

À Larice, pelo amor e carinho, pela compreensão e apoio, e principalmente por, mesmo à distância, estar sempre ao meu lado, e pelos momentos inesquecíveis da minha vida.

Aos irmãos “Canalhas”, pela amizade e por compartilharem vários momentos durante a realização do mestrado.

À grandiosa e eterna República VIRA SAIA, por me ajudar a chegar onde cheguei, e pelo aprendizado e convivência que tenho até hoje e espero ter pelo resto da vida.

A todos os demais, que de alguma forma contribuíram para a realização deste trabalho.

RESUMO

O problema de escalonamento de tripulações consiste na atribuição da tarefa de condução dos veículos às tripulações, de tal forma que todas as viagens realizadas por uma empresa sejam executadas com o menor custo possível. Esse processo é uma etapa de fundamental importância no planejamento das operações para uma empresa do setor de transporte coletivo e pertence à classe de problemas de difícil solução. Vários métodos têm sido aplicados para resolver tal problema, mostrando resultados bem variados. Este trabalho faz uma revisão de alguns trabalhos correlatos encontrados na literatura, e apresenta uma metodologia híbrida, ainda não explorada, denominada ATP/PL, para resolver tal problema, com o objetivo de alcançar soluções variadas para problemas que considerem características de problemas reais. A ATP/PL é baseada na técnica de Geração de Colunas e é implementada através da interação entre o Algoritmo de Treinamento Populacional com a Programação Linear. Além disso, as soluções obtidas por tal metodologia são comparadas com soluções obtidas pela metaheurística *Simulated Annealing*, que por sua vez apresentou excelentes resultados para problemas reais em trabalhos anteriores.¹

¹Este trabalho contou com apoio financeiro da FAPESP (processo 03/04547-2).

NEW HEURISTICS TO CREW SCHEDULING PROBLEM

ABSTRACT

The crew scheduling problem consists in attributing to the crews the job of driving vehicles, in such a way that trips of different lines assisted by a company are executed with the smallest possible cost. This process is of fundamental importance in planning the operations for the public transportation sector and belongs to a class of difficult problems. Several methods have been applied to this problem presenting different results. This work reviews some of them presented in the literature, and presents a hybrid methodology, not yet explored, named ATP/PL, to solve such problem, searching varied solutions for problems considering real characteristics. The ATP/PL is based in a Column Generation technique and implemented through Population Training Algorithm with Linear Programming interaction. Besides, the obtained solutions are compared against solutions obtained by the Simulated Annealing metaheuristic, that had presented good results for real problems in previous works. ¹

¹This work had the financial support of FAPESP (process 03/04547-2).

SUMÁRIO

	Pág.
LISTA DE FIGURAS	
LISTA DE TABELAS	
CAPÍTULO 1 INTRODUÇÃO	21
CAPÍTULO 2 DESCRIÇÃO DO PROBLEMA	23
CAPÍTULO 3 REVISÃO BIBLIOGRÁFICA	27
3.1 PET via <i>Simulated Annealing</i>	31
CAPÍTULO 4 MÉTODOS DE SOLUÇÃO	35
4.1 Busca Local	35
4.2 <i>Simulated Annealing</i>	36
4.3 Programação Linear	37
4.4 Geração de Colunas	38
4.5 Algoritmos Genéticos	39
4.6 Algoritmo Genético Construtivo	40
4.7 Algoritmo de Treinamento Populacional	40
CAPÍTULO 5 TRATAMENTO DO PROBLEMA	43
5.1 ATP/PL	45
5.2 <i>Simulated Annealing</i>	51
5.3 Instâncias Utilizadas	51
CAPÍTULO 6 ANÁLISE DA ATP/PL	53
CAPÍTULO 7 RESULTADOS COMPUTACIONAIS	65
CAPÍTULO 8 CONCLUSÃO	73
REFERÊNCIAS BIBLIOGRÁFICAS	75
APÊNDICE A PARÂMETROS E INFORMAÇÕES DAS ATP/PL's	83
APÊNDICE B PARÂMETROS E INFORMAÇÕES DAS SA's	99

LISTA DE FIGURAS

	Pág.
2.1 Planejamento comum no setor de transporte.	23
2.2 Representação do bloco de um veículo.	24
2.3 Representação de um turno.	25
3.1 Movimento <i>Realocar Tarefa</i>	34
4.1 Algoritmo <i>Simulated Annealing</i>	36
4.2 Algoritmo de Treinamento Populacional (ATP).	41
5.1 Representação do problema.	43
5.2 Heurística de treinamento.	46
5.3 Mutação.	47
5.4 Cruzamento.	47
5.5 População inicial direcionada ao PPC.	49
5.6 Algoritmo ATP/PL.	50
5.7 Formatação das instâncias utilizadas.	52
6.1 Função objetivo da ATP/PL ₁ durante a interação ATP e PL.	54
6.2 Número de colunas da ATP/PL ₁ durante a interação ATP e PL.	54
6.3 Função objetivo da ATP/PL ₂ durante a interação ATP e PL.	55
6.4 Número de colunas da ATP/PL ₂ durante a interação ATP e PL.	55
6.5 Função objetivo da ATP/PL ₃ durante a interação ATP e PL.	56
6.6 Número de colunas da ATP/PL ₃ durante a interação ATP e PL.	56
6.7 Função objetivo da ATP/PL ₄ durante a interação ATP e PL.	57

6.8	Número de colunas da ATP/PL_4 durante a interação ATP e PL.	57
6.9	Função objetivo da ATP/PL_5 durante a interação ATP e PL.	58
6.10	Número de colunas da ATP/PL_5 durante a interação ATP e PL.	58
6.11	Função objetivo da ATP/PL_6 durante a interação ATP e PL.	59
6.12	Número de colunas da ATP/PL_6 durante a interação ATP e PL.	59
6.13	Função objetivo da ATP/PL durante a interação ATP e PL.	60
6.14	Número de colunas da ATP/PL durante a interação ATP e PL.	60
6.15	Função objetivo da ATP/PL_7 durante a interação ATP e PL.	61
6.16	Número de colunas da ATP/PL_7 durante a interação ATP e PL.	62
6.17	Função objetivo da ATP/PL_8 durante a interação ATP e PL.	62
6.18	Número de colunas da ATP/PL_8 durante a interação ATP e PL.	63
6.19	Número de colunas geradas pelas metodologias ATP/PL's.	63
6.20	Custo das soluções obtidas.	64

LISTA DE TABELAS

	Pág.
6.1 Soluções obtidas pelas ATP/PL's.	64
7.1 Resultados obtidos para a primeira abordagem do PET.	65
7.2 Desvio obtido com 25 tarefas.	66
7.3 Desvio obtido com 50 tarefas.	66
7.4 Desvio obtido com 100 tarefas.	67
7.5 Desvio obtido com 250 tarefas.	67
7.6 Desvio obtido com 500 tarefas.	67
7.7 Resultados obtidos com 25 tarefas para a segunda abordagem do PET. . .	68
7.8 Resultados obtidos com 50 tarefas para a segunda abordagem do PET. . .	68
7.9 Resultados obtidos com 100 tarefas para a segunda abordagem do PET. . .	69
7.10 Resultados obtidos com 250 tarefas para a segunda abordagem do PET. . .	70
7.11 Resultados obtidos com 500 tarefas para a segunda abordagem do PET. . .	70
A.1 Parâmetros e informações da ATP+PL.	83
A.2 Parâmetros e informações da ATP/PL_1.	84
A.3 Parâmetros e informações da ATP/PL_2.	85
A.4 Parâmetros e informações da ATP/PL_3.	86
A.5 Parâmetros e informações da ATP/PL_4.	87
A.6 Parâmetros e informações da ATP/PL_5.	88
A.7 Parâmetros e informações da ATP/PL_6.	89
A.8 Parâmetros e informações da ATP/PL.	90

A.9	Parâmetros e informações da ATP/PL_7.	91
A.10	Parâmetros e informações da ATP/PL_8.	92
A.11	Parâmetros da ATP/PL para todos os experimentos.	93
A.12	Experimentos com 25 tarefas para a primeira abordagem do PET.	93
A.13	Experimentos com 50 tarefas para a primeira abordagem do PET.	94
A.14	Experimentos com 100 tarefas para a primeira abordagem do PET.	94
A.15	Experimentos com 250 tarefas para a primeira abordagem do PET.	94
A.16	Experimentos com 500 tarefas para a primeira abordagem do PET.	95
A.17	Experimentos com 25 tarefas para a segunda abordagem do PET.	95
A.18	Experimentos com 50 tarefas para a segunda abordagem do PET.	96
A.19	Experimentos com 100 tarefas para a segunda abordagem do PET.	96
A.20	Experimentos com 250 tarefas para a segunda abordagem do PET.	97
A.21	Experimentos com 500 tarefas para a segunda abordagem do PET.	97
B.1	Parâmetros da SA e SA_20 para todos os experimentos.	99
B.2	Experimentos da SA para a primeira abordagem do PET.	99
B.3	Experimentos da SA_20 para a primeira abordagem do PET.	100
B.4	SA com 25 tarefas para a segunda abordagem do PET.	100
B.5	SA com 50 tarefas para a segunda abordagem do PET.	100
B.6	SA com 100 tarefas para a segunda abordagem do PET.	101
B.7	SA com 250 tarefas para a segunda abordagem do PET.	101
B.8	SA com 500 tarefas para a segunda abordagem do PET.	101
B.9	SA_20 com 25 tarefas para a segunda abordagem do PET.	102
B.10	SA_20 com 50 tarefas para a segunda abordagem do PET.	102

B.11 SA_20 com 100 tarefas para a segunda abordagem do PET.	102
B.12 SA_20 com 250 tarefas para a segunda abordagem do PET.	103
B.13 SA_20 com 500 tarefas para a segunda abordagem do PET.	103

CAPÍTULO 1

INTRODUÇÃO

Desde a década de 60, vários autores vêm se dedicando ao desenvolvimento de modelos capazes de gerar escalas de tripulações que satisfaçam às restrições impostas, e que possuam o menor custo possível. Este é um problema permanentemente estudado, uma vez que as realidades dos sistemas de transporte estão em contínua transformação e exigem cada vez mais uma gerência eficiente dos recursos disponíveis. Prova disso é a existência de congressos especializados sobre o tema nas últimas décadas, como por exemplo, a *International Conference on Computer-Aided Scheduling of Public Transport (CASPT)* (Wilson, 1999; Shen e Kwan, 2001; Wren, 2004; Kwan et al., 2004; Gintner et al., 2004; Dallaire et al., 2004).

Os sistemas heurísticos foram os primeiros a serem utilizados na resolução do Problema de Escalonamento de Tripulações (PET) (Elias, 1964). Tais sistemas consistiam apenas na automação do trabalho antes realizado manualmente, não detectando assim possibilidades de otimização. Manington e Wren (1975) iniciaram a inclusão de procedimentos de otimização neste tipo de sistema (Wren e Rousseau, 1995).

Com o surgimento de metaheurísticas tais como *Algoritmos Genéticos* (Goldberg, 1989), *Busca Tabu* (Glover, 1989a,b), *Simulated Annealing* (Kirkpatrick et al., 1983) entre outras, abriu-se um novo horizonte na resolução de problemas *NP-hard* como o PET (ver Lourenço et al. (2001); Mauri (2003); Marinho et al. (2004); Groot e Huisman (2004); Lee (2004)). Embora tais métodos não garantam a obtenção do ótimo global, eles permitem incluir com facilidade vários tipos de restrições. Além desses métodos, alguns outros baseados em técnicas de geração de colunas (ver Desrochers e Soumis (1989); Wilhelm (2001); Dallaire et al. (2004)) e de programação matemática (ver Beasley e Cao (1996); Wren et al. (2003)) também mostraram bons resultados para a solução do PET.

Apesar de esse problema ter sido largamente estudado e aplicado nos países mais desenvolvidos, suas técnicas de resolução são ainda pouco difundidas e raramente aplicadas à nossa realidade. Isso se deve às condições em que atuam as empresas de transporte público no Brasil e, principalmente, às restrições quanto ao cumprimento das legislações trabalhistas e às próprias normas impostas pelas empresas do setor, que dificultam ou até impossibilitam que uma metodologia aplicada em outro país possa ser adaptada à nossa realidade (Mauri, 2003).

Atualmente, com a crescente competitividade, as empresas devem aprimorar cada vez

mais seus processos produtivos, tentando manter a qualidade de seus produtos e/ou serviços e conseqüentemente reduzir seus custos. Com esse intuito, elas devem procurar utilizar, de maneira eficiente, seus recursos materiais e humanos para se manterem lucrativas, sem, no entanto, comprometer a qualidade dos seus serviços oferecidos.

Além disso, a democratização do acesso ao sistema de transporte público no Brasil depende da garantia de tarifas suportáveis para os usuários, e esta depende tanto do aumento da eficiência e da redução dos custos operacionais, quanto da concessão de subsídios em situações específicas, sob controle social (ANTP, 2005). E, como é sabido que nas empresas de transporte coletivo a mão-de-obra operacional representa um dos maiores custos (Bouzada, 2002), uma pequena redução neste item pode significar um ganho considerável no custo total, podendo reduzir significativamente o valor das tarifas para o usuário final, justificando assim qualquer trabalho no sentido de minimizar os custos com a mão-de-obra.

Este trabalho, como vários outros encontrados na literatura, (Forsyth e Wren, 1997; Lourenço et al., 2001; Shen e Kwan, 2001; Souza et al., 2003; Wren et al., 2003; Marinho et al., 2004), aborda o que diz respeito ao setor de transporte rodoviário, ou seja, o problema de escalonamento de tripulações (motoristas e cobradores) para ônibus urbano.

Neste trabalho, para solucionar tal problema, é apresentada uma nova metodologia, chamada de **ATP/PL**, baseada na aplicação do Algoritmo de Treinamento Populacional (ATP) juntamente com outras técnicas (Geração de Colunas, Programação Linear, e outras heurísticas), com o objetivo de gerar escalas viáveis e de baixo custo para as tripulações de empresas do setor de transporte rodoviário. Além disso, este trabalho propõe a utilização de técnicas empregadas em problemas que considerem características de problemas reais, e uma comparação do desempenho dessa metodologia ao da metaheurística *Simulated Annealing*.

O restante do trabalho está organizado da seguinte forma. O Capítulo 2 faz uma descrição minuciosa sobre o PET. O Capítulo 3 faz uma breve revisão bibliográfica sobre outras metodologias utilizadas para solucionar o PET. No Capítulo 4 são descritas, brevemente, algumas técnicas que são utilizadas neste trabalho. No Capítulo 5 é descrito, em detalhes, o tratamento do PET e as metodologias utilizadas para resolvê-lo. No Capítulo 6 é feita uma análise da ATP/PL, apresentando algumas características e peculiaridades notadas durante seu desenvolvimento. O Capítulo 7 apresenta os resultados computacionais obtidos pela ATP/PL e pela *Simulated Annealing*. Finalmente, o Capítulo 8 apresenta algumas conclusões a respeito da ATP/PL e do trabalho como um todo, e sugere possíveis extensões.

CAPÍTULO 2

DESCRIÇÃO DO PROBLEMA

O planejamento no setor de transporte é uma etapa complexa e fundamental para as empresas do setor, sendo então dividido em diversos subproblemas (ver Figura 2.1).

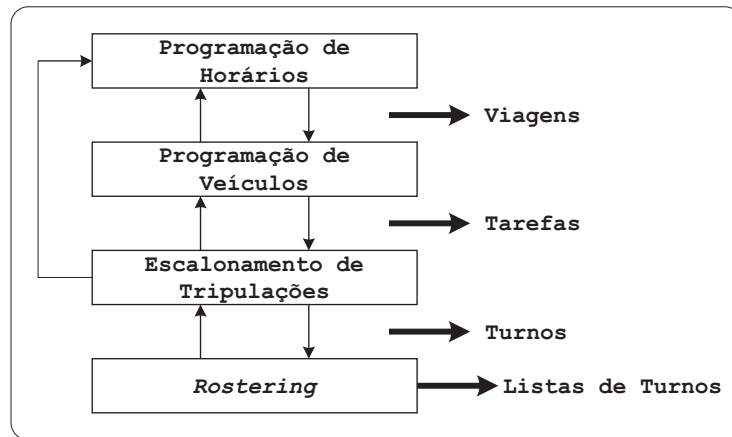


FIGURA 2.1 – Planejamento comum no setor de transporte.

FONTE: adaptada de Lourenço et al. (2001, p. 3).

Esse planejamento é composto por um conjunto de linhas, identificadas por um número, que corresponde à viagem de um veículo entre duas localidades. Para cada linha, a respectiva frequência é determinada de acordo com a demanda. A partir das linhas, uma programação de horários é construída, resultando em viagens que correspondem a uma localidade de início e uma de fim. A programação de veículos atribui veículos às viagens, e o escalonamento de tripulações gera turnos que serão executados durante um dia de trabalho. E por fim, o *rostering* constrói escalas, normalmente mensais, que são formadas pelos turnos formados na fase anterior juntamente com folgas, feriados, férias das tripulações e algumas restrições.

O problema de escalonamento de tripulações, de uma forma geral, consiste na atribuição da tarefa de condução de veículos às tripulações (motoristas e cobradores). Esse problema é geralmente conhecido, na América do Norte, como *run-cutting*, e como *crew scheduling* na maioria dos países europeus (Wren e Rousseau, 1995). Entretanto, várias outras nomenclaturas são encontradas na literatura, sendo que dentre todas essas, as mais utilizadas são *Bus Driver Scheduling* (Dias et al., 2001; Forsyth e Wren, 1997; Lourenço et al., 2001; Wren e Rousseau, 1995) e *Driver Scheduling* (Kwan et al., 1999; Shen e Kwan, 2001; Wren et al., 2003).

No Brasil, apesar de tal problema não ser muito estudado, alguns pesquisadores da Universidade Estadual de Campinas (ver Yunes et al. (1999); Yunes (2000); Yunes et al. (2000, 2001)), da Universidade Federal de Ouro Preto e da Universidade Federal Fluminense (ver Souza et al. (2003); Mauri (2003); Marinho et al. (2004)) vêm desenvolvendo trabalhos interessantes, que por sua vez têm apresentado bons resultados para problemas reais. Esses pesquisadores denominam tal problema como *Crew Scheduling Problem (CSP)* ou *Problema de Programação de Tripulações (PPT)*. Entretanto, por uma questão de identificação, tal problema será tratado, neste trabalho, como *Problema de Escalonamento de Tripulações (PET)*.

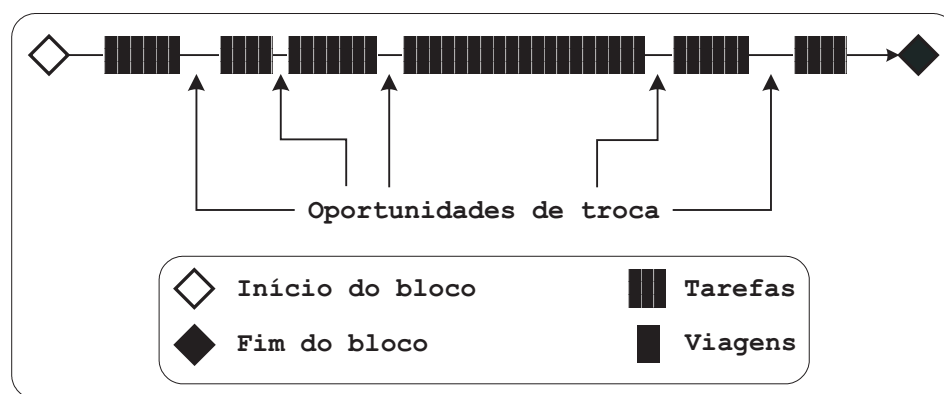


FIGURA 2.2 – Representação do bloco de um veículo.

No transporte coletivo rodoviário (como ilustrado na Figura 2.1) usualmente o escalonamento de tripulações é feito após a programação dos veículos. Nesta, as viagens são reunidas em *Blocos* (ver Figura 2.2), onde um bloco apresenta a seqüência de viagens distribuídas sucessivamente a um ônibus começando e terminando na garagem. Cada bloco mostra também as *Oportunidades de Troca* (*Relief Opportunities* segundo Forsyth e Wren (1997); Lourenço et al. (2001); Shen e Kwan (2001); Wren e Rousseau (1995); Wren et al. (2003); Wren (2004) ou *Relief Points* segundo Desaulniers (2002); Desrochers e Soumis (1989); Dallaire et al. (2004)), que são intervalos de tempo suficiente para haver a troca das tripulações (ver Figura 2.2).

A partir das viagens dos blocos dos veículos são formadas as *tarefas* (ver Figura 2.2) (*tasks* segundo Beasley e Cao (1996); Desaulniers (2002) ou *pieces of work* segundo Forsyth e Wren (1997); Lourenço et al. (2001); Wren et al. (2003)), a partir das quais são criados os *Turnos* (ver Figura 2.3), também chamados por Mauri (2003) de *Jornadas Diárias* (*shift* segundo Forsyth e Wren (1997); Desaulniers (2002); Valouxis e Housos (2002); Wren et al. (2003), ou *duty* segundo Lourenço et al. (2001); Shen e Kwan (2001)), que são atribuídos às tripulações, as quais irão executá-los durante um dia de trabalho.

Esses turnos, de uma forma geral, são classificados em dois tipos. O primeiro é aquele em que as tripulações trabalham diretamente sem um intervalo ou têm um pequeno intervalo para satisfazer as regras impostas. Os turnos desse tipo são freqüentemente chamados *Turnos Diretos* (jornadas do tipo *Pegada Simples*, segundo Mauri (2003)). Já o segundo tipo, que é usado pela maioria das companhias de ônibus, consiste naquele onde as tripulações têm dois períodos de trabalho, usualmente centrados nos períodos máximos, separados por um “intervalo” de várias horas. Turnos desse tipo são chamados de *Turnos Rachados* (jornadas do tipo *Dupla Pegada*, segundo Mauri (2003) e Yunes et al. (2001)).

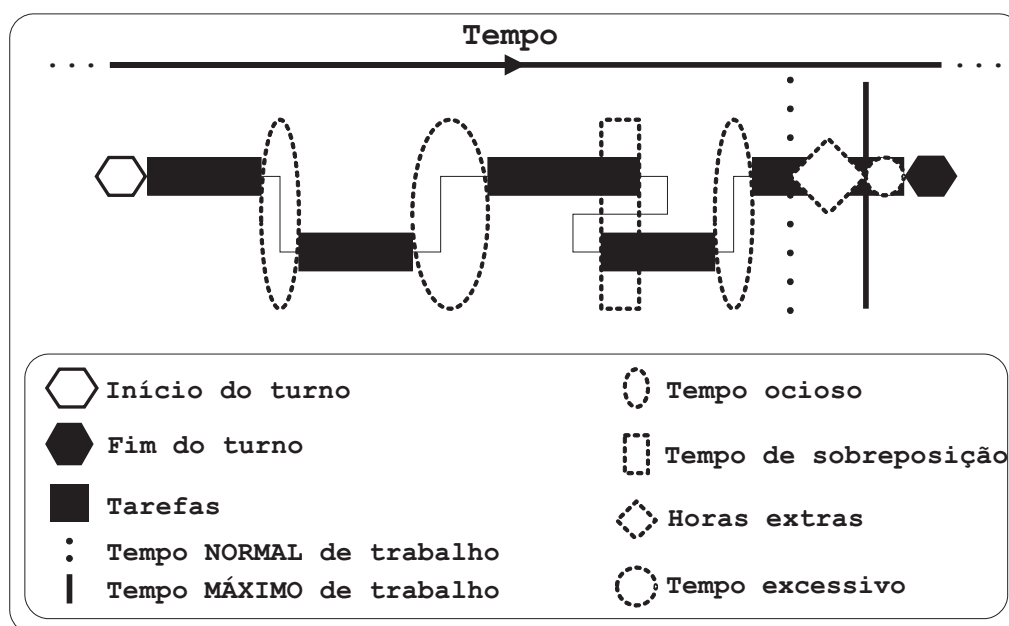


FIGURA 2.3 – Representação de um turno.

O processo de formação dos turnos está sujeito a um conjunto de regras que são específicas a uma organização. Essas regras são geralmente combinadas de regras nacionais e locais, podendo ser obrigatórias ou não. Tipicamente, há restrições no tempo máximo trabalhado, no tempo que pode ser trabalhado sem um intervalo para almoço, na extensão total do turno (duração entre o início e o fim do turno), etc.

Algumas dessas regras podem ser vistas na Figura 2.3, onde *Tempo NORMAL de trabalho* refere-se ao tempo “normal” de duração de um turno (sem horas extras), *Tempo MÁXIMO de trabalho* refere-se ao tempo máximo de duração de um turno (com horas extras), *Tempo ocioso* indica o tempo em que uma tripulação não está trabalhando durante seu turno, *Tempo de sobreposição* indica o tempo em que uma tripulação está, teoricamente, realizando duas tarefas ao mesmo tempo, *Horas extras* indica o tempo em

que o turno de uma tripulação excede o tempo normal de trabalho, e *Tempo Excessivo* refere-se ao tempo em que um turno excede o tempo máximo de trabalho.

A maioria dos países desenvolvidos tem leis trabalhistas que requerem um intervalo para almoço após um período máximo entre quatro horas e meia e cinco horas e meia. E ainda, a comunidade europeia tem um complexo conjunto de regras que proporcionam a qualquer momento um simples intervalo (de 30 minutos) ou dois intervalos menores, ou três intervalos ainda menores, ou um turno sem nenhum intervalo formal, proporcionando a folga cumulativa exigida entre os turnos. Apesar disso, muitas, se não a maioria, das companhias de ônibus na Europa obedecem a regras locais estipulando as providências necessárias para o intervalo para o almoço (Wren e Rousseau, 1995).

Já no Brasil, especificamente na cidade de Belo Horizonte (MG), as leis que abrangem o PET consideram basicamente restrições que impõem que a duração máxima de um turno seja de 7:10 horas para as tripulações com turno direto e 6:40 horas para aquelas com turno rachado, acrescidas de até duas horas extras, e restrições que garantam o direito, para todas as tripulações, de 30 minutos para descanso ou alimentação durante seu turno, podendo este período ser fracionado em intervalos menores, desde que um deles seja maior ou igual a 15 minutos. ¹

Por fim, a partir dos turnos gerados, busca-se então construir uma escala com o menor custo operacional possível, isto é, construir um conjunto de turnos que cubram todas as tarefas formadas a partir dos blocos dos veículos e que atendam a todas as regras impostas com o menor custo possível.

Deve-se destacar que a determinação da maneira de se medir o custo, ou seja, a sua “qualidade”, não é um trabalho óbvio, pois cada empresa tem sua política de trabalho, obedece às leis vigentes nas localidades onde operam, e ainda impõe as suas próprias regras. Mas, de uma forma geral, essa qualidade é determinada através do número total de tripulações usadas, do custo sobre as horas pagas, de alguma combinação dessas duas, ou ainda através de alguma medida subjetiva de qualidade (Wren et al., 2003).

¹Essas restrições constam na Convenção Coletiva de Trabalho 2002/2003 celebrada entre o Sindicato das Empresas de Transporte de Passageiros de Belo Horizonte (SETRABH) e o Sindicato dos Trabalhadores em Transporte Rodoviário de Belo Horizonte (STTRBH). Disponível em: <http://www.sttrbh.cjb.net>.

CAPÍTULO 3

REVISÃO BIBLIOGRÁFICA

O problema de escalonamento de tripulações (PET) é um problema muito utilizado e difundido dentro da Pesquisa Operacional e da Otimização Combinatória. Estudado por vários pesquisadores dessas áreas, ele apresenta muitos métodos de solução e inúmeros trabalhos realizados segundo seus preceitos e aplicações.

De uma forma geral, esses trabalhos abordam a resolução do PET em três etapas:

- Geração e atribuição de custos a um conjunto inicial (C) de possíveis turnos candidatos.
- Formulação de um problema de selecionar de C um subconjunto de turnos de custo mínimo que execute todas as tarefas (normalmente através de um problema de cobertura ou particionamento de conjuntos (PCC ou PPC respectivamente)).
- Resolução do PCC/PPC usando um algoritmo apropriado (normalmente baseado em programação linear (PL)).

Beasley e Cao (1996) propuseram um algoritmo baseado em relaxação Lagrangeana, programação linear inteira (PLI), otimização por sub-gradientes e um procedimento de busca em árvore (*branch and bound*) para resolver o problema de escalonamento de tripulações. Eles trataram o PET como um problema de atribuir K tripulações para N tarefas, sem exceder um limite de tempo. Eles representaram o problema como uma formulação de PLI binária, que é relaxada de uma forma Lagrangeana para encontrar um limitante inferior (*lower bound*), que por sua vez é melhorado pela otimização por sub-gradientes. Por fim, o algoritmo de busca em árvore é incorporado para refinar a solução encontrada até então, garantindo a obtenção da otimalidade para o problema. Para validar o método implementado, os autores utilizaram instâncias geradas aleatoriamente (envolvendo de 50 a 500 tarefas), nas quais obtiveram resultados que mostraram que o algoritmo implementado resolveu com sucesso problemas relativamente grandes.

Mingozi et al. (1999) apresentaram uma abordagem para o PET na qual tal problema foi formulado como um PPC, onde cada coluna corresponde a um possível turno. Eles propuseram um procedimento que computa um limitante inferior para o PET através de uma heurística para encontrar uma solução dual relaxada para o PPC. Essa solução é usada para reduzir o número de variáveis (colunas) do PPC de tal forma que esse

problema pudesse ser resolvido através de um algoritmo *branch and bound*. Porém, apesar dos bons resultados obtidos por tal abordagem, as instâncias utilizadas foram geradas de maneira simplificada (apenas com uma restrição de tempo máximo de trabalho) e aleatória, o que pouco condiz com situações reais.

Dias et al. (2001) implementaram um algoritmo genético (AG) para resolver o PET. Eles formularam o problema como um PPC relaxado no qual não se garantiu, em princípio, a cobertura total das tarefas e o atendimento de todas as restrições obrigatórias. O desempenho desse algoritmo foi avaliado através de vários problemas reais (de tamanho médio) de companhias de transporte urbano de Portugal, para os quais os resultados encontrados mostraram que o AG pôde produzir soluções satisfatórias em um tempo razoável.

Lourenço et al. (2001) propuseram a aplicação de uma metaheurística multi-objetiva baseada em Busca Tabu e Algoritmos Genéticos para a solução do PET. Esse trabalho foi parte de um projeto utilizado no planejamento de transportes chamado GIST. O GIST é um sistema de suporte a decisões para auxiliar o departamento de planejamento de companhias de transportes público e privado, ou autoridades de trânsito no gerenciamento de suas operações. O sistema inclui a produção de quadro de horários, escala de veículos, geração de turnos diários para as tripulações, e a construção de turnos de determinadas tripulações por um certo período (semanas ou meses). Inicialmente, o GIST utilizava um algoritmo baseado em programação linear (Beasley, 1987; Vasko e Wolf, 1988) e heurísticas (Beasley, 1987) para obter os limitantes inferiores e superiores para o PET, formulado como um problema de cobertura de conjuntos. Entretanto, as companhias envolvidas no GIST não estavam satisfeitas com o modelo baseado na PL *single-objective*, mesmo sabendo que essa encontrava a solução ótima. Então, como as companhias de transporte tinham diferentes e vários objetivos além da função custo, como por exemplo, a qualidade do serviço, que pode ser medida de diferentes maneiras em cada companhia, o PET foi formulado como um problema de cobertura multi-objetivo, sendo incorporado com sucesso no GIST e resolvendo problemas relativamente grandes, substituindo assim o método de PL que encontrava soluções ótimas, mas era criticado pelos usuários.

Shen e Kwan (2001) desenvolveram, através da aplicação da metaheurística Busca Tabu, um sistema para escalonamento de tripulações chamado HACS. O algoritmo utilizado pelo HACS se mostrou mais rápido em relação aos que utilizavam a PL como base, e capaz de obter resultados comparáveis (um pouco piores) aos melhores resultados conhecidos até então.

Valouxis e Housos (2002) implementaram um algoritmo denominado *Quick Shift (QS)* que foi utilizado juntamente com as técnicas de geração de colunas e de programação linear para solucionar o PET. Basicamente esse algoritmo é constituído de uma combinação dos movimentos *2-opt*, *3-opt* e do algoritmo de caminho mínimo. Nessa abordagem, o *QS* é utilizado para gerar uma população inicial, que por sua vez é melhorada por uma relaxação linear. Posteriormente o *QS* é utilizado para resolver a PLI, e para gerar novas colunas que são então adicionadas a PL. Esse processo é repetido enquanto determinadas regras (critérios de parada) não forem cumpridas. Esse algoritmo, juntamente com as técnicas de geração de colunas e programação linear, apresentou um desempenho satisfatório.

TRACS II (ver Fores et al. (2002); Wren et al. (2003)) é um sistema de escalonamento de tripulações que é formado por um conjunto de módulos. A sua versão anterior denominada IMPACS, foi instalada em uma empresa de Londres em 1984, e a partir de então se tornou a principal ferramenta para escalonar tripulações em tal cidade. O IMPACS e o TRACS II juntos têm sido usados para construir escalas reais em aproximadamente 100 empresas de transporte rodoviário e ferroviário. Recentemente, o TRACS II foi instalado nas vinte e cinco maiores empresas de transporte rodoviário do Reino Unido. Essencialmente, ele trabalha em duas etapas: geração de um grande conjunto de possíveis turnos (possivelmente pouco mais de um milhão) e seleção de um sub-conjunto desses turnos que formará a escala. A geração de possíveis turnos é feita através de um componente chamado BUILD, que durante a geração dos turnos aplica várias heurísticas para filtrar esse conjunto. A seleção dos turnos é feita através da resolução de um problema de cobertura de conjuntos, que é refinado posteriormente por regras heurísticas. Maiores detalhes desse sistema podem ser vistos em Kwan et al. (2004).

Forsyth e Wren (1997) trataram parte do PET através da metaheurística *Ant Colony*, proposta por Dorigo et al. (1996). Eles utilizaram, inicialmente, um componente do sistema TRACS II chamado BUILD (ver Fores et al. (2002)) para gerar um enorme conjunto de possíveis turnos. A partir desses turnos, os autores modelaram o PET segundo um problema de programação linear inteira, representado através de grafos. Para resolução dessa PLI, eles utilizaram tal metaheurística, cujo funcionamento consiste, basicamente, na simulação do comportamento natural das formigas para encontrar alimento. Essa abordagem, apesar de apresentar alguns aspectos interessantes, como a representação em grafos, por exemplo, não apresentou bons resultados.

Várias metodologias, como a descrita no último parágrafo, foram abordadas com o

intuito de melhorar ainda mais a eficiência do sistema TRACS II. Entretanto, nenhuma delas conseguiu tal façanha. Uma boa parte do funcionamento desse sistema pode ser encontrada nos trabalhos referenciados anteriormente, porém alguns pontos cruciais para o desempenho tão “satisfatório” desse sistema não são relatados na literatura.

Além dessas abordagens, muitas outras foram desenvolvidas, sendo que a maioria delas acabaram resultando em sistemas comerciais, os quais são constantemente utilizados na maioria dos países europeus. Além dos sistemas já citados anteriormente, destacam-se também:

- HASTUS (ver Rousseau et al. (1985); Hamer e Séguin (1992)).
- RUCUS (ver Wilhelm (1975)).
- HOT (ver Daduna e Mojsilovic (1988)).
- CARMEN (ver Wedelin (1995)).
- CREW-OPT (ver Desrochers et al. (1992); Desrosiers e Rousseau (1995); Dallaire et al. (2004)).

Apesar da existência de inúmeras abordagens (e sistemas comerciais) sobre o PET, no Brasil tal problema ainda é resolvido de uma maneira precária. Entretanto, alguns pesquisadores vêm desenvolvendo trabalhos interessantes a esse respeito, e o que é mais importante, obtendo bons resultados. Dentre esses trabalhos destacamos três.

O primeiro, proposto por Yunes (2000), trata o PET através de técnicas de programação matemática e programação por restrições, e apresenta ainda um algoritmo híbrido de geração de colunas combinado a essas técnicas. O problema considerado nesse trabalho foi um problema real de escalonamento de motoristas de uma empresa de ônibus urbano da cidade de Belo Horizonte, para o qual o algoritmo híbrido apresentou um desempenho significativamente melhor do que as outras técnicas isoladas.

O segundo consiste em uma abordagem heurística proposta por Souza et al. (2003) que utiliza as metaheurísticas *Simulated Annealing*, *Busca Tabu* e uma hibridização dessas duas técnicas. Nesse trabalho, tais métodos foram testados com dados reais de uma empresa que também opera na cidade de Belo Horizonte.

Já o terceiro, proposto por Mauri (2003), faz uso apenas da metaheurística *Simulated Annealing*. Esse trabalho, além de resolver o PET, propõe também um protótipo de um sistema, chamado *PPT Solver*, cujo enfoque é voltado para a realidade das empresas brasileiras de transporte público, mais especificamente para as empresas que

operam no município de Belo Horizonte. Esse protótipo ainda não está concluído, e conseqüentemente, não disponível para comercialização. Maiores detalhes desse trabalho são mostrados na Seção 3.1.

Um pouco da história do PET, e dos estudos a seu respeito, pode ser visto em Wren (2004) e Ernst et al. (2004), respectivamente.

3.1 PET via *Simulated Annealing*

Mauri (2003) propôs, em seu trabalho, uma aplicação da metaheurística *Simulated Annealing* para solucionar o PET de uma empresa de transporte público urbano. Ele abordou o problema em duas etapas, a de programação diária e a de programação mensal da tripulação.

A programação diária foi tratada, basicamente, da seguinte maneira:

- A partir dos blocos dos veículos são geradas as tarefas (grupos de viagens contidas no bloco dos veículos, separados por oportunidades de troca).
- Feito isso, essas tarefas são distribuídas, aleatoriamente, a um determinado número de tripulações, gerando assim uma solução inicial (C).
- A partir daí, as restrições trabalhistas e operacionais e os custos apresentados em C são penalizados com valores pré-definidos, formando assim a função objetivo de C .
- A seguir, a solução C é utilizada como entrada para o SA (ver Seção 4.2) que executa uma busca por melhores soluções vizinhas (C') no espaço de soluções.

A função objetivo utilizada é obtida através da expressão

$$FO(C) = h(C) + t(C)$$

sendo que

$$h(C) = \sum_{i=1}^m \sum_{j=1}^n \alpha_i B_{ij} \quad e \quad t(C) = \sum_{i=1}^p \sum_{j=1}^n \beta_i B_{ij}$$

onde

C : solução corrente;

$h(C)$: função que avalia o não atendimento dos requisitos essenciais da solução C ;

$t(C)$: função que avalia o não atendimento dos requisitos não essenciais da solução C ;

n : número de tripulações;

m : número de requisitos essenciais;

p : número de requisitos não-essenciais;

α_i : peso que reflete a importância do requisito essencial i ;

β_i : peso que reflete a importância do requisito não-essencial i ;

B_{ij} : quantificação do requisito i na programação da tripulação j .

O autor considerou como requisitos essenciais aquelas restrições que obrigatoriamente devem ser atendidas, e as não essenciais como sendo os custos, ou seja, aquelas que devem ser reduzidas.

As restrições essenciais adotadas foram:

- *Sobreposição de Tarefas*: uma tripulação não pode realizar mais de uma tarefa ao mesmo tempo.
- *Troca de Pontos Proibida*: uma tripulação não pode realizar duas tarefas consecutivas na qual o ponto do final da primeira seja diferente do ponto do início da segunda.
- *Troca de Linhas Proibida*: uma tripulação não pode realizar duas tarefas consecutivas na qual a linha final da primeira seja de um grupo diferente da linha inicial da segunda.
- *Horas Excedentes*: por determinações trabalhistas uma tripulação não pode exercer mais que 9:10 horas de atividades diárias.
- *Folga Acumulada Deficiente*: durante a realização de uma jornada de trabalho a tripulação tem direito a uma folga de 30 minutos destinada a descanso e alimentação. Por acordo feito pelos trabalhadores, esta folga pode ser dividida e distribuída no decorrer do dia, conquanto que uma das frações seja de pelo menos 15 minutos.
- *Tempo entre jornadas*: tempo entre o horário de fim e o horário de início de uma jornada deve haver um período de, no mínimo, 13 horas. Essa condição

garante que entre o horário de término da jornada em um dia e o horário dessa mesma jornada no dia subsequente seja de 11 horas, no mínimo.

E, as restrições não essenciais foram:

- *Duplas Pegadas*: o algoritmo procura reduzir o número de tripulações que têm jornadas do tipo dupla pegada.
- *Horas Extras*: o número de horas que superam 7:10 horas (6:40h + 0:30h) e não ultrapassam 9:10 horas diárias é considerado hora extra. Este montante é minimizado.
- *Troca de Pontos Permitida*: uma troca de pontos é permitida quando uma tripulação realiza duas tarefas consecutivas, cujo ponto de troca do final da primeira é diferente do ponto de início da segunda, e o tempo para ir de um para o outro é menor que o intervalo entre elas. Apesar de permitida é desejável que não haja muitas trocas desse tipo.
- *Troca de Linhas Permitida*: uma troca de linhas é permitida quando uma tripulação realiza duas tarefas consecutivas cuja linha final da primeira pode ser diferente da linha inicial da segunda, desde que a jornada da tripulação seja do tipo dupla pegada.
- *Troca de Veículos*: quando uma tripulação realiza uma tarefa em um veículo e, após o término desta, troca de veículo e realiza uma outra tarefa.
- *Número de Tripulantes*: o número total de tripulações deve ser minimizado até o ponto que o número de horas extras não seja excessivamente grande.
- *Ociosidade*: a soma dos tempos que uma tripulação deixa de trabalhar nas 7:10 horas diárias, descontado o intervalo de 30 minutos para descanso e alimentação.

A estrutura de vizinhança utilizada pela metodologia proposta por Mauri (2003) foi obtida através de um movimento chamado *Realocar Tarefa* (ver Figura 3.1), consistindo na realocação de uma tarefa de uma tripulação i a uma tripulação j .

Essa metodologia apresentou uma diminuição significativa dos custos em relação aos custos reais com os quais Mauri (2003) validou seu trabalho, e ainda mostrou o cumprimento das leis trabalhistas e regras operacionais, validando assim a sua metodologia e a utilização da metaheurística *Simulated Annealing* para resolver o PET.

A Figura 3.1 ilustra a aplicação desse movimento. Em (a) mostram-se as jornadas de duas tripulações, i e j . Em (c) mostra-se que a tarefa t_s do tripulante i foi realocada ao tripulante j .

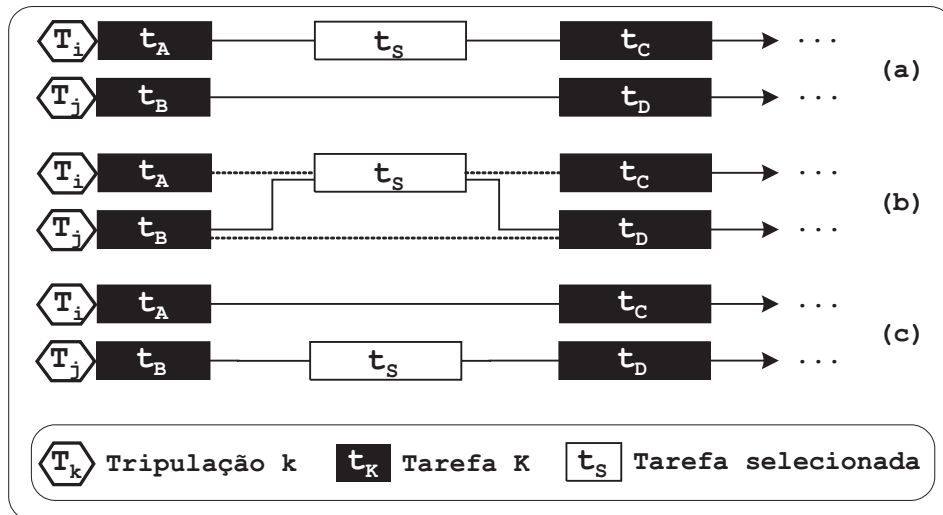


FIGURA 3.1 – Movimento *Realocar Tarefa*.

FONTE: adaptada de Mauri (2003, p. 17).

CAPÍTULO 4

MÉTODOS DE SOLUÇÃO

Problemas de Otimização Combinatória aparecem em áreas tão diversas como posicionamento de satélites, projetos de computadores e de chips VLSI, roteamento ou escalonamento de veículos, alocação de trabalhadores ou máquinas a tarefas, empacotamento de caixas em containers, etc.

Muitos desses problemas podem ser modelados como problemas de minimizar (ou maximizar) uma função cujas variáveis devem obedecer a certas restrições. Encontrar soluções ótimas, ou mesmo aproximadas, para esses tipos de problemas é um desafio nem sempre fácil de ser vencido. Para alguns desses problemas são conhecidos métodos eficientes (rápidos) para se resolvê-los; para outros, principalmente para problemas reais, métodos de enumeração implícita e relaxação são alguns dos aplicados com maior sucesso.

O problema de escalonamento de tripulações (PET), como vários outros problemas, é um problema de “minimização”, ou seja, para a sua solução deve-se minimizar uma função (custo da solução) respeitando-se as restrições impostas. Uma forma de resolver tais problemas seria simplesmente enumerar todas as soluções possíveis e guardar aquela de menor custo. Entretanto, para qualquer problema de um tamanho razoavelmente grande (problema real), este método torna-se impraticável, já que o número de soluções possíveis é muito grande. Portanto, torna-se extremamente necessária a aplicação de métodos mais inteligentes do que a simples enumeração de todas as possíveis soluções.

Neste trabalho, são utilizados vários métodos que já foram adotados para solucionar diversos problemas de otimização. Esses métodos são descritos brevemente nas próximas seções.

4.1 Busca Local

Buscas locais em problemas de otimização são baseadas na noção de vizinhança. Para definirmos o que é uma vizinhança, seja C o espaço de pesquisa de um problema de otimização. O conjunto $N(C) \subseteq C$, o qual depende da estrutura do problema tratado, reúne um número determinado de soluções C' , denominado vizinhança de C . Cada solução $C' \in N(C)$ é chamada de vizinha de C e é obtida de C a partir de uma operação chamada de movimento. Em linhas gerais, uma busca local, começando de uma solução inicial C_0 , navega pelo espaço de pesquisa, através do movimento, passando de uma solução para outra que seja sua vizinha, melhorando o valor de uma função $f(C)$ a ser otimizada enquanto isso for possível.

4.2 Simulated Annealing

O *Simulated Annealing* (SA) é um método de busca local que aceita movimentos de piora para escapar de ótimos locais. Ele foi proposto originalmente por Kirkpatrick et al. (1983), e se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos. Esta técnica começa sua busca a partir de uma solução inicial qualquer. O procedimento principal consiste em um loop que gera aleatoriamente, em cada iteração, um único vizinho C' da solução corrente C .

A cada geração de um vizinho C' de C , é testada a variação do valor da função objetivo (custo), isto é, $\Delta = f(C') - f(C)$. Para um problema de minimização, se $\Delta < 0$, o método aceita a solução e C' passa a ser a nova solução corrente. Caso $\Delta \geq 0$ a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade $e^{-\Delta/T}$, onde T é um parâmetro do método, chamado de *temperatura*, que regula a probabilidade de aceitação de soluções de pior custo.

```
1.  dado  $\alpha$ , SAmax,  $T_0$ ,  $T_c$  e  $s$ ;  
2.   $s^* \leftarrow s$ ;           {Melhor solução obtida até então}  
3.  IterT  $\leftarrow 0$ ;       {Número de iterações na temperatura T}  
4.   $T \leftarrow T_0$ ;        {Temperatura corrente}  
5.  enquanto ( $T > T_c$ )  
6.      enquanto (IterT < SAmax)  
7.          IterT  $\leftarrow$  IterT + 1;  
8.          gerar (um vizinho qualquer  $s' \in N(s)$ );  
9.           $\Delta \leftarrow f(s') - f(s)$ ;  
10.         se ( $\Delta < 0$ )  
11.              $s \leftarrow s'$ ;  
12.             se ( $f(s') < f(s^*)$ )  
13.                  $s^* \leftarrow s'$ ;  
14.             fim-se  
15.         senão  
16.             tomar ( $x \in [0,1]$ );  
17.             se ( $x < e^{-\Delta/T}$ )  
18.                  $s \leftarrow s'$ ;  
19.             fim-se  
20.         fim-se  
21.     fim-enquanto  
22.      $T \leftarrow \alpha * T$ ;  
23.     IterT  $\leftarrow 0$ ;  
24. fim-enquanto  
25.  $s \leftarrow s^*$ ;  
26. retornar  $s$ ;
```

FIGURA 4.1 – Algoritmo *Simulated Annealing*.

FONTE: adaptada de Mauri (2003, p. 12).

A temperatura T assume, inicialmente, um valor elevado T_0 . Após um número fixo de iterações: SAm_{ax} (o qual representa o número de iterações necessárias para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por uma razão de resfriamento α , tal que $T_n \leftarrow \alpha * T_{n-1}$, sendo $0 < \alpha < 1$. Com esse procedimento, no início, dá-se uma chance maior para escapar de mínimos locais e, à medida que T aproxima-se de zero, o algoritmo comporta-se como o método de descida, uma vez que diminui a probabilidade de se aceitar movimentos de piora ($T \rightarrow 0 \Rightarrow e^{-\Delta/T} \rightarrow 0$).

O procedimento pára quando a temperatura chega a um valor próximo de zero (temperatura de congelamento: T_C) e nenhuma solução que piore o valor da melhor solução é mais aceita, isto é, quando o sistema está estável. A solução obtida quando o sistema encontra-se nesta situação evidencia o encontro de um mínimo local, que em alguns casos também pode representar um mínimo global.

Os parâmetros de controle do procedimento são a razão de resfriamento α , o número de iterações para cada temperatura SAm_{ax} , a temperatura inicial T_0 e a temperatura de congelamento T_C . A Figura 4.1 apresenta o algoritmo *Simulated Annealing*.

4.3 Programação Linear

Programação linear (PL) pode ser descrita como um processo de transformar um problema da vida real em um modelo matemático (Sierksma, 1996). O modelo matemático consiste de uma função objetivo, que deve ser minimizada ou maximizada, e de um conjunto finito de restrições, sendo que tanto a função objetivo quanto as restrições são lineares em um número finito de variáveis de decisão. Caso essas variáveis de decisão sejam estritamente inteiras, a programação linear é chamada de programação linear inteira (PLI), ou simplesmente programação inteira (Sierksma, 1996).

O método *Simplex*, que é a principal forma de se resolver um modelo de PL, foi desenvolvido em 1947 por G. B. Dantzig, e apresentado em 1963 em um de seus mais famosos livros, *Linear Programming and Extensions*.

A PL está entre as técnicas de maior sucesso e mais amplamente aplicadas em problemas de otimização, e é utilizada, com sucesso, em praticamente todos os tipos de problemas combinatoriais, como por exemplo, problemas de geração de escalas, geração de quadros de horários, roteamento de veículos, etc. Para maiores detalhes dessa técnica, ver Sierksma (1996) e Ignizio e Cavalier (1994).

4.4 Geração de Colunas

A Geração de Colunas é uma técnica baseada no princípio de decomposição de Dantzig-Wolfe, que é uma técnica de programação linear generalizada. Nessa técnica, um problema mestre (PM) permite a seleção de um melhor subconjunto de colunas (i.e. variáveis de decisão), e é resolvido por uma formulação de PL do problema ou normalmente sua versão relaxada. Esse problema, por sua vez, tem seu próprio modelo, baseado freqüentemente em um modelo de cobertura ou particionamento de conjuntos com restrições adicionais, como por exemplo, restrições básicas de cobertura (i.e. cada tarefa deve ser coberta por uma coluna), e restrições globais, como o número de recursos disponíveis (tripulações, caminhões,...) (Jacquet-Lagrèze e Lebbar, 2000).

Conhecida há várias décadas pela sua capacidade de resolver problemas grandes de PL, a idéia principal dessa técnica consiste em: considerar um pequeno número de variáveis (colunas) de cada vez; resolver o problema de PL (problema mestre formado por essas colunas) e obter as soluções primal e dual; gerar novas colunas interessantes, com custo reduzido negativo (sub-problema), através da solução dual do problema principal para melhorar a solução da PL anterior; e repetir esse processo até que nenhuma melhora possa ser obtida na solução da PL (ver Jacquet-Lagrèze e Lebbar (2000)).

Entretanto, o termo “geração de colunas” tem sido usado em três contextos diferentes, todos aplicados a problemas com um grande número de colunas. Ao invés de enumerar tantas colunas explicitamente, estes métodos as tratam implicitamente, através de um conjunto seletivo. Cada um desses métodos envolve um problema mestre (PM), que deve ser otimizado. Esse problema é restringido, pois as suas colunas não são todas explicitamente conhecidas. Conseqüentemente, ele é chamado de problema mestre restrito (PMR) (Wilhelm, 2001).

O primeiro contexto de geração de colunas usa um modelo auxiliar (MA) para identificar um conjunto de colunas “atrativo”, definindo um PMR que é otimizado a partir dessas colunas definidas explicitamente. O PMR aceita essas colunas e não interage mais com o MA. Nesse contexto, um MA é utilizado para gerar um grande número de possíveis colunas e um PMR para prescrever o melhor subconjunto destas.

Já o segundo, usa um sub-problema (SP), o qual interage com o PMR para identificar as melhores colunas. Ao invés de definir as colunas explicitamente, esse método usa um SP para identificar as colunas não básicas com o melhor critério de entrada do *Simplex*. O PMR provê os valores das variáveis duais para atualizar os coeficientes da função objetivo no SP, permitindo gerar melhores colunas quando necessário.

O terceiro contexto, aplica a Decomposição de Dantzig-Wolfe (ver Dantzig e Wolfe (1960); Bazaraa et al. (1990)) para o relaxamento linear de uma PLI. A cada iteração, os valores das variáveis duais do PMR atualizam os coeficientes da função objetivo em cada SP, e um ou mais SP's são resolvidos para gerar melhores colunas, se possível.

Consideremos um problema clássico de PL:

Minimizar

$$F = \sum_j c_j x_j$$

Sujeito à

$$\sum_j a_{ij} x_j \geq b_i \quad \forall i \quad e \quad x_j \geq 0 \quad \forall j$$

Consideremos também os valores duais das restrições para o ótimo: λ_i . Assumindo que alguma variável (coluna) x_j ainda não esteja inserida no problema, para analisar se essa nova variável poderá melhorar a sua função objetivo, devemos calcular seu custo reduzido usando o elemento correspondente a_{ij} associado a ela:

$$\theta_j = c_j - \sum_i a_{ij} \lambda_i$$

Se $\theta_j < 0$ é interessante que a coluna seja introduzida na PL para melhorar a sua função objetivo (este é o primeiro critério de Dantzig do algoritmo Simplex).

Dessa forma, um método eficiente para gerar novas colunas, com custo reduzido negativo, é gerar essas colunas dinamicamente, levando em consideração os valores das variáveis duais da solução corrente.

4.5 Algoritmos Genéticos

Desde os anos 80, os Algoritmos Genéticos (AG's) são utilizados para solucionar problemas de otimização combinatória e de outras áreas. AG é um procedimento computacional capaz de emular o processo Darwiniano de seleção natural, que vem se mostrando ser bastante interessante em aplicações da otimização combinatória (Holland, 1975; Goldberg, 1989; Colin, 1995).

Em analogia com o sistema biológico, cada candidato à solução do problema representa um cromossomo ou indivíduo da população, que é o conjunto de todos esses candidatos. Cada indivíduo da população pode ser representado no AG por um conjunto binário 0 e 1, por números inteiros, por números de ponto flutuante ou por caracteres; e tem um

custo associado que determina sua habilidade para sobreviver e produzir descendentes no processo de seleção natural. Para melhorar o custo ou a aptidão dos indivíduos das populações sucessivas são necessários os operadores genéticos. Esses operadores permitem selecionar indivíduos mais aptos à sobrevivência para reprodução, além de manter as características de adaptação adquiridas pelas gerações passadas, criando indivíduos cada vez mais aptos à sobrevivência.

4.6 Algoritmo Genético Construtivo

O Algoritmo Genético Construtivo (AGC) foi, primeiramente, abordado por Lorena e Lopes (1996). O aspecto construtivo do AGC está relacionado ao fato de, ao longo do processo seletivo, uma população de soluções incompletas (esquemas) é utilizada como base para a construção de uma população de soluções completas (estruturas). Tanto esquemas quanto estruturas competem entre si durante o processo evolutivo. Estruturas são formadas a partir de recombinações de esquemas e, então, tem-se uma melhora tanto das estruturas quanto dos esquemas ao longo das gerações. Outras características importantes e inovadoras do AGC estão na utilização de populações de tamanho variável ao longo das gerações e a possibilidade de usar heurísticas para definição das duas funções de avaliação de aptidão dos indivíduos. O AGC também foi aplicado com sucesso em problemas de otimização combinatória em Ribeiro Filho e Lorena (2000a), Ribeiro Filho e Lorena (2000b), Ribeiro Filho e Lorena (2001), Lorena e Furtado (2001), Arakaki e Lorena (2001) e Lorena et al. (2002).

4.7 Algoritmo de Treinamento Populacional

Trata-se de uma técnica evolutiva empregada pioneiramente por Oliveira (2002), derivada do Algoritmo Genético Construtivo (AGC), e desenvolvida no Instituto Nacional de Pesquisas Espaciais (INPE). Segundo Oliveira (2002), o termo treinamento significa fazer com que certo modelo seja adaptado a um ambiente inicialmente desconhecido através do ajuste de seus parâmetros livres de acordo com alguma medida de desempenho esperada.

Na proposta de tese de Oliveira (2002) é apresentado o Algoritmo de Treinamento Populacional (ATP), baseando-se nas idéias do AGC, e principalmente, no uso de heurísticas para definição das aptidões dos indivíduos na população. O ATP pode ser visto como uma nova forma de direcionar a evolução de indivíduos (soluções candidatas) utilizando-se para isso, informações sobre o ambiente (problema) a ser assimilado. Tais informações são extraídas do ambiente através de heurísticas específicas. Um indivíduo será considerado bem adaptado de acordo com certa heurística se ele não puder evoluir (ser melhorado) com relação à heurística de treinamento empregada. Um indivíduo bem

adaptado deve, então, participar ao máximo de gerações possíveis do processo evolutivo. A adaptação no treinamento populacional é, portanto, utilizada como direcionador, ou seja, regiões do espaço de busca (universo de possíveis combinações de soluções) ocupadas por indivíduos bem adaptados serão mais exploradas que as demais. A Figura 4.2 apresenta o ATP.

```

1.  inicializar (P,alpha,t);
2.  enquanto (o critério de parada não for satisfeito)
3.      t ← t+1;
4.      selecionar (P,base,percentual base);
5.      selecionar (P,guia,toda a população);
6.      novo ← recombinar (base,guia);
7.      avaliar (novo);
8.      se (rand() < probabilidade de mutação)
9.          mutação (novo,tamanho da vizinhança);
10.     fim-se
11.     avaliar (vizinhança(novo));
12.     calcular (delta(novo));
13.     se (delta(novo) > alpha)
14.         atualizar (população(P,novo));
15.     fim-se
16.     alpha ← alpha + incrementar (Step);
17.     enquanto (existir delta(S) < alpha)
18.         eliminar (S);
19.     fim-enquanto
20. fim-enquanto

```

FIGURA 4.2 – Algoritmo de Treinamento Populacional (ATP).
 FONTE: adaptada de Oliveira e Lorena (2003).

O ATP começa a partir de uma população P gerada aleatoriamente ou através de um outro método de inicialização. A partir dessa população, são criados novos indivíduos a cada geração (t), até que um critério de parada seja satisfeito (normalmente um número máximo de gerações). Os novos indivíduos são criados a partir de um cruzamento entre dois indivíduos já existentes na população. Esses dois indivíduos, chamados de “base” e “guia”, são selecionados a partir de uma parte da população que contém os indivíduos mais adaptados e a partir de toda a população, respectivamente. Os novos indivíduos podem sofrer ainda, com certa probabilidade, “mutações”, diversificando assim as características herdadas de seus “ancestrais”.

O novo indivíduo é então avaliado através de duas funções, g e f , utilizadas para o processo de treinamento evolutivo. A função $g(S_k)$ mostra o custo que determina a habilidade de sobrevivência do indivíduo S_k . Para a definição da função $f(S_k)$, são avaliados vários indivíduos alternativos em uma vizinhança de S_k , tomando-se o melhor (de menor custo),

isto é, $f(S_k) = \text{Melhor } g(S_k') \mid S_k' \in \text{Vizinhança}(S_k)$. O processo evolutivo é desenvolvido privilegiando os indivíduos de menor diferença $[g(S_k) - f(S_k)]$, atribuindo os seguintes valores de *rank* aos indivíduos:

$$\delta(S_k) = d \times [g_{\max} - g(S_k)] - [g(S_k) - f(S_k)] \quad (4.1)$$

onde g_{\max} é um limitante superior de custo alto, e d um percentual constante de g_{\max} , isto é, uma constante de proporcionalidade para o intervalo $[g_{\max} - g(S_k)]$.

A população é controlada dinamicamente por um parâmetro de evolução, denominado α , que é definido por:

$$\alpha_i = \alpha_{i-1} + \text{Step} \times PS \times \frac{\delta_{bst} - \delta_{wst}}{RG} \quad (4.2)$$

onde *Step* é uma constante que controla a velocidade da evolução, *PS* é o tamanho da população corrente, $(\delta_{bst} - \delta_{wst})$ é a variação, no momento, entre os valores dos *ranks* do melhor e do pior indivíduo, respectivamente, e *RG* é o número de gerações que faltam para terminar o processo. Dessa forma, a partir dos valores dos *ranks* (ver Equação 4.1), é determinada a permanência ou não dos novos indivíduos na população, ou seja, são aceitos apenas indivíduos com $\delta(S_k) > \alpha$, enquanto os demais são descartados. Essa constante também é atualizada a cada geração, eliminando-se no fim de cada uma destas, os indivíduos menos adaptados ($\delta(S_k) < \alpha$).

CAPÍTULO 5

TRATAMENTO DO PROBLEMA

Neste trabalho, são tratadas duas abordagens como meta para o PET. A primeira, baseada na maioria dos trabalhos referentes ao problema, diz respeito ao custo total da solução (escala) encontrada, ou seja, minimizar o custo total do conjunto de tripulações que formarão a solução. Já a segunda, baseada nos trabalhos desenvolvidos por Beasley e Cao (1996) e Mingozzi et al. (1999), tem o mesmo objetivo, porém ela leva em consideração um número fixo de tripulações.

Devido à necessidade encontrada em problemas reais de se determinar qual a melhor maneira de avaliar uma escala (Wren et al., 2003), essas duas abordagens são implementadas com o intuito de apresentar soluções variadas para cada instância utilizada. Para cada instância procura-se, na primeira abordagem, encontrar o número de tripulações necessário para que a escala contenha o menor custo possível. Na segunda abordagem tem-se o mesmo objetivo, porém considera-se números de tripulações fixos, próximos ao obtido na primeira abordagem. Trata-se assim, características encontradas em problemas reais, onde são utilizadas escalas com o custo alto, porém com um número reduzido de tripulações, e escalas com o menor custo possível obtidas sem se importar com esse número.

A representação do PET abordada neste trabalho toma como base as entidades propostas por Mauri (2003) devido à sua identificação com outras encontradas na literatura, à sua clareza e ao resultado obtido em seu trabalho para problemas reais. Sendo assim, o problema é modelado basicamente através das entidades: *Tarefas*, *Tripulações* e *Turnos* (ver Capítulo 2).

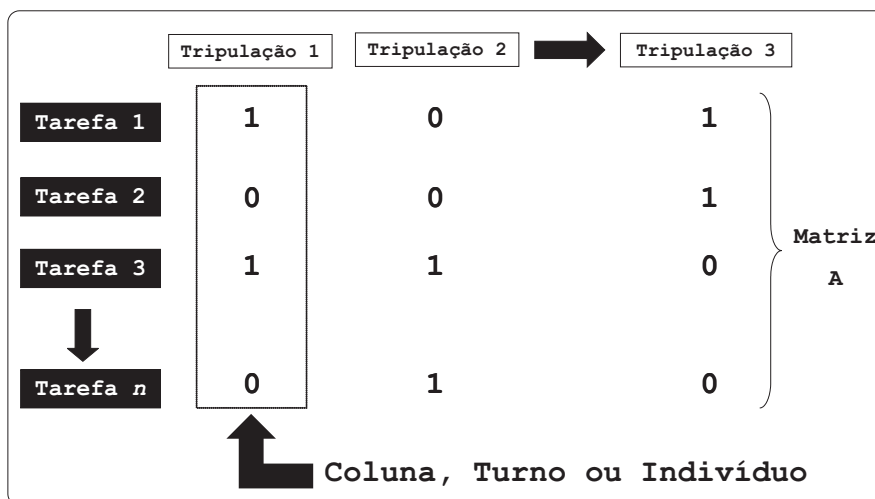


FIGURA 5.1 – Representação do problema.

O problema é descrito como o de formação de uma matriz, onde as colunas representam as tripulações, e as linhas as tarefas (ver Figura 5.1). Cada elemento $a_{ij} \in \{0, 1\}$, sendo $i \in M = \{1 \dots m\}$ e $j \in N = \{1 \dots n\}$, m o número de tarefas (linhas), e n o número de tripulações (colunas), onde $a_{ij} = 1$ se a tarefa i pertence ao turno da tripulação j , e 0 caso contrário.

Assim, a representação de um turno é dada por: $S_k = (a_{1k}, a_{2k}, \dots, a_{mk})$. Por exemplo, $S_k = (1, 0, 1, 1, \dots, 0)$ significa que as tarefas 1, 3, 4, ..., fazem parte do turno da tripulação k . A partir desse modelo as tripulações são avaliadas de acordo com a equação:

$$c_j = (TE_j \times w_e) + (TS_j \times w_e) + (HE_j \times w_{ne}) + (TO_j \times w_{ne}) \quad (5.1)$$

onde

$$TE_j = \max(0, [HT(t_{ult}) - HI(t_{pri})] - TMT)$$

$$TS_j = \sum_{i=1}^{p-1} \max(0, [HT(t_i) - HI(t_{i+1})])$$

$$HE_j = \max(0, [HT(t_{ult}) - HI(t_{pri})] - TNT)$$

$$TO_j = \max(0, TNT - [HT(t_{ult}) - HI(t_{pri})]) + \sum_{i=1}^{p-1} \max(0, [HI(t_{i+1}) - HT(t_i)])$$

c_j é o custo da tripulação j ; TE_j é o excesso no tempo máximo de trabalho da tripulação j (tempo excessivo); TS_j é o tempo de sobreposição da tripulação j (tempo em que a tripulação está, teoricamente, realizando duas tarefas ao mesmo tempo); HE_j é o tempo que excede o tempo normal de trabalho da tripulação j (horas extras); TO_j é o tempo ocioso da tripulação j (tempo em que a tripulação não está trabalhando durante seu turno); w_e é uma “penalização” para o não atendimento das restrições essenciais; w_{ne} é uma “penalização” para as restrições não-essenciais; TMT é o tempo máximo de trabalho (8h + 2h¹); TNT é o tempo normal de trabalho (8h); p é o número de tarefas do turno da tripulação j ; $HT(t_i)$ é o horário de término da tarefa i ; $HI(t_i)$ é o horário de início da tarefa i ; t_{pri} é a primeira tarefa do turno da tripulação j ; t_{ult} é a última tarefa do turno da tripulação j . Essas restrições são apresentadas na Figura 2.3, página 25.

Essas restrições são consideradas, como na maioria das companhias do setor de transportes, onde é essencial (obrigatório) evitar o tempo excessivo e o tempo de sobreposição, mas o tempo ocioso e as horas extras são toleráveis (não-essenciais). Não

¹Esses valores foram aproximados dos valores utilizados pelas empresas brasileiras e por valores citados na literatura. Os valores não foram iguais aos reais devido ao fato de que neste trabalho não são consideradas algumas restrições (como por exemplo intervalo para descanso e alimentação).

são tratadas todas aquelas restrições descritas na Seção 3.1 devido ao fato de que o PET é abordado de uma forma mais geral (baseada nos trabalhos de Beasley e Cao (1996); Mingozzi et al. (1999)), porém com algumas características de problemas reais (representadas na Equação 5.1).

5.1 ATP/PL

Neste trabalho, o PET é tratado através de uma metodologia ainda não explorada na literatura, baseada na geração de colunas. Essa metodologia, denominada **ATP/PL**, é baseada na aplicação do Algoritmo de Treinamento Populacional (ATP) (ver Seção 4.7) juntamente com a Programação Linear (PL) (ver Seção 4.3) através da técnica de Geração de Colunas (ver Seção 4.4). O ATP e a PL são aplicados de maneira interativa, sendo o ATP, através de informações da relaxação da PL, responsável pela geração de boas colunas (baixo custo e boa cobertura das tarefas), e a PL pela resolução de um problema de particionamento de conjuntos (PPC) formado por essas colunas. O PPC é então formulado da seguinte maneira:

Minimizar:

$$\sum_{j=1}^n c_j x_j \quad (5.2)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j = 1 \quad i = 1, \dots, m \quad (5.3)$$

$$x_j \in \{0, 1\}; \quad j = 1, \dots, n \quad (5.4)$$

onde c_j representa o custo da coluna j (conforme Equação 5.1) e x_j é igual a 1 se a coluna (turno) j pertencer à solução do problema e 0 caso contrário. Esta é uma formulação clássica e constantemente utilizada em diversos trabalhos encontrados na literatura (Desrochers e Soumis, 1989; Wilhelm, 2001).

Já para a segunda abordagem, o PPC é formulado da mesma maneira, porém com a inclusão de uma nova restrição:

$$\sum_{j=1}^n x_j = Z \quad (5.5)$$

onde n representa o número total de colunas (tripulações) do problema e Z um número

pré-determinado. Essa abordagem foi baseada nos trabalhos desenvolvidos por Beasley e Cao (1996) e Mingozzi et al. (1999).

Inicialmente, para a utilização do ATP na geração de colunas deve-se definir os operadores genéticos (cruzamento e mutação), a geração da população inicial e a função de treinamento (função f) utilizada por tal método, sendo que todos esses processos devam sempre resultar em colunas válidas, isto é, colunas (turnos) que atendam à todas as restrições essenciais. Nota-se que, dessa forma, todas as colunas inseridas no PPC representam turnos factíveis, possibilitando a geração de soluções válidas para o problema.

Como função de treinamento, é utilizada uma heurística de busca local simples, que é responsável por avaliar vários indivíduos (colunas) alternativos em uma vizinhança. Essa heurística é descrita na Figura 5.2.

```
1.  seja  $S_k$  um turno qualquer;  
2.   $f(S_k) \leftarrow g(S_k)$ ;  
3.   $S'_k \leftarrow \text{clone}(S_k)$ ;  
4.  por (tamanho_da_vizinhança vezes)  
5.     $i \leftarrow \text{selecionar}$  (uma tarefa de  $S'_k$ );  
6.     $j \leftarrow \text{selecionar}$  (uma tarefa qualquer);  
7.    remover (a tarefa  $i$  de  $S'_k$ );  
8.    adicionar (a tarefa  $j$  a  $S'_k$ );  
9.    calcular ( $g(S'_k)$ );  
10.   se ( $g(S'_k) < f(S_k)$ )  
11.      $f(S_k) \leftarrow g(S'_k)$ ;  
12.   fim-se  
13. fim-por
```

FIGURA 5.2 – Heurística de treinamento.

A mutação utilizada também é baseada em uma busca local, implementada através da troca de uma tarefa (selecionada aleatoriamente) do turno de uma determinada tripulação (indivíduo) por uma outra qualquer, também selecionada aleatoriamente. Esse processo é repetido até que se tenha um turno válido (ver Figura 5.3).

O cruzamento é responsável por gerar novos indivíduos (sempre válidos) da seguinte maneira: dois indivíduos são selecionados (base e guia) e seus cromossomos são fundidos. Como a população é ordenada de maneira decrescente através dos valores dos *ranks* (ver Equação 4.1) dos indivíduos, um indivíduo base é selecionado da primeira parte da população, enquanto o guia é selecionado de toda ela. Feito isso, para garantir que o indivíduo filho seja válido, uma posição do cromossomo fundido é selecionada (aleatoriamente), as tarefas posicionadas antes dessa posição são removidas, e as

tarefas subseqüentes são inseridas de uma forma seqüencial enquanto o novo indivíduo representar um turno válido. Esse processo é descrito na Figura 5.4.

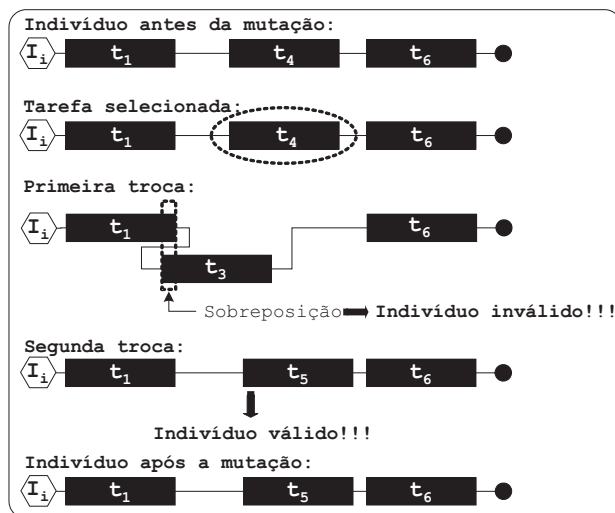


FIGURA 5.3 – Mutação.

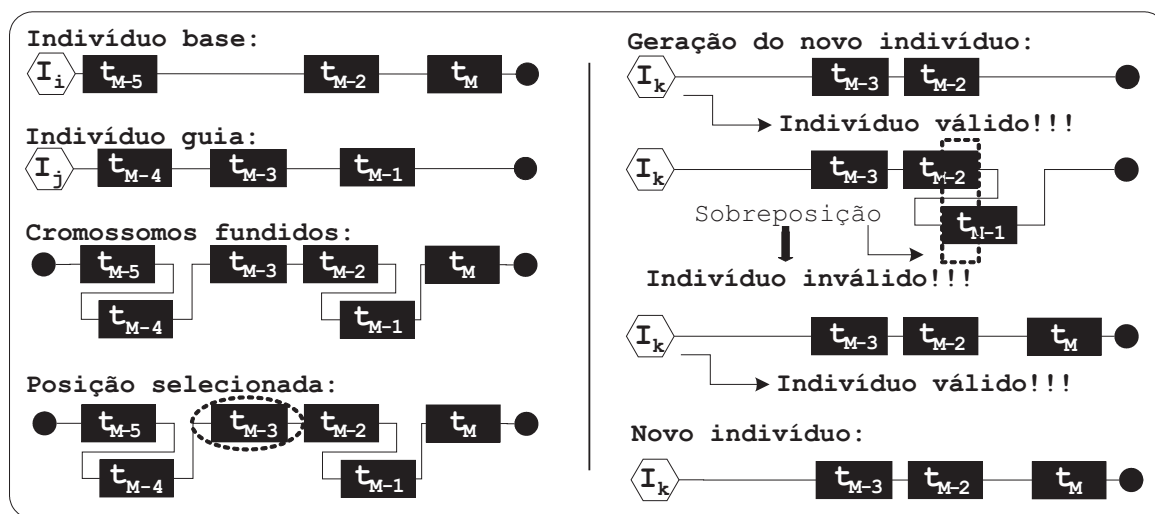


FIGURA 5.4 – Cruzamento.

A população inicial utilizada pelo ATP, na ATP/PL, também é implementada de forma simples, sendo gerados aleatoriamente q (tamanho da população inicial) indivíduos, podendo estes serem válidos ou não. A geração de indivíduos (colunas) inválidos proporciona um certo grau de diversidade na população, e durante a evolução desta, tais indivíduos são eliminados naturalmente.

É interessante notar que, através desses processos, as melhores colunas deverão incluir um número de tarefas diversificado, caracterizado pela heurística de treinamento (função

de treinamento) que guiará o processo evolutivo.

A interatividade do ATP com a PL é baseada principalmente através da definição da função de adaptação (função g) dos indivíduos no ATP. Tal função é definida com o uso de informações das variáveis duais retiradas da PL. Para a primeira abordagem do PET, a função g é definida por:

$$g(S_k) = \frac{c(S_k)}{\sum_{i=1}^m \lambda_i a_{ik}} \quad \text{para} \left(\sum_{i=1}^m \lambda_i a_{ik} \right) > 0 \quad (5.6)$$

e

$$g(S_k) = c(S_k) \quad \text{para} \left(\sum_{i=1}^m \lambda_i a_{ik} \right) \leq 0 \quad (5.7)$$

Para a segunda abordagem do PET, usando a restrição definida pela Equação 5.5, essa função é definida por:

$$g(S_k) = \frac{c(S_k)}{\sum_{i=1}^m \lambda_i a_{ik} + \lambda_{m+1}} \quad \text{para} \left(\sum_{i=1}^m \lambda_i a_{ik} + \lambda_{m+1} \right) > 0 \quad (5.8)$$

e

$$g(S_k) = c(S_k) \quad \text{para} \left(\sum_{i=1}^m \lambda_i a_{ik} + \lambda_{m+1} \right) \leq 0 \quad (5.9)$$

onde $c(S_k)$ é o custo do turno da tripulação (coluna) k (conforme Equação 5.1); λ_i é a variável dual para a tarefa i e λ_{m+1} é a variável dual para a restrição definida pela Equação 5.5.

Utilizando os conceitos da técnica de geração de colunas (ver Seção 4.4), podemos calcular o custo reduzido da k -ésima coluna inseridas no PPC, denotado por θ_k , através das equações:

$$\theta_k = c(S_k) - \sum_{i=1}^m \lambda_i a_{ik} \quad (5.10)$$

e

$$\theta_k = c(S_k) - \left(\sum_{i=1}^m \lambda_i a_{ik} + \lambda_{m+1} \right) \quad (5.11)$$

sendo que a Equação 5.10 refere-se à primeira abordagem do PET, e a Equação 5.11 à segunda.

A partir das Equações 5.6 e 5.10 e Equações 5.8 e 5.11, pode-se notar que para custos reduzidos negativos ($\theta_k < 0$), o valor da função g estará dentro do intervalo $[0,1)$, fazendo com que a heurística de treinamento, que define o valor da função f correspondente (melhor g em uma vizinhança), proporcione uma pequena diferença ($g - f$) para colunas que tenham custo reduzido negativo. Para custos reduzido positivos ($\theta_k \geq 0$) o valor da função g será igual ao custo, ou seja, um valor “elevado”. Assim, a população é treinada indiretamente para indivíduos com custo reduzido negativo (melhorando a cobertura das tarefas para o PPC) evitando a geração de um número excessivo de colunas, e conseqüentemente acelerando o processo.

Definida a implementação do ATP, tal metodologia é incorporada à ATP/PL. Nesta, uma população inicial é gerada aleatoriamente, porém de forma direcionada ao problema, ou seja, contendo apenas colunas válidas (exceto a coluna com todas as tarefas) e que cubram todas as tarefas (ver Figura 5.5).

```

1.  para (i = 1 até número_de_tarefas)
2.      criar (uma coluna com a tarefa i);
3.      adicionar (essa coluna na população);
4.  fim-para
5.  criar (uma coluna com todas as tarefas);
6.  adicionar (essa coluna na população);
7.  por (número_inicial_de_colunas vezes)
8.      criar (uma coluna sem tarefas);
9.      i ← uma tarefa qualquer;
10.     adicionar (i à coluna criada);
11.     enquanto (i <= número_de_tarefas)
12.         j ← uma tarefa qualquer posterior a i;
13.         adicionar (j à coluna);
14.         se (coluna for inválida)
15.             remover (j);
16.         fim-se
17.         i ← j;
18.     fim-enquanto
19.     adicionar (a coluna criada na população);
20. fim-por

```

FIGURA 5.5 – População inicial direcionada ao PPC.

No início da Figura 5.5 (linhas 1 à 4) pode-se notar que todas as tarefas serão cobertas, e conseqüentemente, para a primeira abordagem do PET, a ATP/PL sempre encontrará uma solução final (para a PLI) válida, pois tal solução não depende do número de colunas que a formará. Já para a segunda abordagem, como o número de colunas (tripulações) que formará a solução é fixo, tal regra não será válida, pois inicialmente não é sabido se existirá uma solução com o número de tripulações desejado. Entretanto, as linhas 5 e 6, auxiliam na obtenção da primeira solução para a PL (não inteira), permitindo assim o início da ATP/PL.

É interessante notar que na prática essa coluna com todas as tarefas (criada nas linhas 5 e 6) terá um custo muito alto, pois ela sempre será uma coluna inválida, e conseqüentemente não irá pertencer à solução final do problema (solução da PLI).

A partir dessa população é formado um PPC, que é resolvido através da PL. Feito isso, novas colunas são geradas através do ATP, considerando os valores das variáveis duais, para construir as funções de adaptação. As colunas com custo reduzido negativo são adicionadas ao PPC corrente, e este é resolvido novamente através da PL. Esse processo é repetido por um determinado número de iterações ou enquanto o valor da PL não estabilizar.

Finalmente, terminada essa interação, a PL é convertida em PLI (através do *software* CPLEX), a qual apresenta uma solução viável para o PPC. A resolução do PPC de uma forma exata ainda poderá ser árdua, principalmente para problemas de porte mais elevado. Assim, após algum *tempo de processamento* ou um *gap limite* (porcentagem da diferença dos valores da melhor PLI e PL) a melhor solução encontrada é tomada como solução para o PET. A Figura 5.6 apresenta o pseudo-código da ATP/PL.

```
1. criar (população direcionada ao problema);
2. resolver (PL);
3. por (número_máximo_de_iterações ou PL não estabilizar)
4.     executar (ATP);
5.     remover (colunas inválidas);
6.     calcular (custo reduzido das novas colunas);
7.     adicionar (colunas com custo reduzido negativo);
8.     resolver (PL);
9. fim-por;
10. converter (PL para PLI);
11. resolver (PLI);
```

FIGURA 5.6 – Algoritmo ATP/PL.

5.2 *Simulated Annealing*

Para comparação com a ATP/PL, foram implementadas duas versões da metaheurística *Simulated Annealing* (ver Seção 4.2). Tais implementações foram baseadas no trabalho de Mauri (2003), o qual apresentou bons resultados para problemas reais.

A primeira implementação, chamada de **SA**, é baseada em uma única execução da *Simulated Annealing*. Já a segunda, **SA_20**, consiste na execução de tal metodologia por 20 vezes, para diferentes soluções iniciais, sendo que a melhor solução obtida em todas as execuções é dada como solução final.

Para a primeira abordagem do PET (obter uma solução independentemente do número de tripulações) o número de tripulações na solução inicial foi igual à metade do número de tarefas, enquanto que para a segunda abordagem (obter uma solução com um número “fixo” de tripulações) o número de tripulações na solução inicial foi igual ao “número de tripulações” desejado.

Nessas implementações (SA e SA_20), a função objetivo a ser minimizada considera apenas as restrições descritas na Equação 5.1, página 44, ou seja, da mesma forma que a utilizada na ATP/PL. Tal função é representada pela seguinte equação:

$$\sum_{j=1}^n c_j \tag{5.12}$$

sendo n igual ao número de tripulações da solução, e c_j igual ao custo da tripulação j (definido pela Equação 5.1).

Como critério de parada foi utilizado a “estabilidade”, ou seja, quando a temperatura chega a um valor próximo de zero (temperatura de congelamento). Em alguns casos, foi fixado um tempo máximo de processamento, sendo que nesses casos, se o sistema estabilizasse antes de tal tempo, ele era reaquecido (à temperatura inicial).

Maiores detalhes da metodologia *Simulated Annealing* implementada podem ser vistos na Seção 4.2 e no trabalho de Mauri (2003).

5.3 Instâncias Utilizadas

Para validação da ATP/PL, foram utilizadas instâncias geradas, aleatoriamente, a partir de um problema real enfrentado por uma empresa brasileira de transporte público. Essa geração foi feita a partir da seleção aleatória de tarefas reais retiradas de tal problema, formando cinco instâncias (contendo 25, 50, 100, 250 e 500 tarefas). Cada uma dessas

instâncias é formada pelos horários de início e de término de cada tarefa, e são ordenadas de acordo com o seu horário de início. A Figura 5.7 apresenta a formatação de tais instâncias.

Foram utilizadas também algumas instâncias apresentadas por Beasley e Cao (1996). Entretanto, a modelagem do problema utilizada por tais autores foi diferente da utilizada neste trabalho. Eles apresentaram uma modelagem baseada em um problema de fluxo com capacidade, onde os custos considerados foram fixos (custos de transição entre as tarefas) e gerados de forma aleatória. Assim, foram encontradas algumas dificuldades para aplicar a ATP/PL às instâncias apresentadas por eles. A maior dificuldade foi na forma de penalizar um turno perante o ATP, pois da maneira como tais autores modelaram o problema, o ATP tendia a gerar colunas com apenas uma tarefa, ou seja, colunas sem “transição” (colunas de custo zero).

Algumas maneiras de contornar tal situação foram implementadas “sem sucesso”, como por exemplo, penalizar turnos com poucas tarefas e penalizar turnos cujo fluxo (tempo de trabalho) fosse muito diferente da “capacidade” (valor dado nas instâncias). Essas estratégias não foram suficientes, pois como os custos de transição foram gerados aleatoriamente, não foi possível determinar uma relação de “peso” entre tais custos e a capacidade (tempo de trabalho). Outros operadores genéticos e funções de treinamento também foram implementados, mas também não foram capazes de contornar as diferenças das modelagens.

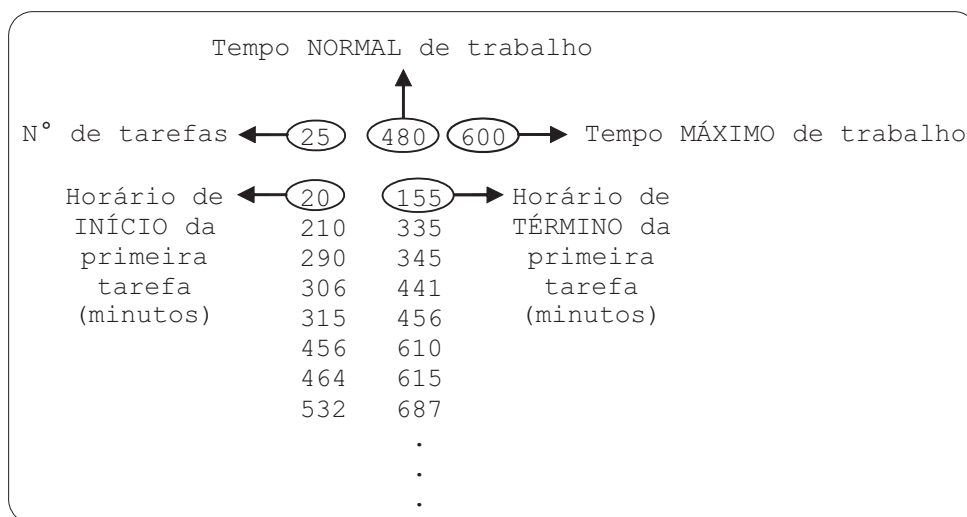


FIGURA 5.7 – Formatação das instâncias utilizadas.

CAPÍTULO 6

ANÁLISE DA ATP/PL

Antes de se desenvolver a metodologia *ATP/PL*, descrita na Seção 5.1, foram implementadas outras metodologias visando tratar o PET. Na prática, algumas se mostraram incapazes para tal fim, enquanto outras, à medida em que foram sendo refinadas, apresentaram resultados interessantes e satisfatórios.

Tais metodologias foram avaliadas considerando-se apenas a instância com 500 tarefas e a primeira abordagem para o PET, cujo objetivo é obter uma solução com o menor custo possível independentemente do número de tripulações.

A primeira metodologia adotada (***ATP+PL***) foi baseada em uma aplicação do ATP para gerar um grande número de colunas com baixo custo e boa cobertura das tarefas, formando assim um PPC, que por sua vez é resolvido pela PLI através do *software* de otimização CPLEX. O custo de uma coluna (turno) foi calculado a partir da Equação 5.1, página 44, porém sem o uso de penalizações diferenciadas para as restrições essenciais e não-essenciais, ou seja:

$$c_j = TE_j + TS_j + HE_j + TO_j \quad (6.1)$$

A função de avaliação dos indivíduos (função g) utilizada foi igual ao próprio custo do turno (Equação 6.1).

O ATP não obteve um bom desempenho para a geração de colunas, pois a solução encontrada pela PLI para o PPC formado por essas colunas, apesar de ter sido obtida utilizando um número relativamente pequeno de colunas, não foi satisfatória, pois apresentou custos altos e um grande número de tripulações necessário para cobrir todas as tarefas. Além disso, a execução dessa metodologia foi muito lenta.

Visto que essa metodologia não foi suficiente para resolver o PET de uma forma satisfatória, uma outra maneira de tratar tal problema foi adotada (*ATP/PL₁*). Essa metodologia foi semelhante à *ATP/PL* (ver Seção 5.1), ou seja, de uma forma interativa, porém, o custo das colunas foi calculado a partir da Equação 6.1 e a função de avaliação dos indivíduos (função g) foi igual a esse custo. Além disso, no fim de cada interação, as colunas com maior custo reduzido (5% das colunas) eram eliminadas do problema.

O desempenho também não foi satisfatório. Entretanto, foi interessante notar que, dessa maneira, o ATP gerava indivíduos considerando apenas a sua própria qualidade (custo), e não tratava em momento algum a qualidade dos indivíduos perante o

PPC. Além disso, a partir de um certo momento, o número de colunas removidas do PPC passou a ser maior do que o número de colunas inseridas, fazendo com que tal problema perdesse informações importantes, uma vez que o número de colunas tendia a diminuir bruscamente, impossibilitando a obtenção de uma solução para o problema. Esse comportamento é ilustrado nas Figuras 6.1 e 6.2.

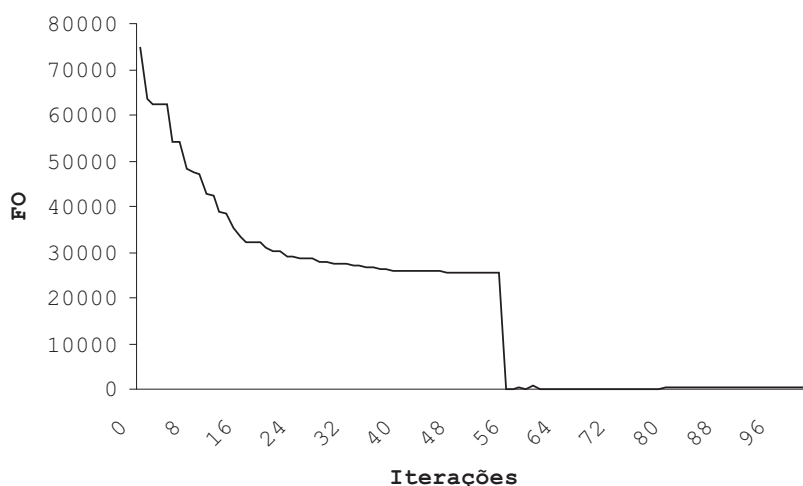


FIGURA 6.1 – Função objetivo da ATP/PL₁ durante a interação ATP e PL.



FIGURA 6.2 – Número de colunas da ATP/PL₁ durante a interação ATP e PL.

Tentando eliminar essas “falhas”, uma nova forma de tratar o PET foi implementada (*ATP/PL₂*). Nessa, o problema foi tratado de forma semelhante a *ATP/PL₁*, porém a função de avaliação dos indivíduos (função *g*) utilizada foi igual à descrita na Seção 5.1 (ver Equações 5.6 e 5.7), fazendo assim com que o ATP levasse em consideração não apenas o custo das colunas, mais sim o custo e a sua influência perante o PPC, ou seja, o ATP gerava as colunas de uma forma direcionada ao PPC. Além disso, todas as colunas

inseridas no PPC eram mantidas até o final da interação (ATP e PL), possibilitando a obtenção de uma solução para o problema.

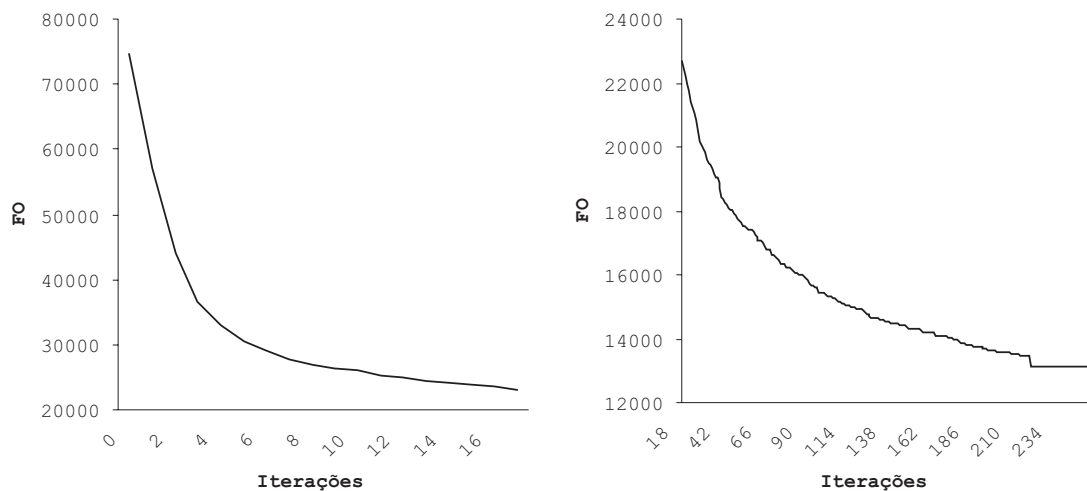


FIGURA 6.3 – Função objetivo da ATP/PL₂ durante a interação ATP e PL.

O desempenho melhorou, mas notou-se que, a partir de um certo momento o ATP não conseguia gerar colunas com custo reduzido negativo, fazendo a solução da PL estabilizar-se precocemente, impedindo uma possível melhora na solução final. As Figuras 6.3 e 6.4) ilustram esse comportamento.

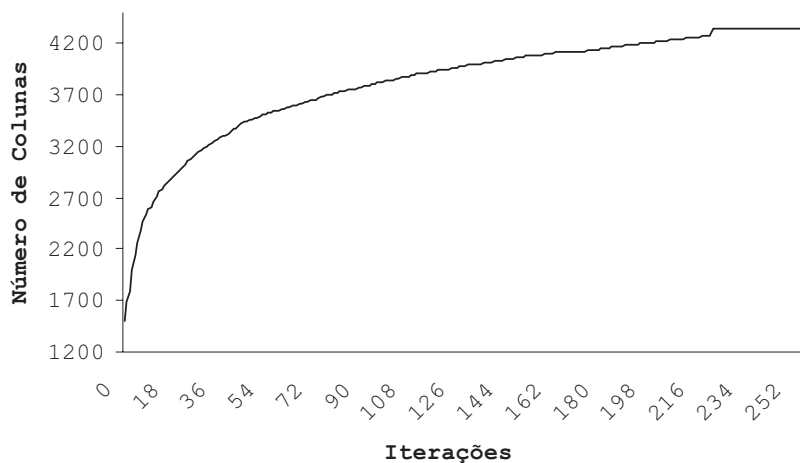


FIGURA 6.4 – Número de colunas da ATP/PL₂ durante a interação ATP e PL.

Notou-se que quando a solução da PL estabilizava, se todas as colunas geradas pelo ATP, independentemente do seu custo reduzido, fossem inseridas no PPC, a solução da PL continuava a convergir. É interessante notar que utilizando essa estratégia, essas colunas inserem informações novas ao PPC, diversificando assim o conjunto de colunas já contidas

nesse problema, e conseqüentemente expandindo as possibilidades para solucioná-lo. Assim se chegou na versão *ATP/PL_3*, que foi apresentada em Mauri e Lorena (2004a) e Mauri e Lorena (2004b).

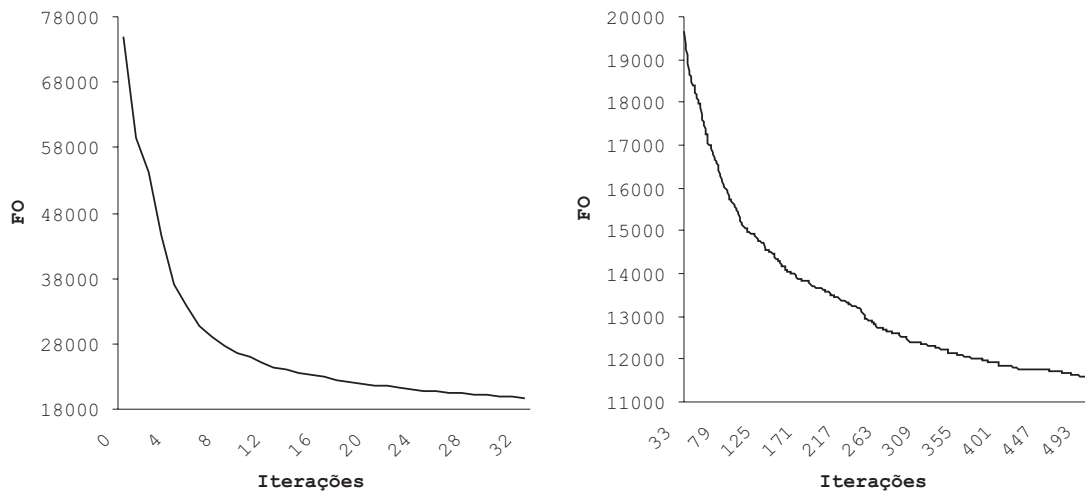


FIGURA 6.5 – Função objetivo da *ATP/PL_3* durante a interação ATP e PL.

O desempenho dessa versão foi ainda melhor, pois foi obtida uma solução satisfatória (em relação à SA e SA₂₀) em um tempo de processamento relativamente pequeno. Porém, o número de colunas geradas ainda foi muito grande, tornando a resolução do PPC através da PLI muito lenta, e portanto, sendo interrompida após certo tempo de processamento (6000 s). O comportamento da *ATP/PL_3* é ilustrado nas Figuras 6.5 e 6.6.

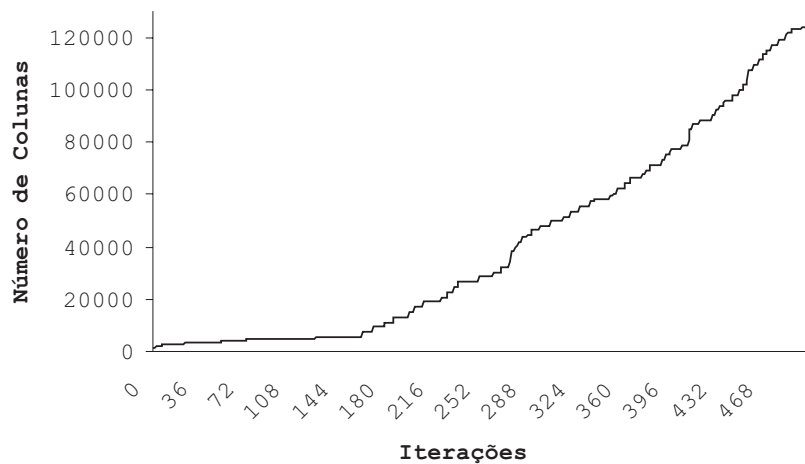


FIGURA 6.6 – Número de colunas da *ATP/PL_3* durante a interação ATP e PL.

A seguir foram realizados vários experimentos visando encontrar um conjunto de parâmetros que pudesse melhorar o desempenho da *ATP/PL_3*. O que se percebeu foi

uma melhora significativa no desempenho da ATP/PL₃ quando executada por mais iterações, porém com menos gerações do ATP por iteração, ou seja, aumentando a interação do ATP com a PL. Por uma questão de notação, a ATP/PL₃ “calibrada” é chamada de ATP/PL₄.

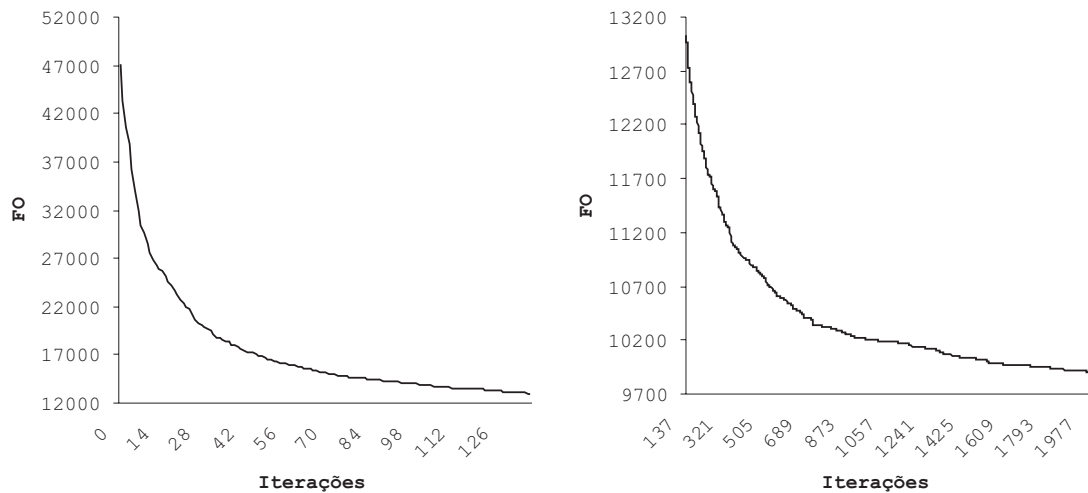


FIGURA 6.7 – Função objetivo da ATP/PL₄ durante a interação ATP e PL.

A ATP/PL₄ apresentou uma solução de menor custo em relação à apresentada pela ATP/PL₃, e o mais interessante, com um número de colunas significativamente menor (menos de 50%). Percebeu-se que na ATP/PL₄ a solução da PL vai melhorando aos poucos, pois o ATP gera poucas colunas por iteração, controlando assim o número de colunas geradas, e conseqüentemente reduzindo o tempo para resolver o PPC através da PL em cada iteração. Isso pode ser observado nas Figuras 6.7 e 6.8.



FIGURA 6.8 – Número de colunas da ATP/PL₄ durante a interação ATP e PL.

Com a metodologia bem calibrada (ATP/PL₄), notou-se que a inserção de colunas com custo reduzido positivo (no momento em que a solução da PL estabilizava) estava fazendo com que o número de colunas crescesse significativamente, e conseqüentemente retardando o tempo de processamento e até mesmo dificultando a obtenção de bons resultados. Dessa forma, diferentemente da ATP/PL₄, apenas as colunas com custo reduzido negativo foram adicionadas ao PPC, chegando-se assim na ATP/PL₅.



FIGURA 6.9 – Função objetivo da ATP/PL₅ durante a interação ATP e PL.

Notou-se também que mesmo estabilizando por algumas iterações, a ATP/PL₅ continuava convergindo, e conseguia obter uma solução expressivamente melhor do que a metodologia anterior e com um número de colunas ainda menor (ver Figuras 6.9 e 6.10).

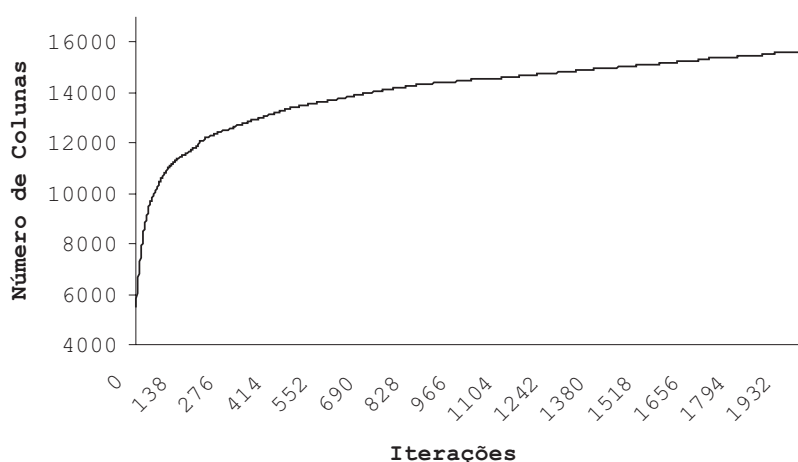


FIGURA 6.10 – Número de colunas da ATP/PL₅ durante a interação ATP e PL.

Como a ATP/PL₅ apresentou melhores soluções em relação às demais metodologias, e com um menor tempo de processamento, ela foi executada por um número maior de

iterações, sendo denominada, após tal alteração, *ATP/PL_6*, a qual melhorou ainda mais os resultados.

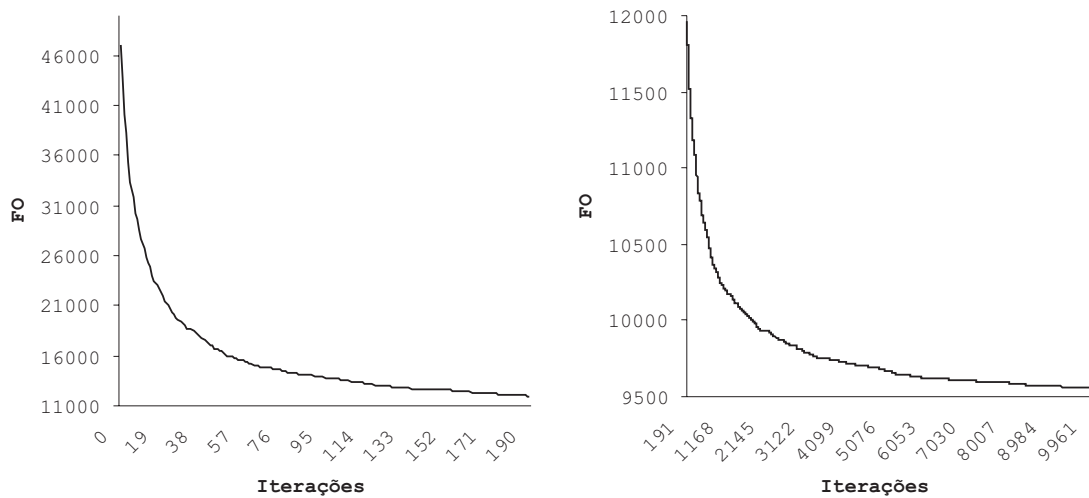


FIGURA 6.11 – Função objetivo da *ATP/PL_6* durante a interação ATP e PL.

É interessante notar que ao contrário da *ATP/PL_3* (onde tanto o número de iterações (entre o ATP e a PL) quanto o número de gerações no ATP foram “elevadas”), a *ATP/PL_6* trabalha com uma maior interatividade entre suas técnicas (ATP e PL). Observa-se também que dessa forma, o ATP gera poucas colunas por iteração, porém tais colunas conseguem melhorar a solução da PL (ver Figuras 6.11 e 6.12). Deve-se destacar também que um número muito pequeno de gerações no ATP pode fazer com que a *ATP/PL_6* estabilize por várias iterações.

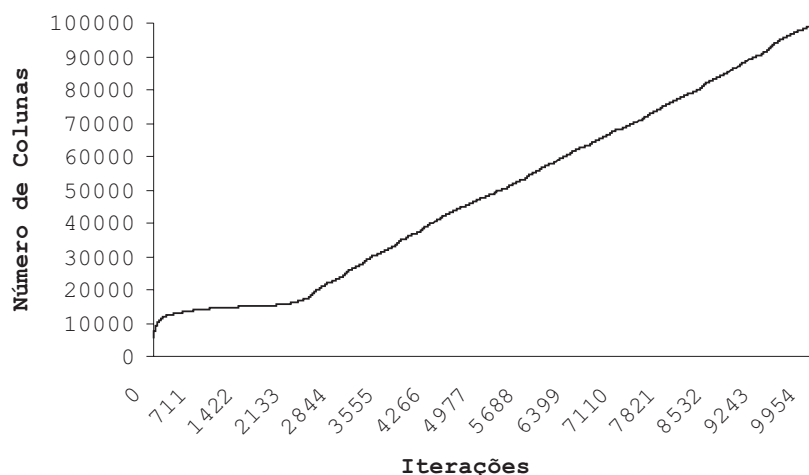


FIGURA 6.12 – Número de colunas da *ATP/PL_6* durante a interação ATP e PL.

Finalmente, analisando a forma de avaliar os turnos, utilizada no SA (como em Mauri (2003)), percebeu-se que a maneira usada nas APT/PL's prejudicava a geração das

colunas no ATP.

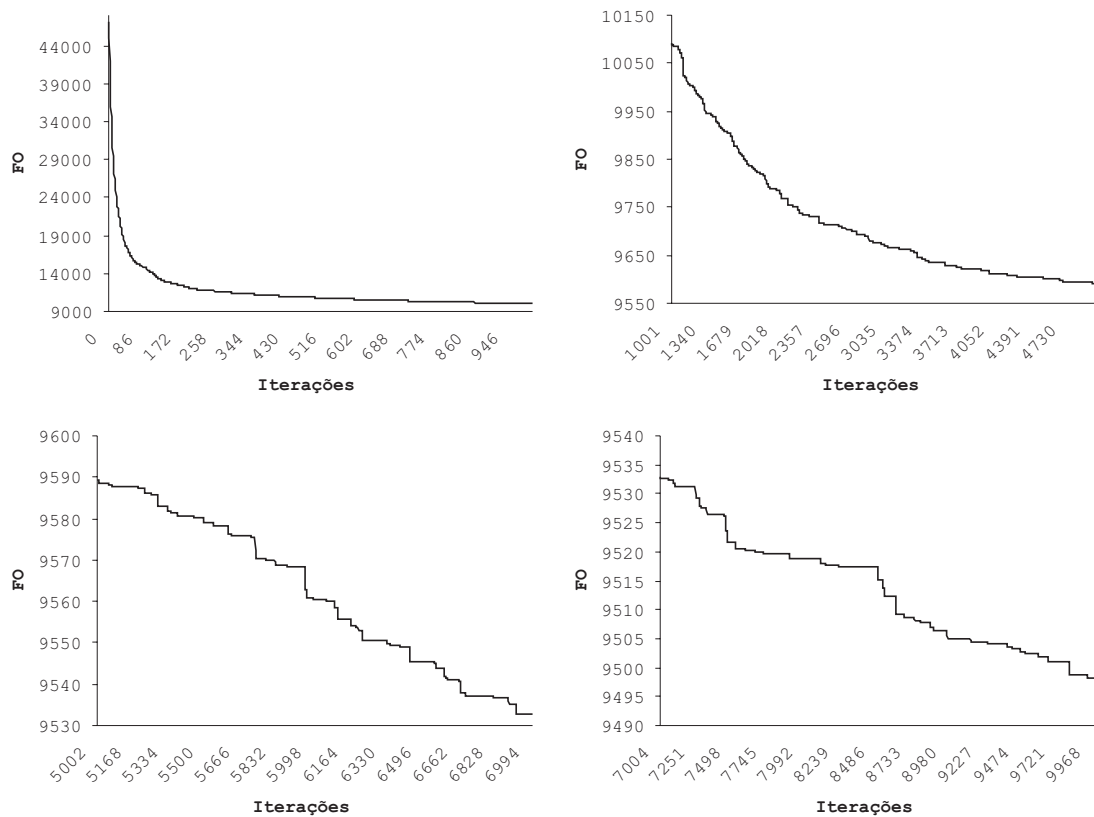


FIGURA 6.13 – Função objetivo da ATP/PL durante a interação ATP e PL.

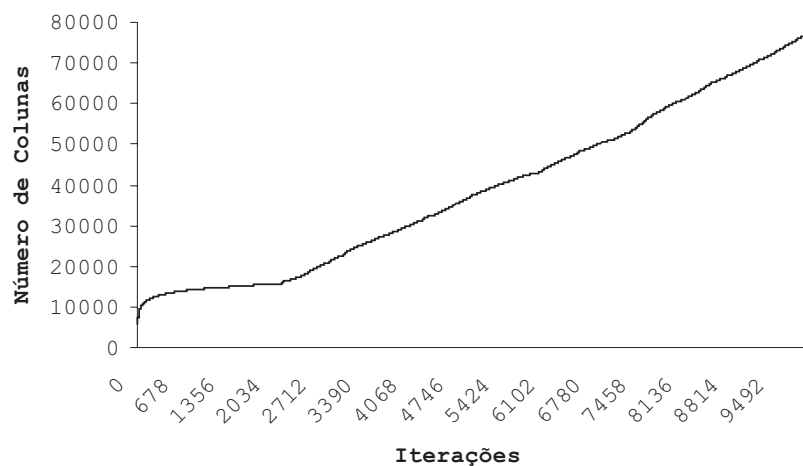


FIGURA 6.14 – Número de colunas da ATP/PL durante a interação ATP e PL.

Dessa forma, todas as metodologias apresentadas, consideravam um minuto de uma restrição essencial com mesmo “peso” de um não-essencial, conseqüentemente para o ATP uma coluna com um minuto de sobreposição tinha o mesmo custo de uma com um minuto de ociosidade. Esse fato prejudicou o desempenho de tais metodologias, pois elas

geravam muitas colunas inválidas (que não atendem as restrições essenciais), as quais não eram adicionadas ao PPC. Percebendo tal fato, uma nova forma de penalizar as colunas (turnos) foi adotada (ver Equação 5.1), chegando-se assim na metodologia *ATP/PL*, apresentada na Seção 5.1.

Como pode ser visto nas Figuras 6.13 e 6.14, na *ATP/PL* o ATP consegue gerar colunas com custo reduzido negativo durante toda a interação com a PL, e o valor da função objetivo da PL não estabiliza, resultando em uma excelente solução para o problema.

Visando reduzir o tempo gasto na solução do PPC através da PLI, tentou-se reduzir o número de colunas geradas pela *ATP/PL* através da “eliminação de colunas”. Essa eliminação foi feita da seguinte forma: Após resolvido o primeiro PPC (via PL), foi calculada a média dos custos reduzidos das colunas inseridas em tal problema. Feito isso, a cada iteração (do ATP com a PL) as colunas com o custo reduzido maior do que essa média foram removidas do PPC. Por uma questão de notação, esta “nova” metodologia será denotada por *ATP/PL-7*.

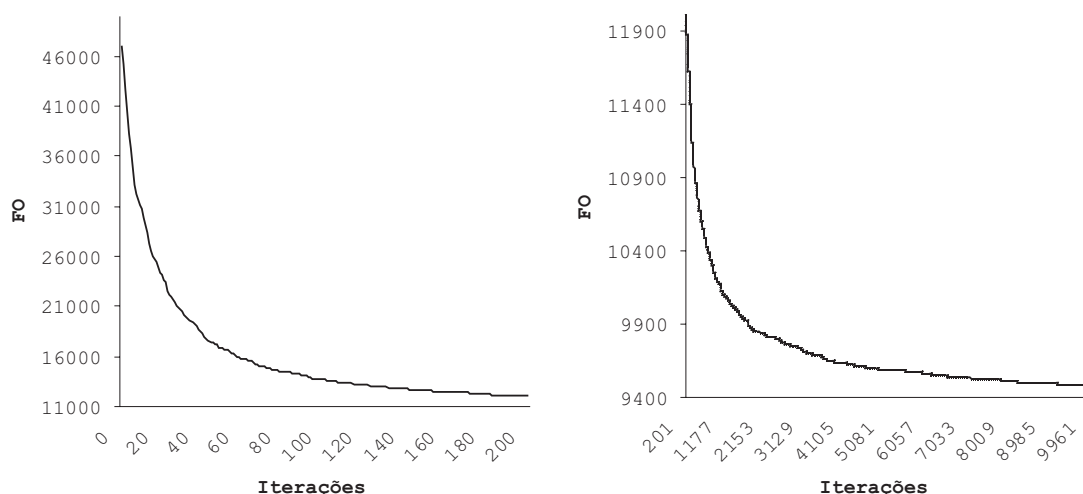


FIGURA 6.15 – Função objetivo da *ATP/PL-7* durante a interação ATP e PL.

A *ATP/PL-7* não foi eficiente, pois a redução no número de colunas foi muito pequena (em relação à *ATP/PL*). Isso aconteceu devido ao fato de que a partir de certa iteração, nenhuma coluna era removida do PPC, ou seja, a média obtida após a resolução do primeiro PPC estava maior do que todas as colunas pertencentes ao problema.

Além disso, as poucas colunas eliminadas fizeram com que houvesse perda de “informações” no PPC, fazendo com que a solução obtida fosse pior do que a obtida pela *ATP/PL*. Esse comportamento é mostrado nas Figuras 6.15 e 6.16.

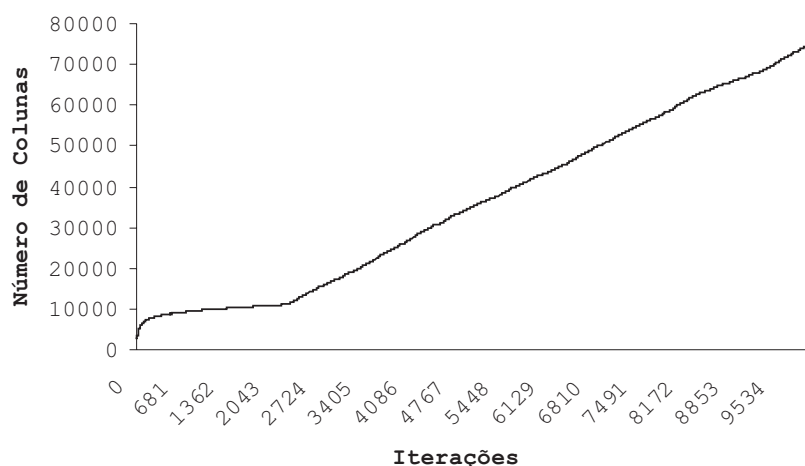


FIGURA 6.16 – Número de colunas da ATP/PL_7 durante a interação ATP e PL.

Ainda tentando reduzir o número de colunas gerados na ATP/PL, um novo recurso foi aplicado: a exclusão de colunas, porém agora com o cálculo da média a cada iteração (*ATP/PL_8*). Essa exclusão foi feita da seguinte forma: Em cada iteração, após resolvido o PPC (via PL), calcula-se a média dos custos reduzidos das colunas inseridas em tal problema. Feito isso, as colunas com o custo reduzido maior do que essa média eram removidas do PPC.

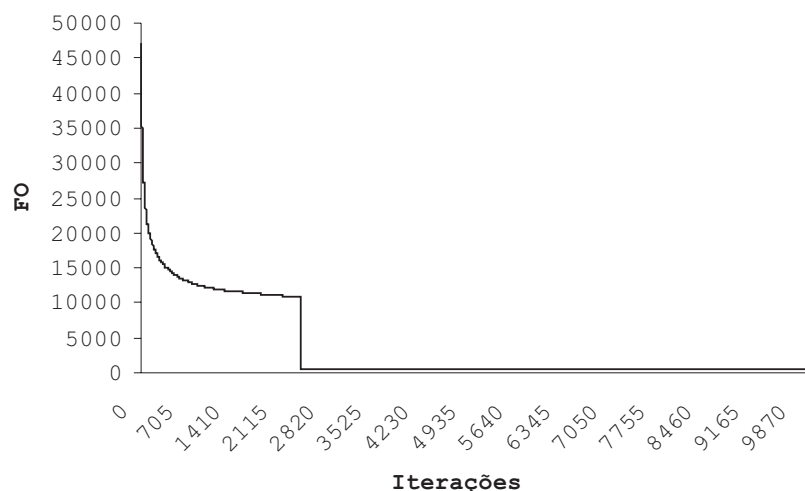


FIGURA 6.17 – Função objetivo da ATP/PL_8 durante a interação ATP e PL.

Essa metodologia também não foi eficiente, pois a redução no número de colunas foi muito grande, fazendo com que nenhuma solução fosse obtida. Essas colunas excluídas do PPC fizeram com que houvesse grandes perdas de “informações” no PPC, impossibilitando a obtenção de uma solução inteira. Isso pode ser visto nas Figuras 6.17 e 6.18.

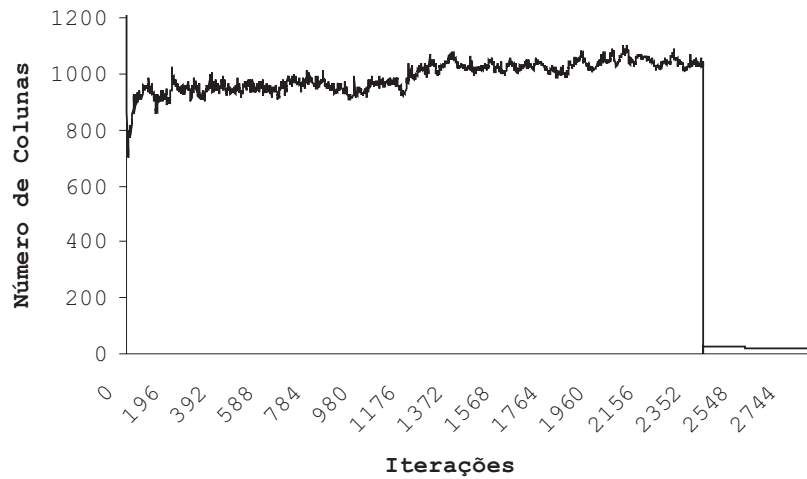


FIGURA 6.18 – Número de colunas da ATP/PL_8 durante a interação ATP e PL.

Na Figura 6.19 é feito um comparativo do número final de colunas geradas (número de colunas do PPC resolvido através da PLI) pelas metodologias apresentadas neste capítulo.

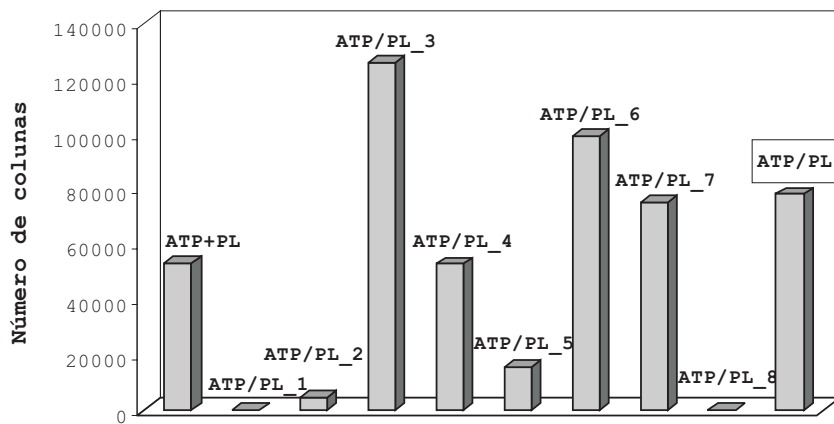


FIGURA 6.19 – Número de colunas geradas pelas metodologias ATP/PL's.

Na Figura 6.20, é feito um comparativo do custo da solução obtida (custo da solução obtida pela PLI) pelas metodologias apresentadas neste capítulo. Maiores detalhes dessas soluções são mostrados na Tabela 6.1.¹

¹ Os parâmetros utilizados nas ATP/PL's descritas neste capítulo, e algumas informações a respeito das suas execuções, são mostrados no Apêndice A.

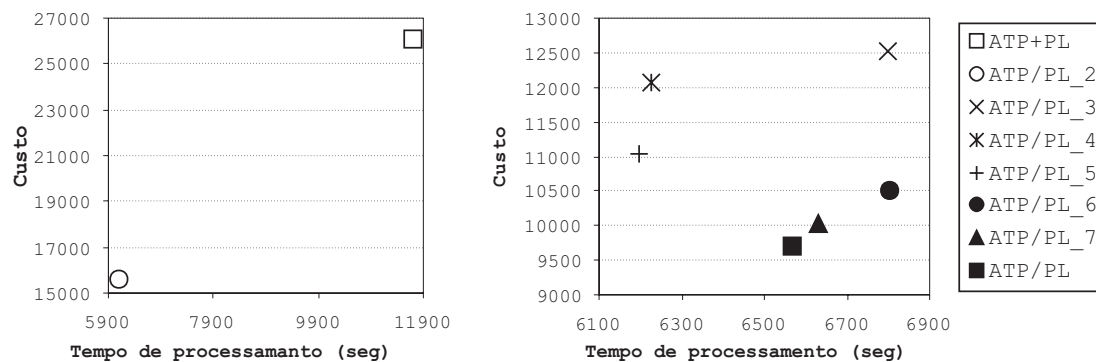


FIGURA 6.20 – Custo das soluções obtidas.

TABELA 6.1 – Soluções obtidas pelas ATP/PL's.

Método de solução	Número de tripulações	Horas extras (min)	Tempo ocioso (min)	Custo da solução	Número de colunas	Tempo (s)
ATP+PL	178	1870	24224	26094	53187	11702,34
ATP/PL_1	-	-	-	-	-	-
ATP/PL_2	156	1898	13692	15590	4346	6107,97
ATP/PL_3	151	1567	10961	12528	125991	6797,40
ATP/PL_4	151	1333	10727	12060	52851	6225,08
ATP/PL_5	149	1304	9738	11042	15656	6196,63
ATP/PL_6	147	1514	8988	10502	99364	6801,90
ATP/PL	145	1601	8115	9716	78082	6567,80
ATP/PL_7	146	1525	8519	10044	74930	6630,99
ATP/PL_8	-	-	-	-	-	-

CAPÍTULO 7

RESULTADOS COMPUTACIONAIS

A ATP/PL, apresentada na Seção 5.1, a SA e a SA_20 descritas na Seção 5.2, foram aplicadas às instâncias descritas na Seção 5.3. Os resultados obtidos, para a primeira abordagem do PET (obter uma solução com o menor custo possível independentemente do número de tripulações) são mostrados na Tabela 7.1. Nessa tabela, a coluna “Número final de colunas” refere-se ao número de colunas contidas no PPC resolvido através da PLI. Além disso, em todas as tabelas deste capítulo, os valores da coluna “Tempo” já incluem o tempo de processamento da PLI, ou seja, indicam o tempo total de processamento da ATP/PL, da SA e da SA_20.

TABELA 7.1 – Resultados obtidos para a primeira abordagem do PET.

Número de tarefas	Método de solução	Número de tripulações	Horas extras (min)	Tempo ocioso (min)	Custo total da solução	Número final de colunas	Tempo (s)
25	ATP/PL	12	15	2356	2371	351	0,10
	SA	12	15	2356	2371	-	1,90
	SA_20	12	15	2356	2371	-	35,48
50	ATP/PL	20	0	2600	2600	1113	4,63
	SA	20	27	2627	2654	-	42,87
	SA_20	20	0	2600	2600	-	949,04
100	ATP/PL	40	0	7395	7395	2375	1,88
	SA	40	0	7395	7395	-	7,60
	SA_20	40	0	7395	7395	-	173,26
250	ATP/PL	81	1103	7751	8854	29834	70,90
	SA	85	815	9383	10198	-	199,86
	SA_20	83	1112	8720	9832	-	3749,02
500	ATP/PL	145	1601	8115	9716	78082	6567,80
	SA	153	1254	11608	12862	-	7061,52
	SA_20	153	1096	11450	12546	-	143565,20

Nota-se pela Tabela 7.1, que para as instâncias com 25, 50 e 100 tarefas a ATP/PL obteve soluções com custo semelhante aos obtidos pela SA_20. Já a SA conseguiu obter soluções idênticas às obtidas pela ATP/PL e pela SA_20 apenas para as instâncias com 25 e 100 tarefas, e para a instância com 50 tarefas tal metodologia não apresentou uma solução com a mesma “qualidade”. Além disso, em todos esses casos, o tempo de processamento da ATP/PL foi significativamente menor do que ambas as aplicações baseadas na *Simulated Annealing*. Para as instâncias com 250 e 500 tarefas, ou seja, para instâncias “grandes”, a ATP/PL obteve soluções expressivamente melhores em relação à SA, tanto no custo da solução quanto no tempo de processamento. Nota-se também que a *Simulated Annealing*, mesmo executada por 20 vezes (SA_20), obteve

piores soluções e com um tempo de processamento superior à ATP/PL. Para a instância com 500 tarefas, o número de colunas do PPC final foi relativamente grande, tornando árdua a sua resolução através da PLI. Assim, a execução desta foi interrompida após 6000 s, e a solução encontrada até então foi adotada como solução para o problema.

Ainda tentando medir o desempenho da ATP/PL, foram realizados mais quatro experimentos para cada instância. Esses experimentos foram realizados a partir da utilização de “sementes aleatórias”, isto é, diferentes sementes utilizadas em cada experimento. A partir desses experimentos e dos experimentos apresentados na Tabela 7.1 (sem o uso de sementes aleatórias) foi calculado o *desvio* obtido nas soluções apresentadas. Esse desvio foi calculado através da seguinte expressão:

$$Desvio = \frac{(Med_Custos - Men_Custo)}{Men_Custo} \times 100 \quad (7.1)$$

onde *Med_Custos* é a média aritmética do custo das soluções obtidas nos experimentos e *Men_Custo* é o menor custo conhecido para os problemas. Além do desvio, também foram calculados os tempos médios de processamento (média aritmética dos tempos de processamento dos experimentos) para cada instância. O desvio e o tempo médio de processamento para cada instância utilizada neste trabalho são apresentados nas Tabelas 7.2, 7.3, 7.4, 7.5 e 7.6.

TABELA 7.2 – Desvio obtido com 25 tarefas.

25 Tarefas					
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Custo	2371	2371	2371	2371	2371
Tempo	0,10	0,10	0,10	0,10	0,11
Desvio:	0,000%		Tempo médio:		0,10 s

TABELA 7.3 – Desvio obtido com 50 tarefas.

50 Tarefas					
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Custo	2600	2600	2600	2602	2600
Tempo	4,63	4,84	4,60	4,74	4,83
Desvio:	0,015%		Tempo médio:		4,73 s

TABELA 7.4 – Desvio obtido com 100 tarefas.

100 Tarefas					
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Custo	7395	7395	7395	7395	7395
Tempo	1,88	1,48	1,53	1,49	1,47
Desvio:	0,000%		Tempo médio:		1,57 s

TABELA 7.5 – Desvio obtido com 250 tarefas.

250 Tarefas					
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Custo	8854	8854	8854	8860	8854
Tempo	70,90	61,44	98,28	66,58	59,48
Desvio:	0,014%		Tempo médio:		71,34 s

TABELA 7.6 – Desvio obtido com 500 tarefas.

500 Tarefas					
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Custo	9716	9716	10016	9726	9916
Tempo	6567,80	6570,20	6599,76	6637,38	6608,16
Desvio:	1,049%		Tempo médio:		6596,66 s

A ATP/PL apresentou um desvio expressivamente pequeno para todas as instâncias. Além disso, o tempo de processamento também variou pouco entre os experimentos, mostrando um bom desempenho da metodologia proposta.

Após a realização dos experimentos descritos anteriormente, a ATP/PL, apresentada na Seção 5.1, a SA e a SA_20 descritas na Seção 5.2, foram aplicadas às instâncias descritas na Seção 5.3 visando tratar a segunda abordagem para o PET, que consiste na obtenção de soluções variadas para o problema, ou seja, na solução do PET com diversos números de tripulações. O número de tripulações obtido nas soluções apresentadas por tais metodologias, em alguns casos, foram diferentes, então foi utilizado como referência, para determinar os número de tripulações usados na segunda abordagem do PET, o número cuja solução apresentou o menor custo (soluções obtidas pela ATP/PL). Os resultados obtidos por tal abordagem são apresentados nas Tabelas 7.7, 7.8, 7.9, 7.10 e 7.11. Nessas tabelas, o valor “-” é usado para indicar que a ATP/PL não obteve soluções para os problemas, e o valor “*” usado para indicar que o experimento com a SA não foi realizado, pois a SA_20 (SA executada por 20 vezes) não conseguiu obter soluções válidas para o problema em questão.

TABELA 7.7 – Resultados obtidos com 25 tarefas para a segunda abordagem do PET.

Método de solução	Número de tripulações	Horas extras (min)	Tempo ocioso (min)	Tempo de sobreposição (min)	Custo total da solução	Tempo (s)
ATP/PL	10	-	-	-	-	-
	11	-	-	-	-	-
	13	0	2821	0	2821	0,13
	14	0	3301	0	3301	0,13
SA	10	*	*	*	*	*
	11	*	*	*	*	*
	13	0	2821	0	2821	2,00
	14	0	3301	0	3301	2,02
SA_20	10	208	1625	36	1833 [^]	37,33
	11	98	1975	16	2073 [^]	42,46
	13	0	2821	0	2821	45,23
	14	0	3301	0	3301	43,43

TABELA 7.8 – Resultados obtidos com 50 tarefas para a segunda abordagem do PET.

Método de solução	Número de tripulações	Horas extras (min)	Tempo ocioso (min)	Tempo de sobreposição (min)	Custo total da solução	Tempo (s)
ATP/PL	18	-	-	-	-	-
	19	-	-	-	-	-
	21	0	3080	0	3080	7,67
	22	0	3560	0	3560	7,87
SA	18	*	*	*	*	*
	19	*	*	*	*	*
	21	0	3080	0	3080	43,27
	22	0	3560	0	3560	43,63
SA_20	18	83	1779	56	1862 [^]	896,50
	19	76	2202	6	2278 [^]	901,98
	21	0	3080	0	3080	898,96
	22	0	3560	0	3560	918,23

Deve-se destacar que, em todos os experimentos, as soluções obtidas por todas as metodologias (ATP/PL, SA e SA_20) não apresentaram tempo excessivo, porém nos casos em que o número de tripulações foi inferior ao obtido pelas soluções apresentadas para a primeira abordagem do PET a SA_20 apresentou tempo de sobreposição, ou seja, soluções inválidas. Nesses casos, a ATP/PL não obteve soluções, pois tal método não conseguiu solucionar o PPC de uma forma inteira com o número de colunas (tripulações) desejado. Para esses casos, como a SA_20 não obteve soluções válidas, a SA não foi utilizada.

TABELA 7.9 – Resultados obtidos com 100 tarefas para a segunda abordagem do PET.

Método de solução	Número de tripulações	Horas extras (min)	Tempo ocioso (min)	Tempo de sobreposição (min)	Custo total da solução	Tempo (s)
ATP/PL	38	-	-	-	-	-
	39	-	-	-	-	-
	41	0	7875	0	7875	2,12
	42	0	8355	0	8355	5,11
SA	38	*	*	*	*	*
	39	*	*	*	*	*
	41	0	7875	0	7875	8,12
	42	0	8355	0	8355	8,00
SA_20	38	0	6442	7	6442 [^]	159,87
	39	0	6917	2	6917 [^]	170,88
	41	0	7875	0	7875	167,16
	42	0	8355	0	8355	179,03

É interessante notar que a ATP/PL não obteve soluções nesses casos, pois como são inseridas no PPC apenas colunas válidas, ela não foi capaz de encontrar um conjunto de colunas (número fixo de colunas) que resolvesse os problemas. Além disso, a SA_20 também não obteve soluções válidas para nenhum desses casos, o que nos leva a acreditar que não há soluções factíveis para esse número de tripulações. Entretanto, isso não pode ser afirmado, pois soluções exatas para tais casos não são conhecidas. Para os casos em que o número de tripulações foi maior do que os obtidos pelas soluções apresentadas na primeira abordagem do PET, a ATP/PL assim como a SA e a SA_20 conseguiu obter as mesmas soluções nas instâncias com 25, 50 e 100 tarefas.

Já para as instâncias com 250 e 500 tarefas, o desempenho da ATP/PL foi expressivamente melhor do que o da SA e da SA_20, pois tal metodologia apresentou soluções com custo significativamente menor em um menor tempo de processamento.

TABELA 7.10 – Resultados obtidos com 250 tarefas para a segunda abordagem do PET.

Método de solução	Número de tripulações	Horas extras (min)	Tempo ocioso (min)	Tempo de sobreposição (min)	Custo total da solução	Tempo (s)
ATP/PL	79	-	-	-	-	-
	80	-	-	-	-	-
	82	909	8037	0	8946	236,48
	83	775	8383	0	9158	181,61
SA	79	*	*	*	*	*
	80	*	*	*	*	*
	82	1291	8419	0	9710	242,41
	83	1118	8726	0	9844	224,74
SA_20	79	1915	7613	10	9528 [^]	3737,32
	80	1631	7804	5	9435 [^]	3801,24
	82	1246	8374	0	9620	3776,08
	83	1096	8704	0	9800	3764,21

TABELA 7.11 – Resultados obtidos com 500 tarefas para a segunda abordagem do PET.

Método de solução	Número de tripulações	Horas extras (min)	Tempo ocioso (min)	Tempo de sobreposição (min)	Custo total da solução	Tempo (s)
ATP/PI	143	-	-	-	-	-
	144	-	-	-	-	-
	146	1446	8440	0	9886	6672,41
	147	1277	8751	0	10028	6664,87
SA	143	*	*	*	*	*
	144	*	*	*	*	*
	146	*	*	*	*	*
	147	2229	9703	0	11932	7115,23
SA_20	143	2694	8366	118	11060 [^]	142336,06
	144	2642	8769	93	11411 [^]	144475,34
	146	2585	9596	17	12181 [^]	142972,60
	147	2175	9649	0	11824	143510,40

Nas tabelas referentes à segunda abordagem do PET, os custos das soluções “inválidas” obtidas pela SA_20 (custos acompanhados do símbolo $\hat{\cdot}$) apresentam apenas os valores das restrições não-essenciais. Nesses casos, os custos referentes às restrições essenciais não foram inseridos na coluna “Custo total da solução” apenas por uma questão de visualização, mas os custos reais, obtidos nos experimentos, são formados pela Equação 5.1 (ver Apêndice B).

Todos os experimentos realizados neste trabalho foram executados em um *laptop* Toshiba com processador Intel Celeron® de 2.0 GHz e 256Mb de memória RAM com a arquitetura Microsoft Windows XP. As implementações foram desenvolvidas na linguagem C++, através do ambiente de programação Borland C++ Builder 6.0, com chamadas à biblioteca CPLEX (ver ILOG (2001)).

Os parâmetros utilizados na ATP/PL, na SA e na SA_20, e algumas informações a respeito da execução de tais metodologias, para todos os experimentos, são mostrados nos Apêndices A e B.

CAPÍTULO 8

CONCLUSÃO

O problema de escalonamento de tripulações (PET) é um problema de grande importância prática no planejamento das operações nas empresas de transporte coletivo. Este trabalho teve por objetivo fazer uma revisão de alguns trabalhos voltados para o tratamento do PET descritos na literatura e apresentar uma metodologia ainda não explorada para resolver tal problema.

Para solucionar o PET, foi implementada uma heurística híbrida, chamada de ATP/PL, baseada na técnica de geração de colunas, implementada através da interação do Algoritmo de Treinamento Populacional (ATP) com a Programação Linear (PL). Devido ao fato de o ATP ser um algoritmo da família dos algoritmos genéticos, foi preciso um tempo extra, e vários experimentos, para calibração de seus parâmetros. Mas, ao mesmo tempo ele mostrou ser bastante robusto para resolver o sub-problema de geração de colunas para instâncias pequenas, médias e grandes.

A influência da função de avaliação (função g) utilizada no ATP foi considerável, conduzindo tal método a gerar colunas de excelente qualidade, e de forma direcionada ao PPC, resultando em melhores soluções para tal problema.

Comparando as metodologias ATP/PL e *Simulated Annealing* (SA), verificou-se que a ATP/PL apresentou resultados de excelente qualidade ao resolver problemas de instâncias pequenas e grandes, enquanto a SA não conseguiu resultados tão bons para as maiores. O tempo de processamento também diferiu significativamente entre tais metodologias, assim como a qualidade dos resultados obtidos.

É interessante notar também (ver Apêndice A) que, a maior parte do tempo de processamento da ATP/PL, para problemas grandes, foi utilizada para resolução da PLI, através do *software* de otimização CPLEX.

A ATP/PL também mostrou robustez, pois mesmo com a inserção de uma nova restrição no problema impondo-se um número fixo de tripulações (segunda abordagem do PET) ela conseguiu obter boas soluções. Além disso, como mostrado no Capítulo 7, ela obteve um desvio expressivamente pequeno em todas as soluções obtidas, para todas as instâncias utilizadas.

Este trabalho se releva segundo diversos aspectos. Primeiramente, por difundir o estudo do PET no Brasil, onde tal problema é resolvido de forma precária pela maioria, senão

todas, as empresas do setor de transporte coletivo, o que impossibilita às vezes a aplicação de certas normas trabalhistas e sindicais. Um segundo aspecto importante e que justifica o desenvolvimento deste trabalho é o fato de que são levadas em consideração características dos problemas reais, já que boa parte dos métodos heurísticos encontrados na literatura é desenvolvida para problemas cujas características pouco condizem com a realidade.

Além disso, a aplicação do ATP (que é um algoritmo evolutivo) de uma forma interativa com a PL, para gerar colunas utilizando informações dessa para direcionar o seu processo evolutivo (através das variáveis duais na função de aptidão g) é inédita, e em princípio, parece ser adequada para a resolução de problemas onde a técnica de geração de colunas é indicada, visto que ela trabalha de uma maneira praticamente independente do problema a ser tratado.

Por fim, os resultados obtidos demonstram o potencial da ATP/PL, uma vez que em todas as instâncias adotadas ela apresentou resultados expressivamente melhores, em um menor tempo de processamento em relação à metaheurística *Simulated Annealing*, que por sua vez apresentou, em trabalhos anteriores, excelentes resultados para problemas reais.

Como extensão deste trabalho, sugere-se maiores esforços no sentido de encontrar técnicas mais sofisticadas, que consigam diminuir ainda mais o tempo de processamento da ATP/PL (principalmente durante a execução da PLI). Além disso, seria interessante também estudar uma maneira de modelar as instâncias apresentadas por Beasley e Cao (1996) visando obter uma maneira de penalizar os custos de um turno durante a execução do ATP, permitindo assim comparar as duas metodologias. Uma outra possível, e interessante, extensão deste trabalho seria a aplicação da ATP/PL a outros problemas combinatoriais, pois dessa forma ficaria mais claro o quão satisfatório é o seu desempenho.

REFERÊNCIAS BIBLIOGRÁFICAS

- Associação Nacional de Transporte Público (ANTP). Transporte público: os grandes desafios do transporte urbano, 2005. Disponível em: <<http://www.antp.org.br>>. Acesso em: 20 jan. 2005. 22
- Arakaki, R. G. I.; Lorena, L. A. N. A constructive genetic algorithm for the maximal covering location problem. In: Metaheuristics International Conference (MIC), 4., 2001, Porto. **Proceedings...** Porto: MIC, 2001. Disponível em: <<http://www.lac.inpe.br/~lorena/arakaki/cgalap.pdf>>. Acesso em: 20 Jan. 2005. 40
- Bazaraa, M. S.; Jarvis, J. J.; Sherali, H. D. **Linear programming and network flows**. New York: John Wiley & Sons, 1990. 704 p. 39
- Beasley, J. E. An algorithm for the set covering problem. **European Journal of Operational Research**, v. 31, p. 85–93, 1987. 28
- Beasley, J. E.; Cao, B. A tree search algorithm for the crew scheduling problem. **European Journal of Operational Research**, v. 94, p. 517–526, 1996. 21, 24, 27, 43, 45, 46, 52, 74
- Bouzada, C. F. **Análise das despesas administrativas no custo do transporte coletivo por ônibus no município de Belo Horizonte**. 2002. 153 p. Dissertação (Mestrado em Administração Pública) – Escola de Governo, Fundação João Pinheiro, Belo Horizonte. 2002. 22
- Colin, R. R. Genetic algorithm. In:_____. **Modern heuristic techniques for combinatorial problems**. London: McGraw-Hill, 1995. p. 151–196. 39
- Daduna, J. R.; Mojsilovic, M. Computer-aided vehicle and duty scheduling using HOT programme system. In: Daduna, J. R.; Wren, A. (eds.) **Computer-aided transit scheduling**. Berlin: Springer-Verlag, 1988. p. 133–146. 30
- Dallaire, A.; Fleurent, C.; Rousseau, J. M. Dynamic constraint generation in CrewOpt, a column generation approach for transit crew scheduling. In: International Conference on Computer-Aided Scheduling of Public Transport (CASPT), 9., 2004, San Diego. **Proceedings...** San Diego: CASPT, 2004. 21, 24, 30
- Dantzig, G. B.; Wolfe, P. Decomposition principle for linear programs. **Operations Research**, v. 8, p. 101–111, 1960. 39

Desaulniers, G. Bus and driver scheduling in urban mass transit systems. In: Travel and Transportation Workshop. Institute for Mathematics and its Applications, 2002, Minneapolis. **Proceedings...** Minneapolis: University of Minnesota, 2002. [24](#)

Desrochers, M.; Gilbert, J.; Sauvé, M.; Soumis, F. CREW-OPT: subproblem modeling in a column generation approach to urban crew scheduling. In: Desrochers, M.; Rousseau, J. M. (eds.) **Computer-aided transit scheduling**. Berlin: Springer-Verlag, 1992. p. 395–406. [30](#)

Desrochers, M.; Soumis, F. A column generation approach to the urban transit crew scheduling problem. **Transportation Science**, v. 23, n. 1, p. 1–13, 1989. [21](#), [24](#), [45](#)

Desrosiers, J.; Rousseau, J. M. Results obtained with Crew-Opt: a column generation method for transit crew scheduling. In: Daduna, J. R.; Branco, I.; Paixão, J. M. P. (eds.) **Computer-aided transit scheduling**. Berlin: Springer-Verlag, 1995. p. 349–358. [30](#)

Dias, T. G.; Sousa, J. P.; Cunha, J. F. A genetic algorithm for the bus driver scheduling problem. In: Metaheuristics International Conference (MIC), 4., 2001, Porto. **Proceedings...** Porto: MIC, 2001. [23](#), [28](#)

Dorigo, M.; Maniezzo, V.; Colorni, A. The ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics - Part B**, v. 26, n. 1, p. 29–41, 1996. [29](#)

Elias, S. E. G. **The use of digital computers in the economic scheduling for both man and machine in public transport**. Kansas: Kansas State University, 1964. [21](#)

Ernst, A.; Jiang, H.; Krishnamoorthy, M.; Owens, B.; Sier, D. An annotated bibliography of personnel scheduling and rostering. In: Jiang, H.; Krishnamoorthy, M.; Sier, D. (eds.) **Annals of operations research**. Netherlands: Kluwer Academic Publishers, 2004. p. 21–144. [31](#)

Fores, S.; Proll, L.; Wren, A. TRACS II: a hybrid IP/heuristic driver scheduling system for public transport. **Journal of the Operational Research Society**, v. 53, p. 1093–1100, 2002. [29](#)

Forsyth, P.; Wren, A. An ant system for bus driver scheduling. In: International Workshop on Computer-Aided Scheduling of Public Transport, 7., 1997, Boston. **Proceedings...** Massachusetts: MIT, 1997. [22](#), [23](#), [24](#), [29](#)

- Gintner, V.; Kliwer, N.; Suhl, L. A crew scheduling approach for public transit enhanced with aspects from vehicle scheduling. In: International Conference on Computer-Aided Scheduling of Public Transport (CASPT), 9., 2004, San Diego. **Proceedings...** San Diego: CASPT, 2004. [21](#)
- Glover, F. Tabu search: part I. **ORSA Journal on Computing**, v. 1, n. 3, p. 190–206, 1989a. [21](#)
- . Tabu search: part II. **ORSA Journal on Computing**, v. 2, n. 1, p. 04–32, 1989b. [21](#)
- Goldberg, D. E. **Genetic algorithms in search, optimization and machine learning**. Boston: Addison-Wesley, 1989. 432 p. [21](#), [39](#)
- Groot, S.; Huisman, D. Vehicle and crew scheduling: solving large real-world instances with an integrated approach. In: International Conference on Computer-Aided Scheduling of Public Transport (CASPT), 9., 2004, San Diego. **Proceedings...** San Diego: CASPT, 2004. [21](#)
- Hamer, N.; Séguin, L. The HASTUS system: New algorithms and modules for the 90s. In: Desrochers, M.; Rousseau, J. M.; Beckmann, M. J.; Krelle, W. (eds.) **Computer-aided transit scheduling**. Berlin: Springer-Verlag, 1992. p. 17–29. [30](#)
- Holland, J. H. **Adaptation in natural and artificial systems**. Michigan: University of Michigan Press, 1975. 211 p. [39](#)
- Ignizio, J. P.; Cavalier, T. M. **Linear programming**. New Jersey: Prentice-Hall, 1994. 666 p. [37](#)
- ILOG CPLEX 7.5 Reference Manual**. 7.5v. 610 p. ©Copyright by ILOG, France, 2001. [71](#)
- Jacquet-Lagrèze, E.; Lebbar, M. A column generation model for a scheduling problem with maintenance constraints. In: Wallace, M.; Caseau, Y.; Jacquet-Lagrèze, E.; Simonis, H.; Pesant, G. (eds.) **Electronic notes in discrete mathematics**. Amsterdam: Elsevier, 2000. p. 1–12. [38](#)
- Kirkpatrick, S.; Gellat, D. C.; Vecchi, M. P. Optimization by simulated annealing. **Science**, p. 671–680, May 1983. [21](#), [36](#)
- Kwan, A.; Parker, M.; Kwan, R.; Fores, S.; Proll, L.; Wren, A. Recent advances in TRACS. In: International Conference on Computer-Aided Scheduling of Public

Transport (CASPT), 9., 2004, San Diego. **Proceedings...** San Diego: CASPT, 2004. 21, 29

Kwan, A. S. K.; Kwan, R. S. K.; Wren, A. Driver scheduling using genetic algorithms with embedded combinatorial traits. In: Wilson, N. H. M. (ed.) **Computer-aided transit scheduling**. Berlin: Springer-Verlag, 1999. p. 81–102. 23

Lee, C. K. The integrated scheduling and rostering problem of train driver using genetic algorithm. In: International Conference on Computer-Aided Scheduling of Public Transport (CASPT), 9., 2004, San Diego. **Proceedings...** San Diego: CASPT, 2004. 21

Lorena, L. A. N.; Furtado, J. C. Constructive genetic algorithm for clustering problems. **Evolutionary Computation**, v. 9, n. 3, p. 309–327, 2001. Disponível em: <http://www.lac.inpe.br/~lorena/cga/cga_clus.pdf>. Acesso em: 20 Jan. 2005. 40

Lorena, L. A. N.; Lopes, L. S. Computational experiments with genetic algorithms applied to set covering problems. **Pesquisa Operacional**, v. 16, n. 1, p. 41–53, jun. 1996. 40

Lorena, L. A. N.; Narciso, M. G.; Beasley, J. E. A constructive genetic algorithm for the generalized assignment problem. **Evolutionary Optimization**, 2002. Disponível em: <http://www.lac.inpe.br/~lorena/gap/cga_revised.pdf>. Acesso em: 20 Jan. 2005. 40

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, Aug. 2001. 21, 22, 23, 24, 28

Manington, P. D.; Wren, A. Experiences with a bus scheduling algorithm which saves vehicles. In: Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services, 1975, Chicago. **Proceedings...** Chicago, 1975. 21

Marinho, E. H.; Ochi, L. S.; Drummond, L. M. A.; Souza, M. J. F.; Silva, G. P. Busca tabu aplicada ao problema de programação de tripulações de ônibus urbano. In: Simpósio Brasileiro de Pesquisa Operacional (SBPO), 36., 2004, São João del Rei. **Anais...** Rio de Janeiro: SOBRAPO, 2004. 1 CD-ROM. 21, 22, 24

Mauri, G. R. **Resolução do problema de programação de tripulações de um sistema de transporte público via simulated annealing**. Ouro Preto:

Departamento de Ciência da Computação - Universidade Federal de Ouro Preto, 2003. Disponível em: <<http://www.lac.inpe.br/~mauri/relTec0203.pdf>>. Acesso em: 20 jan. 2005. 21, 24, 25, 30, 31, 33, 34, 36, 43, 51, 59

Mauri, G. R.; Lorena, L. A. N. Driver scheduling generation using a population training algorithm. In: Brazilian Symposium in Neural Networks (SBRN), 8., 2004, São Luís. **Anais...** Porto Alegre: SBC, 2004a. 1 CD-ROM. Disponível em: <http://www.lac.inpe.br/~mauri/arquivos/mauri_lorena_sbrn04.pdf>. Acesso em: 20 jan. 2005. 56

———. Método interativo para resolução do problema de escalonamento de tripulações. In: Simpósio Brasileiro de Pesquisa Operacional (SBPO), 36., 2004, São João del Rei. **Anais...** Rio de Janeiro: SOBRAPO, 2004b. 1 CD-ROM. Disponível em: <http://www.lac.inpe.br/~mauri/arquivos/mauri_lorena_sbpo04.pdf>. Acesso em: 20 jan. 2005. 56

Mingozzi, A.; Boschetti, M. A.; Ricciardelli, S.; Bianco, L. A set partitioning approach to the crew scheduling problem. **Operations Research**, v. 47, n. 6, p. 873–888, nov. 1999. 27, 43, 45, 46

Oliveira, A. C. M. **Treinamento populacional em heurísticas: aplicações em otimização**. 2002. Proposta de Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos. 2002. 40

Oliveira, A. C. M.; Lorena, L. A. N. Sequencing cutting patterns and VLSI gates by population training algorithms. In: EURO/INFORMS, 2003, Istanbul. **Proceedings...** Instambul: EURO/IFORS, 2003. Disponível em: <http://www.lac.inpe.br/~lorena/alexandre/ptamosp_IJOC.pdf>. Acesso em: 20 Jan. 2005. 41

Ribeiro Filho, G.; Lorena, L. A. N. A constructive evolutionary approach to the machine-part cell formation problem. In: Fleury, A.; Yoshisaki, H.; Guimaraes, L. B. M.; Ribeiro, J. L. D. (eds.) **Buildings competencies for international manufacturing: perspectives for developing countries**. Porto Alegre: UFRGS/FEENG, 2000a. p. 340–348. 40

———. Constructive genetic algorithm an column generation: an application to graph coloring. In: Conference of the Association of Asian-Pacific Operations Research Societies within IFORS (APORS), 5., 2000, Singapore. **Proceedings...** Singapore: IFORS, 2000b. Disponível em:

<http://www.lac.inpe.br/~lorena/geraldo/cga_column.pdf>. Acesso em: 20 Jan. 2005. 40

———. A Constructive evolutionary approach to school timetabling. **Computer Science**, v. 2037, p. 130–139, 2001. 40

Rousseau, J. M.; Lessard, R.; Blais, J. Y. Enhancements to the HASTUS crew scheduling algorithm. In: Rousseau, J. M. (ed.) **Computer scheduling of public transport**. New York: North-Holand, 1985. p. 295–310. 30

Shen, Y.; Kwan, R. S. K. Tabu search for driver scheduling. In: Voß, S.; Daduna, J. (eds.) **Computer-aided scheduling of public transport**. Berlin: Springer-Verlag, 2001. p. 121–136. 21, 22, 23, 24, 28

Sierksma, G. **Linear and integer programming: theory and practice**. New York: Marcel Dekker, Inc., 1996. 673 p. 37

Souza, M. J. F.; Cardoso, L. X. T.; Silva, G. P. Programação de tripulações de ônibus urbano: Uma abordagem heurística. In: Simpósio Brasileiro de Pesquisa Operacional (SBPO), 35., 2003, Natal. **Anais...** Rio de Janeiro: SOBRAPO, 2003. 1 CD-ROM. 22, 24, 30

Valouxis, C.; Housos, E. Combined bus and driver scheduling. **Computers and Operation Research Journal**, v. 29, n. 3, p. 243–259, Mar. 2002. 24, 28

Vasko, F. J.; Wolf, F. E. Solving large set covering problems on a personal computer. **Computers & Operations Research**, v. 15, p. 115–121, 1988. 28

Wedelin, D. The design of a 0-1 integer optimizer and its application in the Carmen system. **European Journal of Operational Research**, v. 87, p. 722–730, 1995. 30

Wilhelm, E. B. Overview of the RUCUS package driver run cutting program (RUNS). In: Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services, 1975, Chicago. **Proceedings...** Chicago, 1975. 30

Wilhelm, W. E. A Technical Review of Column Generation in Integer Programming. **Optimization and Engineering**, v. 2, p. 159–200, 2001. 21, 38, 45

Wilson, N. H. M. **Computer-aided transit scheduling**. Berlin: Springer-Verlag, 1999. 475 p. 21

Wren, A. Scheduling vehicles and their drivers: forty years' experience. In: International Conference on Computer-Aided Scheduling of Public Transport (CASPT), 9., 2004, San Diego. **Proceedings...** San Diego: CASPT, 2004. [21](#), [24](#), [31](#)

Wren, A.; Fores, S.; Kwan, A.; Kwan, R.; Parker, M.; Proll, L. A flexible system for scheduling drivers. **Journal of Scheduling**, v. 6, p. 437–455, 2003. [21](#), [22](#), [23](#), [24](#), [26](#), [29](#), [43](#)

Wren, A.; Rousseau, J. M. Bus driver scheduling: an overview. In: Daduna, J. R.; Branco, I.; Paixão, J. M. P. (eds.) **Computer-aided transit scheduling**. Berlin: Springer-Verlag, 1995. p. 173–18. [21](#), [23](#), [24](#), [26](#)

Yunes, T. H. **Problemas de escalonamento no transporte coletivo: programação por restrições e outras técnicas**. 2000. 127 p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Computação, Universidade Estadual de Campinas (UNICAMP), Campinas. 2000. [24](#), [30](#)

Yunes, T. H.; Moura, A. V.; Souza, C. C. **Solving large scale crew scheduling problems with constraint programming and integer programming**. Campinas: Instituto de Computação, Universidade Estadual de Campinas (UNICAMP), 1999. [24](#)

———. Solving very large crew scheduling problems to optimality. In: ACM Symposium on Applied Computing (SAC), 14., 2000, Itália. **Proceedings...** New York: ACM, 2000. [24](#)

———. Hybrid column generation approaches for solving real world crew management problems. In: Conferência Latinoamericana de Informática (CLEI), 27., 2001, Venezuela. **Proceedings...** Venezuela: CLEI, 2001. [24](#), [25](#)

APÊNDICE A

PARÂMETROS E INFORMAÇÕES DAS ATP/PL's

Nas próximas tabelas são apresentados os parâmetros utilizados e algumas informações a respeito das ATP/PL's descritas no Capítulo 6.

TABELA A.1 – Parâmetros e informações da ATP+PL.

Parâmetros	
Tamanho da população inicial	100000
<i>Step</i>	10^{-10}
Número máximo de gerações	150000
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1
Peso para as restrições não-essenciais	1
Número de iterações	-
Numero inicial de colunas	-
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	53187
N. de tripulações	178
Total de horas extras	1870
Tempo ocioso total	24224
Tempo excessivo total	0
Tempo de sobreposição total	0
Custo total da solução	26094
Tempo de processamento da interação (ATP e PL)	-
Tempo de processamento da PLI	-
Tempo total de processamento (s)	11702,34
Semente aleatória	-

TABELA A.2 – Parâmetros e informações da ATP/PL_1.

Parâmetros	
Tamanho da população inicial	6000
<i>Step</i>	10^{-10}
Número máximo de gerações	8000
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1
Peso para as restrições não-essenciais	1
Número de iterações	-
Numero inicial de colunas	-
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	77
N. de tripulações	-
Total de horas extras	-
Tempo ocioso total	-
Tempo excessivo total	-
Tempo de sobreposição total	-
Custo total da solução	-
Tempo de processamento da interação (ATP e PL)	69,32
Tempo de processamento da PLI	-
Tempo total de processamento (s)	-
Semente aleatória	-

TABELA A.3 – Parâmetros e informações da ATP/PL_2.

Parâmetros	
Tamanho da população inicial	6000
<i>Step</i>	10^{-10}
Número máximo de gerações	8000
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1
Peso para as restrições não-essenciais	1
Número de iterações	257
Numero inicial de colunas	1000
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	4346
N. de tripulações	156
Total de horas extras	1898
Tempo ocioso total	13692
Tempo excessivo total	0
Tempo de sobreposição total	0
Custo total da solução	15590
Tempo de processamento da interação (ATP e PL)	107,91
Tempo de processamento da PLI	6000,06
Tempo total de processamento (s)	6107,97
Semente aleatória	-

TABELA A.4 – Parâmetros e informações da ATP/PL_3.

Parâmetros	
Tamanho da população inicial	6000
<i>Step</i>	10^{-10}
Número máximo de gerações	8000
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1
Peso para as restrições não-essenciais	1
Número de iterações	500
Numero inicial de colunas	1000
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	125991
N. de tripulações	151
Total de horas extras	1567
Tempo ocioso total	10961
Tempo excessivo total	0
Tempo de sobreposição total	0
Custo total da solução	12528
Tempo de processamento da interação (ATP e PL)	797,18
Tempo de processamento da PLI	6000,22
Tempo total de processamento (s)	6797,40
Semente aleatória	-

TABELA A.5 – Parâmetros e informações da ATP/PL_4.

Parâmetros	
Tamanho da população inicial	100
<i>Step</i>	10^{-17}
Número máximo de gerações	500
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1
Peso para as restrições não-essenciais	1
Número de iterações	2000
Numero inicial de colunas	5000
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	52851
N. de tripulações	151
Total de horas extras	1333
Tempo ocioso total	10727
Tempo excessivo total	0
Tempo de sobreposição total	0
Custo total da solução	12060
Tempo de processamento da interação (ATP e PL)	224,07
Tempo de processamento da PLI	6001,01
Tempo total de processamento (s)	6225,08
Semente aleatória	-

TABELA A.6 – Parâmetros e informações da ATP/PL_5.

Parâmetros	
Tamanho da população inicial	100
<i>Step</i>	10^{-17}
Número máximo de gerações	500
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1
Peso para as restrições não-essenciais	1
Número de iterações	2000
Numero inicial de colunas	5000
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	15656
N. de tripulações	149
Total de horas extras	1304
Tempo ocioso total	9738
Tempo excessivo total	0
Tempo de sobreposição total	0
Custo total da solução	11042
Tempo de processamento da interação (ATP e PL)	196,62
Tempo de processamento da PLI	6000,01
Tempo total de processamento (s)	6196,63
Semente aleatória	-

TABELA A.7 – Parâmetros e informações da ATP/PL_6.

Parâmetros	
Tamanho da população inicial	100
<i>Step</i>	10^{-17}
Número máximo de gerações	500
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1
Peso para as restrições não-essenciais	1
Número de iterações	10000
Numero inicial de colunas	5000
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	99364
N. de tripulações	147
Total de horas extras	1514
Tempo ocioso total	8988
Tempo excessivo total	0
Tempo de sobreposição total	0
Custo total da solução	10502
Tempo de processamento da interação (ATP e PL)	800,04
Tempo de processamento da PLI	6001,87
Tempo total de processamento (s)	6801,91
Semente aleatória	-

TABELA A.8 – Parâmetros e informações da ATP/PL.

Parâmetros	
Tamanho da população inicial	100
<i>Step</i>	10^{-17}
Número máximo de gerações	500
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1000
Peso para as restrições não-essenciais	1
Número de iterações	10000
Numero inicial de colunas	5000
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	78082
N. de tripulações	145
Total de horas extras	1601
Tempo ocioso total	8115
Tempo excessivo total	0
Tempo de sobreposição total	0
Custo total da solução	9716
Tempo de processamento da interação (ATP e PL)	567,45
Tempo de processamento da PLI	6000,36
Tempo total de processamento (s)	6567,80
Semente aleatória	-

TABELA A.9 – Parâmetros e informações da ATP/PL_7.

Parâmetros	
Tamanho da população inicial	100
<i>Step</i>	10^{-17}
Número máximo de gerações	500
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1000
Peso para as restrições não-essenciais	1
Número de iterações	10000
Numero inicial de colunas	5000
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	74930
N. de tripulações	146
Total de horas extras	1525
Tempo ocioso total	8519
Tempo excessivo total	0
Tempo de sobreposição total	0
Custo total da solução	10044
Tempo de processamento da interação (ATP e PL)	630,71
Tempo de processamento da PLI	6000,28
Tempo total de processamento (s)	6630,99
Semente aleatória	-

TABELA A.10 – Parâmetros e informações da ATP/PL8.

Parâmetros	
Tamanho da população inicial	100
<i>Step</i>	10^{-17}
Número máximo de gerações	500
Percentual base	40
Percentual de mutação	20
Tamanho da vizinhança	7
Peso para as restrições essenciais	1000
Peso para as restrições não-essenciais	1
Número de iterações	10000
Numero inicial de colunas	5000
Tempo limite para a PLI (s)	6000
<i>Gap</i> limite para a PLI (%)	0
Função objetivo limite para a PLI	0
Informações	
Numero final de colunas	3
N. de tripulações	-
Total de horas extras	-
Tempo ocioso total	-
Tempo excessivo total	-
Tempo de sobreposição total	-
Custo total da solução	-
Tempo de processamento da interação (ATP e PL)	279,92
Tempo de processamento da PLI	-
Tempo total de processamento (s)	-
Semente aleatória	-

A Tabela A.11 apresenta os parâmetros utilizados pela ATP/PL (descrita na Seção 5.1) para todas as instâncias (25, 50, 100, 250 e 500 tarefas) em todos os experimentos (da primeira e segunda abordagens do PET).

TABELA A.11 – Parâmetros da ATP/PL para todos os experimentos.

Número de tarefas	25	50	100	250	500
Tamanho da população inicial	100	100	100	100	100
<i>Step</i>	1×10^{-17}	1×10^{-17}	1×10^{-17}	1×10^{-17}	1×10^{-17}
Número máximo de gerações	500	1000	1000	1000	500
Percentual base	40	40	40	40	40
Percentual de mutação	20	20	20	20	20
Tamanho da vizinhança	7	7	7	7	7
Peso para as restrições essenciais	1000	1000	1000	1000	1000
Peso para as restrições não-essenciais	1	1	1	1	1
Número de iterações	3	50	15	700	10000
Numero inicial de colunas	250	500	1000	2500	5000
Tempo limite para a PLI (s)	6000	6000	6000	6000	6000
<i>Gap</i> limite para a PLI (%)	0	0	0	0	0
Função objetivo limite para a PLI	0	0	0	0	0

A seguir, são apresentadas algumas características das melhores soluções obtidas nos experimentos realizados, com a ATP/PL, para a *primeira* abordagem do PET (obter uma solução sem considerar um número fixo de tripulações). Nas tabelas seguintes, o experimento 1 representa a melhor solução obtida para cada instância (sem o uso de sementes aleatórias), e os demais, foram os utilizados para o cálculo do desvio.

TABELA A.12 – Experimentos com 25 tarefas para a primeira abordagem do PET.

Experimentos	1	2	3	4	5
Numero final de colunas	351	371	373	354	364
N. de tripulações	12	12	12	12	12
Total de horas extras	15	15	15	15	15
Tempo ocioso total	2356	2356	2356	2356	2356
Tempo excessivo total	0	0	0	0	0
Tempo de sobreposição total	0	0	0	0	0
Custo total da solução	2371	2371	2371	2371	2371
Tempo da interação (ATP e PL)	0,09	0,10	0,10	0,09	0,10
Tempo de processamento da PLI	0,01	0,00	0,00	0,01	0,01
Tempo total de processamento	0,10	0,10	0,10	0,10	0,11
Semente aleatória	-	503680	596428	34897	16865

TABELA A.13 – Experimentos com 50 tarefas para a primeira abordagem do PET.

Experimentos	1	2	3	4	5
Numero final de colunas	1113	1375	1137	1376	1180
N. de tripulações	20	20	20	20	20
Total de horas extras	0	0	0	1	0
Tempo ocioso total	2600	2600	2600	2601	2600
Tempo excessivo total	0	0	0	0	0
Tempo de sobreposição total	0	0	0	0	0
Custo total da solução	2600	2600	2600	2602	2600
Tempo da interação (ATP e PL)	4,61	4,70	4,59	4,73	4,82
Tempo de processamento da PLI	0,02	0,14	0,01	0,01	0,01
Tempo total de processamento (s)	4,63	4,84	4,60	4,74	4,83
Semente aleatória	-	551696	98957	86677	579840

TABELA A.14 – Experimentos com 100 tarefas para a primeira abordagem do PET.

Experimentos	1	2	3	4	5
Numero final de colunas	2375	2861	3160	2258	2244
N. de tripulações	40	40	40	40	40
Total de horas extras	0	0	0	0	0
Tempo ocioso total	7395	7395	7395	7395	7395
Tempo excessivo total	0	0	0	0	0
Tempo de sobreposição total	0	0	0	0	0
Custo total da solução	7395	40	40	40	40
Tempo da interação (ATP e PL)	1,43	1,44	1,51	1,46	1,43
Tempo de processamento da PLI	0,45	0,04	0,02	0,03	0,04
Tempo total de processamento (s)	1,88	1,48	1,53	1,49	1,47
Semente aleatória	-	729492	917459	710763	89382

TABELA A.15 – Experimentos com 250 tarefas para a primeira abordagem do PET.

Experimentos	1	2	3	4	5
Numero final de colunas	29834	21200	37441	31099	29100
N. de tripulações	81	81	81	81	81
Total de horas extras	1103	1103	1103	1106	1103
Tempo ocioso total	7751	7751	7751	7754	7751
Tempo excessivo total	0	0	0	0	0
Tempo de sobreposição total	0	0	0	0	0
Custo total da solução	8854	8854	8854	8860	
Tempo da interação (ATP e PL)	60,16	56,89	58,23	60,06	58,19
Tempo de processamento da PLI	10,74	4,55	40,05	6,52	1,29
Tempo total de processamento (s)	70,90	61,44	98,28	66,58	59,48
Semente aleatória	-	428993	426823	102122	675

TABELA A.16 – Experimentos com 500 tarefas para a primeira abordagem do PET.

Experimentos	1	2	3	4	5
Numero final de colunas	78082	78084	84349	93366	87217
N. de tripulações	145	145	146	146	146
Total de horas extras	1601	1601	1511	1366	1461
Tempo ocioso total	8115	8115	8505	8360	8455
Tempo excessivo total	0	0	0	0	0
Tempo de sobreposição total	0	0	0	0	0
Custo total da solução	9716	9716	10016	9726	9916
Tempo da interação (ATP e PL)	567,45	570,07	598,98	636,55	608,10
Tempo de processamento da PLI	6000,36	6000,13	6000,78	6000,84	6000,06
Tempo total de processamento (s)	6567,80	6570,20	6599,76	6637,38	6608,16
Semente aleatória	-	628177	10560	52378	163158

As tabelas a seguir apresentam informações a respeito dos experimentos realizados, com a ATP/PL, para a *segunda* abordagem do PET (obter uma solução considerando um número fixo de tripulações). Nessas tabelas, o valor “-” significa que tal metodologia não obteve soluções factíveis para o problema em questão.

TABELA A.17 – Experimentos com 25 tarefas para a segunda abordagem do PET.

Número de tripulações	10	11	13	14
Numero final de colunas	-	-	313	276
Total de horas extras	-	-	0	0
Tempo ocioso total	-	-	2821	3301
Tempo excessivo total	-	-	0	0
Tempo de sobreposição total	-	-	0	0
Custo total da solução	-	-	2821	3301
Tempo da interação (ATP e PL)	-	-	0,12	0,12
Tempo de processamento da PLI	-	-	0,01	0,01
Tempo total de processamento (s)	-	-	0,13	0,13
Semente aleatória	-	-	-	-

TABELA A.18 – Experimentos com 50 tarefas para a segunda abordagem do PET.

Número de tripulações	18	19	21	22
Numero final de colunas	-	-	5338	966
Total de horas extras	-	-	0	0
Tempo ocioso total	-	-	3080	3560
Tempo excessivo total	-	-	0	0
Tempo de sobreposição total	-	-	0	0
Custo total da solução	-	-	3080	3560
Tempo da interação (ATP e PL)	-	-	7,64	7,87
Tempo de processamento da PLI	-	-	0,03	0,00
Tempo total de processamento (s)	-	-	7,67	7,87
Semente aleatória	-	-	-	-

TABELA A.19 – Experimentos com 100 tarefas para a segunda abordagem do PET.

Número de tripulações	38	39	41	42
Numero final de colunas	-	-	2178	6313
Total de horas extras	-	-	0	0
Tempo ocioso total	-	-	7875	8355
Tempo excessivo total	-	-	0	0
Tempo de sobreposição total	-	-	0	0
Custo total da solução	-	-	7875	8355
Tempo da interação (ATP e PL)	-	-	2,09	2,12
Tempo de processamento da PLI	-	-	0,03	2,98
Tempo total de processamento (s)	-	-	2,12	5,11
Semente aleatória	-	-	-	-

TABELA A.20 – Experimentos com 250 tarefas para a segunda abordagem do PET.

Número de tripulações	79	80	82	83
Numero final de colunas	-	-	21341	36450
Total de horas extras	-	-	909	775
Tempo ocioso total	-	-	8037	8383
Tempo excessivo total	-	-	0	0
Tempo de sobreposição total	-	-	0	0
Custo total da solução	-	-	8946	9158
Tempo da interação (ATP e PL)	-	-	84,50	89,51
Tempo de processamento da PLI	-	-	151,98	92,10
Tempo total de processamento (s)	-	-	236,48	181,61
Semente aleatória	-	-	-	-

TABELA A.21 – Experimentos com 500 tarefas para a segunda abordagem do PET.

Número de tripulações	143	144	146	147
Numero final de colunas	-	-	84571	82174
Total de horas extras	-	-	1446	1277
Tempo ocioso total	-	-	8440	8751
Tempo excessivo total	-	-	0	0
Tempo de sobreposição total	-	-	0	0
Custo total da solução	-	-	9886	10028
Tempo da interação (ATP e PL)	-	-	671,68	664,35
Tempo de processamento da PLI	-	-	6000,73	6000,53
Tempo total de processamento (s)	-	-	6672,41	6664,87
Semente aleatória	-	-	-	-

APÊNDICE B

PARÂMETROS E INFORMAÇÕES DAS SA's

A Tabela B.1 apresenta os parâmetros utilizados pela SA e pela SA_20 (descritas na Seção 5.2) para todas as instâncias (25, 50, 100, 250 e 500 tarefas) em todos os experimentos referentes à primeira e segunda abordagens do PET.

TABELA B.1 – Parâmetros da SA e SA_20 para todos os experimentos.

Número de tarefas	25	50	100	250	500
Numero máximo de iterações	700	10000	1000	10000	10000
Temperatura inicial	1000	1000	1000	1000	1000
Temperatura de congelamento	0,01	0,01	0,01	0,01	0,01
Taxa de resfriamento	0,975	0,975	0,975	0,975	0,975
Peso para as restrições essenciais	1000	1000	1000	1000	1000
Peso para as restrições não-essenciais	1	1	1	1	1
Numero inicial de tripulações	12	25	50	125	250
Tempo limite para a execução (s)	-	-	-	-	7000

Nas Tabelas B.2 e B.3 são apresentadas características das melhores soluções obtidas nos experimentos realizados, envolvendo as metodologias SA e a SA_20 respectivamente, para a *primeira* abordagem do PET (obter uma solução sem considerar um número fixo de tripulações).

TABELA B.2 – Experimentos da SA para a primeira abordagem do PET.

Número de tarefas	25	50	100	250	500
N. de tripulações	12	20	40	85	153
Total de horas extras	15	27	0	815	1254
Tempo ocioso total	2356	2627	7395	9383	11608
Tempo excessivo total	0	0	0	0	0
Tempo de sobreposição total	0	0	0	0	0
Custo total da solução	2371	2654	7395	10198	12862
Tempo total de processamento (s)	1,90	42,87	7,60	199,86	7061,52

TABELA B.3 – Experimentos da SA_20 para a primeira abordagem do PET.

Número de tarefas	25	50	100	250	500
N. de tripulações	12	20	40	83	153
Total de horas extras	15	0	0	1112	1096
Tempo ocioso total	2356	2600	7395	8720	11450
Tempo excessivo total	0	0	0	0	0
Tempo de sobreposição total	0	0	0	0	0
Custo total da solução	2371	2600	7395	9832	12546
Tempo total de processamento (s)	35,48	949,04	173,26	3749,02	143565,20

A seguir, são apresentadas características das melhores soluções obtidas nos experimentos realizados, envolvendo a metodologia SA para a *segunda* abordagem do PET (obter uma solução considerando um número fixo de tripulações). Nessas tabelas, o valor “*” significa que o experimento não foi realizado, pois a SA_20 (SA executada por 20 vezes) não conseguiu obter soluções válidas para o problema em questão.

TABELA B.4 – SA com 25 tarefas para a segunda abordagem do PET.

Número de tripulações	10	11	13	14
Total de horas extras	*	*	0	0
Tempo ocioso total	*	*	2821	3301
Tempo excessivo total	*	*	0	0
Tempo de sobreposição total	*	*	0	0
Custo total da solução	*	*	2821	3301
Tempo total de processamento (s)	*	*	2,00	2,02

TABELA B.5 – SA com 50 tarefas para a segunda abordagem do PET.

Número de tripulações	18	19	21	22
Total de horas extras	*	*	0	0
Tempo ocioso total	*	*	3080	3560
Tempo excessivo total	*	*	0	0
Tempo de sobreposição total	*	*	0	0
Custo total da solução	*	*	3080	3560
Tempo total de processamento (s)	*	*	43,27	43,63

TABELA B.6 – SA com 100 tarefas para a segunda abordagem do PET.

Número de tripulações	38	39	41	42
Total de horas extras	*	*	0	0
Tempo ocioso total	*	*	7875	8355
Tempo excessivo total	*	*	0	0
Tempo de sobreposição total	*	*	0	0
Custo total da solução	*	*	7875	8355
Tempo total de processamento (s)	*	*	8,12	8,00

TABELA B.7 – SA com 250 tarefas para a segunda abordagem do PET.

Número de tripulações	79	80	81	82
Total de horas extras	*	*	1291	1118
Tempo ocioso total	*	*	8419	8726
Tempo excessivo total	*	*	0	0
Tempo de sobreposição total	*	*	0	0
Custo total da solução	*	*	9710	9844
Tempo total de processamento (s)	*	*	242,41	224,74

TABELA B.8 – SA com 500 tarefas para a segunda abordagem do PET.

Número de tripulações	143	144	146	147
Total de horas extras	*	*	*	2229
Tempo ocioso total	*	*	*	9703
Tempo excessivo total	*	*	*	0
Tempo de sobreposição total	*	*	*	0
Custo total da solução	*	*	*	11932
Tempo total de processamento (s)	*	*	*	7115,23

As próximas tabelas apresentam informações a respeito das melhores soluções obtidas nos experimentos realizados com a SA_20 para a *segunda* abordagem do PET.

TABELA B.9 – SA_20 com 25 tarefas para a segunda abordagem do PET.

Número de tripulações	10	11	13	14
Total de horas extras	208	98	0	0
Tempo ocioso total	1625	1975	2821	3301
Tempo excessivo total	0	0	0	0
Tempo de sobreposição total	36	16	0	00
Custo total da solução	37833	18073	2821	3301
Tempo total de processamento (s)	37,33	42,46	45,23	43,43

TABELA B.10 – SA_20 com 50 tarefas para a segunda abordagem do PET.

Número de tripulações	18	19	21	22
Total de horas extras	83	76	0	0
Tempo ocioso total	1779	2202	3080	3560
Tempo excessivo total	0	0	0	0
Tempo de sobreposição total	56	6	0	0
Custo total da solução	57862	8278	3080	3560
Tempo total de processamento (s)	896,50	901,98	898,96	918,23

TABELA B.11 – SA_20 com 100 tarefas para a segunda abordagem do PET.

Número de tripulações	38	39	41	42
Total de horas extras	0	0	0	0
Tempo ocioso total	6442	6917	7875	8355
Tempo excessivo total	0	0	0	0
Tempo de sobreposição total	7	2	0	0
Custo total da solução	13442	8917	7875	8355
Tempo total de processamento (s)	159,87	170,88	167,16	179,03

TABELA B.12 – SA_20 com 250 tarefas para a segunda abordagem do PET.

Número de tripulações	79	80	81	82
Total de horas extras	1915	1631	1246	1096
Tempo ocioso total	7613	7804	8374	8704
Tempo excessivo total	0	0	0	0
Tempo de sobreposição total	10	5	0	0
Custo total da solução	19528	14435	9620	9800
Tempo total de processamento (s)	3737,32	3801,24	3776,08	3764,21

TABELA B.13 – SA_20 com 500 tarefas para a segunda abordagem do PET.

Número de tripulações	143	144	146	147
Total de horas extras	2694	2642	2585	2175
Tempo ocioso total	8366	8769	9596	9649
Tempo excessivo total	0	0	0	0
Tempo de sobreposição total	118	93	17	0
Custo total da solução	129060	104411	29181	11824
Tempo total de processamento (s)	142336,06	144475,34	142972,60	143510,40