

Universidade Federal de Santa Catarina  
Programa de Pós-graduação em Engenharia de Produção

**GERAÇÃO DE ESCALAS DE TRABALHO EM  
TRANSPORTE URBANO DE PASSAGEIROS: UMA  
TÉCNICA HEURÍSTICA PARA A FORMULAÇÃO DO  
PROBLEMA DE COBERTURA DE CONJUNTOS**

**João Renato Padula Castro**

Dissertação apresentada ao Programa  
de Pós-Graduação em Engenharia de  
Produção da Universidade Federal de  
Santa Catarina para obtenção do título  
de Mestre em Engenharia

2002

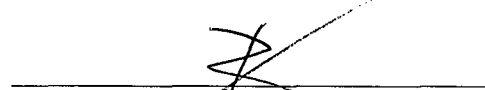
# JOÃO RENATO PADULA CASTRO

GERAÇÃO DE ESCALAS DE TRABALHO EM TRANSPORTE URBANO DE PASSAGEIROS: UMA TÉCNICA HEURÍSTICA PARA A FORMULAÇÃO DO PROBLEMA DE COBERTURA DE CONJUNTOS

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA OBTENÇÃO DO TÍTULO DE

**MESTRE EM ENGENHARIA**

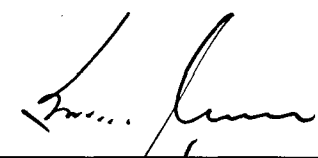
ESPECIALIDADE EM ENGENHARIA DE PRODUÇÃO E APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO



---

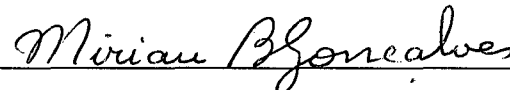
Prof. Ricardo Miranda Barcia PH.D.  
COORDENADOR DO CURSO

BANCA EXAMINADORA:



---

Prof. Sérgio Fernando Mayerle, DR.  
ORIENTADOR



---

Prof. Mirian Buss Gonçalves, DRA.



---

Prof. Antônio Sérgio Coelho, DR.

## RESUMO

O sistema de transporte coletivo urbano de passageiros é composto de várias partes, sendo que cada uma destas pode ser considerada como um problema distinto a ser equacionado.

A alocação de frota e tripulação são problemas de nível operacional que representam os maiores custos do sistema, no qual prevalece os custos com pessoal. Os modelos de alocação de tripulação existentes focam seu desenvolvimento nos métodos de otimização propriamente ditos, menosprezando a geração do conjunto inicial de escalas, que é a base para a alocação. Dentre estes métodos existentes, atualmente o mais utilizado é o problema de cobertura de conjuntos, cuja maior limitação é a quantidade de colunas do conjunto inicial.

Assim, este trabalho desenvolve uma técnica heurística que gera um conjunto reduzido de escalas viáveis para serem alocadas através do problema de cobertura de conjunto.

O desenvolvimento da técnica, a influência da variação dos parâmetros de geração e seleção e os resultados de alocação obtidos são então mostrados e analisados nos capítulos que se seguem.

## **ABSTRACT**

*The urban mass transport is a system composed of several parts, and each one of these can be considered as different problem to be solved*

*The scheduling of fleet and crew are problems of operational level that represent the largest costs of the system, in which prevails the costs with crew. All models of crew's scheduling just worry with the optimization methods properly said, belittling the generation of the initial group of scales, that is essential for the scheduling. Nowadays, the most used method is the set covering problem (SCP), whose largest limitation is the number of columns of the initial group of scales.*

*Thus, this work develops a heuristic that generates a reduced group of viable scales, that can be allocated through the set covering problem.*

*The development of the technique, the influence of the generation and selection parameters, and the results obtained on the scheduling, are shown and analyzed in the chapters that are proceeded.*

## ÍNDICE

LISTA DE QUADROS .....	v
LISTA DE FIGURAS .....	vi
CAPÍTULO I .....	1
1. INTRODUÇÃO .....	1
1.1 Considerações gerais .....	1
1.2 Objetivos do trabalho .....	6
1.3 Importância do trabalho .....	7
1.3.1 Importância para as empresas .....	7
1.3.2 Importância para a comunidade .....	8
1.3.3 Importância para a área de conhecimento .....	9
1.4 Limitações .....	9
1.5 Estrutura do trabalho .....	10
CAPÍTULO II .....	11
2. REVISÃO DA LITERATURA .....	11
2.1 Considerações iniciais .....	11
2.2 Alocação de frota .....	12
2.2.1 Alocação de frota com uma garagem – VSP ( <i>Vehicle Scheduling Problem</i> ) .....	15
2.2.2 Alocação de frota com restrição no comprimento da seqüência de viagens – VSPLPR ( <i>Length of Path Restriction</i> ) .....	16
2.2.3 Alocação de frota com múltiplos tipos de veículos – VSPMVT ( <i>Multiple Vehicle Types</i> ) .....	16
2.2.4 Alocação de frota com múltiplas garagens – VSPMD ( <i>Multiple Depots</i> ) .....	18
2.2.5 Modelos e algoritmos de alocação de frota .....	18
2.2.6 Considerações finais .....	24
2.3 Alocação de tripulação .....	25
2.3.1 Modelos e algoritmos de alocação de tripulação .....	31
2.3.2 Considerações finais .....	39

CAPÍTULO III .....	41
3. TÉCNICA DE SOLUÇÃO .....	41
3.1 Considerações iniciais .....	41
3.2 Alocação prévia da frota .....	41
3.3 Caracterização das regras trabalhistas .....	43
3.4 Construção das partições .....	45
3.4.1 Restrições .....	47
3.5 Geração das escalas .....	51
3.6 Seleção das escalas .....	54
CAPÍTULO IV .....	57
4. IMPLEMENTAÇÃO E RESULTADOS COMPUTACIONAIS .....	57
4.1 Considerações iniciais .....	57
4.2 Estrutura dos ensaios .....	57
4.3 Dados .....	58
4.4 Ensaio I – Parâmetros de geração e seleção das escalas .....	58
4.4.1 Parâmetros de geração das partições .....	59
4.4.2 Parâmetros de seleção das escalas .....	62
4.5 Ensaio II – Aplicação de cobertura de conjuntos .....	65
4.5.1 Mesmo conjunto de escalas .....	65
4.5.2 Conjuntos distintos de escalas .....	66
4.6 Ensaio III – Análise comparativa da técnica proposta .....	67
CAPÍTULO V .....	70
5. CONCLUSÕES E RECOMENDAÇÕES .....	70
5.1 Conclusões .....	70
5.2 Recomendações .....	71
REFERÊNCIAS BIBLIOGRAFIA .....	73
APÊNDICE: CÓDIGO DO ALGORITMO IMPLEMENTADO .....	78

## LISTA DE QUADROS

QUADRO 2.1: Quadro de horários parcial de uma empresa. ....	13
QUADRO 2.2 – Quadro de horários da Line College – Potomac Park .....	27
QUADRO 3.1 – Alocação prévia da frota.....	42
QUADRO 3.2 – Tabela de horários .....	50
TABELA 4.1 – Variação uniforme de li e ls .....	59
TABELA 4.2 – Variação diferenciada de li e ls .....	60
TABELA 4.3 – Variação do parâmetro de relaxamento dos limites .....	61
TABELA 4.4 – Variação diferenciada do parâmetro de relaxamento dos limites .....	61
TABELA 4.5 – Variação da quantidade de coberturas por viagem .....	64
TABELA 4.6 – Alocação conforme variação nos parâmetros de seleção. ....	66
TABELA 4.7 – Alocação de tripulação conforme os diferentes conjuntos de escalas filtradas.....	67

## LISTA DE FIGURAS

FIGURA 2.1 – Grafo representativo de uma alocação de frota .....	14
FIGURA 2.2 – Grafos representativos de um VSPLPR .....	23
FIGURA 2.3 – Relação entre esquemas de alocação de frota e condutores....	27
FIGURA 2.4 – Formação dos blocos, partições e escalas.....	30
FIGURA 2.5 – Histograma de trabalhadores requeridos por tempo .....	32
FIGURA 2.6 – Perfil da demanda e decomposição por tipo de escalas .....	37
FIGURA 3.1 – Formulação de Fluxo.....	50
FIGURA 4.5 – Distribuição das escalas geradas a partir da eficiência .....	63



# CAPÍTULO I

## 1. INTRODUÇÃO

### 1.1 Considerações gerais

O vertiginoso crescimento dos centros urbanos é um fato global e, até o presente momento, irreversível. Com isto, aumentaram os deslocamentos realizados pelas pessoas, seja entre cidades ou nos próprios centros urbanos, gerando então uma maior necessidade de meios de transporte. Este rápido processo é um fator complicador para o planejamento urbano, uma vez que o estado possui pouca flexibilidade e recursos para tratar os problemas na velocidade e presteza necessárias. Aliado a isto, a indústria automobilística coloca todo ano no mercado centenas de milhares de automóveis, saturando ainda mais o sistema viário e exigindo investimentos cada vez maiores para adequá-lo à demanda.

Dentro deste contexto está o transporte coletivo de passageiros, sendo que das várias modalidades existentes, o sistema mais utilizado é o transporte rodoviário, normalmente operado por ônibus. Os sistemas de metrô e trens estão presentes basicamente nas grandes metrópoles, onde existe maior volume de recursos financeiros e demandas que exigem um modal mais apropriado.

O transporte rodoviário de passageiros é considerado atividade de caráter essencial. Possui um universo bastante amplo, no qual interagem a

comunidade, os usuários, a iniciativa privada e o estado. A comunidade está diretamente relacionada, pois os benefícios e transtornos causados por este afetam a vida de todos, até mesmo daqueles que não utilizam o sistema coletivo. Os usuários representam a necessidade de transporte, a demanda requerida para o sistema. A iniciativa privada, na maioria dos casos, é o meio operacional para atender a demanda. O estado, ou poder público, é o detentor do serviço, podendo delegá-lo ou até mesmo executá-lo. A atual legislação federal determina aos estados e municípios a obrigatoriedade de regulamentação do setor. Assim, a variedade de cenários para o problema de transporte rodoviário de passageiros é enorme.

A formulação deste problema pode ser dividida em dois níveis de responsabilidades: o do poder público e o das empresas operadoras. Assim, no âmbito do poder público, pode-se elencar alguns dos subproblemas mais comuns:

- levantamento e acompanhamento da demanda de passageiros (origens, destinos e quantidades);
- configuração do sistema (determinação das rotas ou linhas, nível de atendimento, terminais de transbordo, etc);
- operacionalização do sistema (quadro de horários, frota requerida, valores tarifários e fiscalização do serviço).

As empresas operadoras por sua vez, além de problemas estratégicos e táticos (renovação de frota, localização das garagens, etc), estão sujeitas a problemas operacionais como:

- roteamento;
- alocação da frota;
- alocação das tripulações (motoristas e cobradores);
- sistema de manutenção da frota.

A questão do roteamento é pouco comum, uma vez que os sistemas e suas respectivas linhas já são uma realidade em operação, além do que o fator político é o critério mais relevante neste tipo de problema. Assim, a alocação de frota e tripulação são os problemas de maior importância e complexidade na operação do sistema. Consiste em executar todas as viagens estabelecidas utilizando uma quantidade de veículos e condutores da maneira mais eficiente possível. O formato final desta distribuição é uma escala de trabalho que descreve a operação diária da empresa.

Dentre as várias modalidades de transporte rodoviário de passageiros, o urbano é a que apresenta as menores extensões de linha e as maiores frequências de viagens. Com isto, a otimização da frota e pessoal tem uma relevância maior nesta modalidade, pois como se trabalha numa escala maior, qualquer pequeno ganho pode representar uma significativa economia.

A alocação de frota e tripulação no transporte rodoviário urbano de passageiros depende de alguns parâmetros e definições, como:

- Linha — trajeto a ser percorrido a partir de um ponto inicial até um ponto final com paradas intermediárias. Uma viagem representa um ciclo neste trajeto, sendo caracterizada por um local e horário de início e um local e horário de término;
- Quadro de horários — relação das viagens das linhas com seus respectivos horários de partida e chegada;
- Locais de referência — localização dos pontos de início e fim das viagens (terminais, garagens, etc) e pontos auxiliares, como os de troca de tripulação;
- Frota disponível — quantidades, modelos, tipos, capacidade de carga (passageiros em pé e sentados) e autonomia para reabastecimento;
- Pessoal — quantidades por função (motoristas e cobradores), disponibilidade e legislação trabalhista vigente (jornada de trabalho, horas extras e descansos);
- Custos — fixos e variáveis (pessoal, capital, etc);
- Paradigmas das empresas — cultura ou particularidade de cada empresa, como a quantidade máxima de motorista por carro,

utilização maior ou menor de horas extras e sistemas de manutenção e estocagem dos ônibus;

- Políticas e legislação do poder concedente — de acordo com a diversidade de cenários existentes, pode ocorrer uma infinidade de restrições extras, como a obrigatoriedade do uso de veículos especiais para determinadas linhas ou horários, como por exemplo, veículos com ar-condicionado ou com acessórios para deficientes;

Dentro deste universo é que ocorre o desenvolvimento das escalas de trabalho. Atualmente, a maioria das empresas de transporte urbano de médio e pequeno porte realiza a alocação de frota e tripulação manualmente através de funcionários experientes, denominados *schedulers*, ou práticos. Apesar de parecer impossível, a experiência e os métodos destes práticos garantem a operação destas empresas, porém não necessariamente com o menor custo. Um dos motivos facilitadores é que em geral as linhas já existem há bastante tempo, e as características da escala foram moldadas ao longo dos anos. Modificações para adequação da alocação, como no caso da inclusão de novas viagens, são solucionadas pontualmente através da utilização de carros extras ou extensões de jornada, não sendo realizada assim uma reotimização do problema.

Com isto, observa-se que uma ferramenta computacional bem estruturada para alocação de frota e tripulação, que contemple esta diversidade de restrições e critérios, pode trazer grandes benefícios para os diversos envolvidos no sistema, possibilitando assim a redução de custos como um todo.

## 1.2 Objetivos do trabalho

Os modelos utilizados na formulação do problema de alocação de frota e tripulação possuem como foco principal os métodos de otimização. A comparação do desempenho destes algoritmos normalmente se dá através da realização de testes com dados provenientes de fonte comum, como os disponibilizados em bibliotecas virtuais (por exemplo, *OR Library*, vide [BEA90]). Na prática, porém, os dados de entrada para a execução da otimização não estão organizados no formato que alguns algoritmos requerem.

Nesta particularidade é que está inserido este trabalho, que visa transformar situações reais em dados de entrada adequados para o funcionamento dos algoritmos de otimização, mais especificamente os modelos baseados no problema de cobertura de conjunto (*set covering*), dissertado no capítulo seguinte.

Portanto, o objetivo principal deste trabalho é gerar um conjunto reduzido de escalas viáveis que possibilite a alocação de condutores através do problema de cobertura de conjuntos.

Outro objetivo é descrever a influência dos parâmetros de geração, utilizados na técnica, no resultado final da otimização, comparando-os ainda com escalas reais desenvolvidas por métodos diversos.

### 1.3 Importância do trabalho

#### 1.3.1 Importância para as empresas

O custo com pessoal e frota, nas empresas de transporte rodoviário urbano de passageiros, normalmente ultrapassam 50% do custo total, que normalmente já é bastante elevado. Com isto, as alocações eficientes representam significativa redução de custos operacionais, além de reduzirem ainda o passivo trabalhista, uma vez que a utilização adequada da mão-de-obra previne ações indenizatórias.

As freqüentes mudanças no quadro de horários, o acréscimo ou cancelamento de linhas e a modificação das normas trabalhistas são exemplo de fatores que podem ser assimilados de forma mais eficientes, uma vez que haveria a possibilidade de reotimizar as escalas de um modo geral e não mais de forma pontual como acontece.

Além disto, um processo automatizado de geração de escalas provoca mudanças significativas nos procedimentos operacionais das empresas. O principal exemplo é o do *scheduler* manual, ou prático, que deixa de ser o responsável pela confecção das escalas, eximindo as empresas desta inconveniente dependência. A experiência destes práticos continua importante para o processo, agora na formulação do problema, proporcionando um conjunto de restrições e conceitos mais próximos da realidade de cada empresa. Assim, democratiza-se um conhecimento, permitindo a geração de escalas mais eficientes, o que reflete em ganho de produtividade.

### 1.3.2 Importância para a comunidade

O transporte coletivo rodoviário movimentava nos centros urbanos brasileiros aproximadamente 50 milhões de passageiros por dia, conforme dados da ANTP <sup>1</sup>. Assim, as melhorias neste sistema proporcionam benefícios para uma razoável parcela da população. A geração de escalas automatizadas pode ser considerada como uma destas melhorias, pois:

- garante respostas rápidas para as mudanças no perfil de demanda, uma vez que permite aos gestores do sistema utilizar soluções mais eficientes e complexas na adequação da oferta;
- pode proporcionar uma redução no valor da tarifa cobrada dos usuários, uma vez que reduz os custos operacionais das empresas (depende da metodologia utilizada no cálculo tarifário);
- permite análises rápidas e concretas do impacto que mudanças advindas de negociações sindicais podem causar (jornada de trabalho, horas extras e remuneração básica), o que também influi na tarifa. Com esta agilidade na análise pode-se evitar possíveis greves, que representam o caos para as populações urbanas.

---

<sup>1</sup> Agência Nacional dos Transportes Públicos ([www.antp.org.br](http://www.antp.org.br))



### **1.3.3 Importância para a área de conhecimento**

A revisão bibliográfica, dissertada no capítulo seguinte, mostra que nenhum dos autores descreve claramente em seus trabalhos como obtém o conjunto inicial de dados utilizado nos modelos de otimização. Este trabalho permite traduzir situações reais para a linguagem dos algoritmos de cobertura de conjunto, gerando um conjunto inicial de combinações possíveis. Com isto, possibilita que os ensaios para avaliação da eficiência de algoritmos de alocação de frota e tripulação sejam desenvolvidos com dados de situações reais e distintas, permitindo assim análises mais fundamentadas.

### **1.4 Limitações**

A prévia alocação de veículos pode impossibilitar que uma solução mais eficiente possa ser encontrada, porém para este trabalho é essencial. Assim entende-se que é uma limitação deste a necessidade de uma alocação de frota predeterminada.

Pode-se citar ainda o fato de este trabalho abranger apenas a modalidade de transporte rodoviário urbano de passageiros, não contemplado então o interurbano, e outros.

Outra limitação é o vínculo com o modelo de otimização utilizando cobertura de conjuntos, não apresentando, portanto interface de dados para outras técnicas de solução.

## 1.5 Estrutura do trabalho

O trabalho é composto por cinco capítulos. Neste primeiro é apresentando o contexto em que se enquadra o trabalho, discriminando seus os objetivos, importâncias e limitações.

O segundo capítulo apresenta a revisão bibliográfica, na qual se buscou subsídio para o desenvolvimento deste trabalho, bem como sua fundamentação teórica.

O terceiro capítulo mostra as técnicas de solução utilizada no algoritmo, seus principais parâmetros e lógica.

O capítulo quarto apresenta os resultados da aplicação do algoritmo em casos reais. Demonstra os efeitos na variação dos parâmetros de entrada, bem como as análises obtidas com os valores finais da otimização.

Já o capítulo quinto é reservado para as conclusões e recomendações do trabalho.

## CAPÍTULO II

### 2. REVISÃO DA LITERATURA

#### 2.1 Considerações iniciais

A alocação de frota e tripulação tem primórdio incerto, uma vez que qualquer remoto conjunto formado por veículos (automotores ou de tração animal) e condutores, com uma tarefa a realizar, representa um exemplo deste problema. A evolução dos meios de transporte, das regras trabalhistas e das restrições impostas pelas cidades fez com que a alocação de frota e tripulação crescesse em importância e complexidade.

Considerando, por exemplo, uma cidade de porte pequeno, onde são realizadas diariamente 1.000 viagens, com uma frota de 50 veículos e 100 condutores. Se cada veículo executa somente 20 viagens, o primeiro pode ser alocado em  $C_{1000}^{20}$  alternativas diferentes, o segundo em  $C_{980}^{20}$ , até o último carro ter só uma opção de alocação. Desta forma, os números possíveis de combinações de alocação de viagens aos veículos seriam de  $C_{1000}^{20} \times C_{980}^{20} \times \dots \times C_{20}^{20} = 1,99 \times 10^{1648}$ . Considerando agora que a escala de um veículo é dividida em duas partes, sendo que cada uma delas é executada por um condutor (padrão mais utilizado pelas empresas), o primeiro condutor pode ser alocado em 100 veículos diferentes, o segundo, 99 e assim por diante. Portanto, o número de combinações para a alocação de condutores é de  $100! = 9,33 \times 10^{157}$ . Considerando então as alocações de veículos e condutores

chega-se a  $1,86 \times 10^{1806}$  alternativas possíveis de solução do problema. Apesar de muitas destas serem inviáveis, a quantidade total de combinações é incrivelmente grande.

Portanto, percebe-se que a utilização de técnicas de otimização combinatorial eficientes é essencial na busca do melhor esquema operacional.

Deve-se observar, no entanto, que o problema em questão é formado por dois problemas distintos e que possuem grande interatividade, a alocação de frota e a alocação de tripulação. Esta interação deve-se ao fato de que a solução de um dos problemas fornece os parâmetros de resolução do outro.

A seguir é apresentado o problema de alocação de frota, depois o de alocação de tripulação, as principais características, variações e os modelos e algoritmos de resolução mais utilizados. Será mostrado também um modelo de resolução simultânea dos dois problemas.

## **2.2 Alocação de frota**

O VSP (*Vehicle Scheduling Problem*), ou problema de alocação de veículos, consiste em realizar um conjunto de tarefas (viagens), minimizando uma função objetivo (custo de capital e operacional), com um número limitado de veículos associados a um ou mais depósitos (garagens). As viagens são caracterizadas por um local e um horário de saída e um local e um horário de chegada. O conjunto de todas as viagens a serem realizadas por uma empresa é chamado de quadro de horários, conforme exemplo mostrado no quadro 2.1.

QUADRO 2.1: Quadro de horários parcial de uma empresa. Fonte: Empresa Estrela Transportes Coletivos Ltda.

Viagem	Horário		Local	
	Saída	Chegada	Saída	Chegada
1	5:15	5:40	Capoeiras	Centro
2	5:25	5:50	São João	Centro
3	5:30	5:55	Abraão	Centro
4	5:30	5:55	Capoeiras	Centro
5	5:50	6:35	Centro	Centro
6	5:55	6:45	Centro	Centro
7	6:00	7:00	Centro	Centro
8	6:27	7:07	Centro	Centro
9	6:35	7:20	Centro	Centro
10	7:00	7:45	Centro	Centro
11	7:08	8:08	Centro	Centro
12	7:20	8:05	Centro	Centro

Cada viagem deverá ser executada por um único veículo, definindo desta forma, uma seqüência viável para cada um deles. A alocação, portanto, é caracterizada pela partida do depósito, seqüência de viagens, e retorno ao depósito. Pode ser representada por um gráfico direcional acíclico, em que as viagens são os nós numerados, com os respectivos horários de partida e chegada. Os arcos de ligação entre os nós indicam os possíveis sequenciamentos, conforme mostra a figura 2.1. A viagem  $j$  pode ser ligada à viagem  $i$ , se o horário de saída de  $j$  ( $HS_j$ ) for maior que o horário de chegada de  $i$  ( $HC_i$ ). Considerando ainda o tempo  $T_{deslocij}$  referente ao deslocamento improdutivo (viagem morta) necessário para ir do local de saída de  $j$  ( $LS_j$ ) ao local de chegada de  $i$  ( $LC_i$ ), a viabilidade da seqüência resume-se a:  $HS_j \geq HC_i + T_{deslocij}$ .

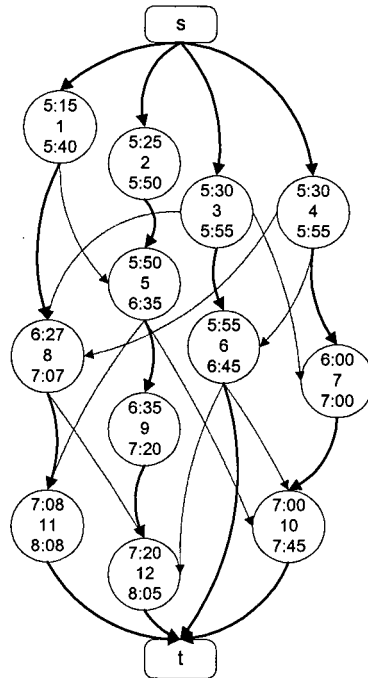


FIGURA 2.1 — Grafo representativo de uma alocação de frota. Fonte: M. Tramontim (vide [TRA01])

Respeitando esta regra de validação das ligações e com os dados do quadro 2.1, chegou-se então ao grafo da figura 2.1. As alocações viáveis dos carros são representadas pelos caminhos de saída e retorno à garagem, sendo que todos os nós devem ser cobertos. Assim, tem-se 4 carros para realizar um conjunto de 12 viagens.

O problema de alocação de frota apresenta algumas formas diferenciadas de formulação conforme determinadas particularidades. Bodin et al. (vide [BOD83]) sugerem três restrições para nortear estas formulações, que são:

- seqüências com limite de comprimento devido à autonomia de combustível dos veículos;

- veículos de diferentes tipos que podem atender apenas a um determinado conjunto de viagens;
- mais de uma garagem (depósitos) para início e fim das viagens.

Estas restrições permitem classificar o problema de alocação de frota em quatro tipos básicos, conforme descritos a seguir.

### **2.2.1 Alocação de frota com uma garagem – VSP (*Vehicle Scheduling Problem*)**

O VSP é o problema de alocação com apenas uma garagem, onde os nós  $s$  e  $t$  representam o mesmo depósito. A função objetivo visa minimizar o custo total da alocação, que significa escolher um determinado número de caminhos com o menor custo e que cubra todas as viagens. Como cada caminho está associado a um veículo, a minimização do número de caminhos reduz o custo de capital investido em frota, bastando para isto acrescentar um custo  $K$  aos arcos  $(s,i)$  de saída da garagem e  $(j,t)$  de retorno, o que penaliza a utilização de mais veículos. Já a escolha dos caminhos mais econômicos representa uma redução do custo operacional, sendo que os veículos são do mesmo tipo, e os custos das viagens independem da alocação, conforme Mayerle (vide [MAY96]).

### **2.2.2 Alocação de frota com restrição no comprimento da seqüência de viagens – VSPLPR (*Lenght of Path Restriction*)**

A restrição no comprimento da seqüência de viagens está relacionada com a quilometragem ou tempo despendido pelo veículo durante a operação.

A restrição de quilometragem percorrida refere-se à autonomia do veículo, conforme o consumo de combustível e volume abastecido. Aspectos relacionados com a manutenção dos veículos também devem ser considerados, como os intervalos para as intervenções mecânicas.

O tempo de duração da seqüência pode ainda ser uma restrição quando se relaciona o VSP com uma alocação de tripulação sujeita às regras trabalhistas. Neste caso restringe-se a alocação da frota a um tempo máximo correspondente á jornada de trabalho de um ou mais condutores.

### **2.2.3 Alocação de frota com múltiplos tipos de veículos – VSPMVT (*Multiple Vehicle Types*)**

As particularidades da operação das empresas de transporte normalmente exigem veículos específicos para algumas tarefas. No transporte urbano de passageiros, por exemplo, isto pode ser verificado pelas diferentes linhas existentes. As linhas longas (acima de 30 km), com poucos pontos de parada, apresentam um índice de renovação de passageiros<sup>1</sup> pequeno (abaixo de 20%). Com isto existe a necessidade de um veículo com poucas portas e com

---

<sup>1</sup> Índice de renovação de passageiros é a quantidade de passageiros que entram e saem do ônibus ao longo de uma viagem



o maior número possível de assentos confortáveis. As linhas curtas, por sua vez, apresentam um índice de renovação de passageiro mais elevado, necessitando assim de veículos com um maior número de portas e com uma área para passageiros em pé também maior. Neste caso os poucos bancos existentes não precisam ser tão confortáveis, sendo na maioria dos casos de fibra, pois os passageiros permanecem pouco tempo dentro do veículo. Empresas que operam conjuntamente as linhas supramencionadas possuem frotas com grupos de veículos específicos para atendê-las. Veículos das linhas longas normalmente podem atender a algumas viagens das linhas curtas, desde que possuam capacidade adequada. Veículos das linhas curtas não podem atender às linhas longas, se possuírem bancos de fibra, uma vez que os passageiros permanecerão mais tempo dentro do veículo.

Este exemplo visa demonstrar um possível tipo de restrição encontrado no VSP. Pode-se citar ainda:

- veículos especiais para deficientes (com elevador) que realizam viagens específicas;
- veículos com diferentes capacidades (articulado, biarticulado, etc);
- veículos com restrição legal de operação (sistema municipal e metropolitano);
- linhas especiais que necessitam de carros específicos, como: terrenos de relevo acidentado; vias estreitas e com pisos perigosos; entre outros.

#### **2.2.4 Alocação de frota com múltiplas garagens – VSPMD (*Multiple Depots*)**

A existência de um ou mais depósitos (garagens) nas empresas de transporte depende de uma decisão estratégica que engloba um universo de variáveis distinto. Estes depósitos têm como finalidade a estocagem e reabastecimento dos veículos, podendo ser então pontos de início e fim de alocações.

A existência de mais de uma garagem permite que os veículos fiquem estocados em pontos diferentes, podendo assim iniciar ou terminar as viagens em outras garagens. Com isto a restrição da quilometragem morta possui mais de uma opção de otimização, porém aumentando o grau de complexidade do problema.

#### **2.2.5 Modelos e algoritmos de alocação de frota**

A forma mais simples do problema de alocação de frota é o VSP, uma vez que apresenta resolução polinomial. Pode ser formulado como um problema de transporte, de atribuição linear, de *matching* ou de fluxo de mínimo custo.

Danzig e Fulkerson (vide [DAN54]) descreveram o seguinte modelo de fluxo em rede para o VSP:

$$\text{Min } \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.1)$$

s.a.:

$$\sum_{i:(i,j) \in A} x_{ij} - \sum_{i:(i,j) \in A} x_{ji} = 0 \quad \forall j \in N - \{s, t\} \quad (2.2)$$

$$\sum_{i:(i,j) \in A} x_{ij} = 1 \quad \forall j \in N - \{s, t\} \quad (2.3)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \quad (2.4)$$

Onde:

- $N$  é o conjunto de nós (viagens), incluindo o de início  $s$  e o de fim  $t$ ;
- $A$  é o conjuntos de arcos;
- $c_{ij}$  é o custo associado a cada arco  $(i,j) \in A$  que é uma função do custo operacional e do custo de capital investido no veículo;
- $x_{ij} = 1$ , se o veículo executa a viagem  $j$  após a viagem  $i$  e  $x_{ij} = 0$  caso contrário;
- A restrição (2.2) representa a conservação do fluxo (um arco chega e outro sai do nó);
- A restrição (2.3) garante a atribuição de cada viagem a exatamente um veículo (limite da capacidade do nó);

- A restrição (2.4) refere-se aos limites da capacidade dos arcos.

Considerando que os custos dos arcos que saem de  $s$  sejam iguais a 1 (Formulação de Dilworth), a solução ótima para este modelo pode ser obtida através do algoritmo *out-of-kilter* de Ford e Fulkerson (vide [FOR62]).

Bodin et al. (vide [BOD83]) formularam o VSP como um problema de fluxo de mínimo custo, onde é acrescido ao custo  $c_{ij}$  dos arcos aspectos relacionados ao deslocamento improdutivo, como por exemplo:

- Custo operacional associado ao deslocamento do local de chegada da viagem  $i$  ( $LC_i$ ) ao local de saída da viagem  $j$  ( $LS_j$ );
- Custo operacional associado ao deslocamento do local de chegada da viagem  $i$  ( $LC_i$ ) à garagem;
- Custo de capital mais custo operacional associados ao deslocamento da garagem ao local de saída da viagem  $i$  ( $LS_i$ ).

Mayerle (vide [MAY96]) resolveu o problema de alocação de frota através da utilização do algoritmo húngaro numa série de problemas de designação. O grafo com as viagens a serem realizadas é dividido em camadas e, para cada uma destas, um problema de designação é solucionado. Propõe ainda um melhoramento da alocação utilizando uma recombinação de mínimo custo das viagens previamente seqüenciadas.

Por meio da teoria dos conjuntos difusos, abordou de forma pioneira:

- O grau de incerteza em relação ao tempo de viagem;
- A tolerância em relação aos horários de saída;
- O grau de adequação da frota e das linhas.

Mais recentemente Freling et al. (vide [FRE99]) formularam o VSP como um problema de *quasi-assignment*, que é uma versão modificada do algoritmo húngaro para o problema de atribuição linear. A solução ótima é obtida através de um algoritmo leilão acrescido de técnicas de redução no número de arcos. O algoritmo é do tipo primal-dual utilizado em problemas de fluxo em rede. As técnicas de redução são: aproximação de duas fases e aproximação de núcleo centrado (*core oriented*). Foi realizado ainda um comparativo em relação aos algoritmos mais utilizados para resolução do VSP (húngaro, fluxo de mínimo custo e sucessivo melhor caminho). Obtiveram com seu modelo um ganho significativo em relação aos analisados.

Baita et al.(vide [BAI00]), utilizando um sistema de suporte a decisão (DSS — *Decision Support System*), fizeram também uma comparação entre métodos de resolução. Com dados reais, compararam o modelo de atribuição (algoritmo húngaro), programação lógica e algoritmos genéticos com as soluções utilizadas na empresa. O método de atribuição, que é polinomial, e a programação lógica, de complexidade exponencial, foram os que apresentaram os melhores resultados e passaram a ser utilizados pela empresa. Já o algoritmo genético não proporcionou resultados equivalentes.

Tramontim (vide [TRA01]) utilizou a metaheurística de Busca Local Dirigida (BLD) na resolução do VSP. Esta técnica foi proposta por Voudooris (vide [VOU97]) e consiste na variação da função custo conforme a inclusão de penalidades, visando com isto orientar a busca local. Pode ser utilizada em vários tipos de problema de otimização combinatorial. Na resolução do VSP, a técnica é dividida em duas etapas: na primeira encontra-se uma solução inicial através de atribuição (vide [MAY96]) ou programação concorrente e na segunda é aplicada a BLD para melhorar a solução.

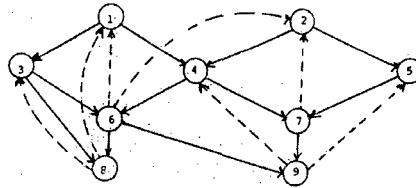
Os problemas VSPMVT, VSPLPR e VSPMD são *NP*-difíceis (vide [BAL80] e [LEN81]), sendo suas resoluções obtidas através de técnicas de redução e algoritmos heurísticos. Neste sentido Bodin et ali. (vide [BOD83]) apresentam três métodos distintos:

- programação concorrente;
- abordagem de duas fases
- trocas heurísticas.

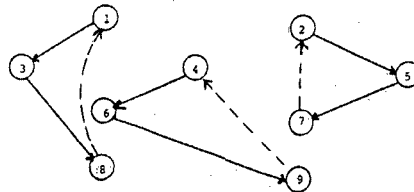
Para o VSPLPR, Bodin et al. utilizam o mesmo grafo do VSP, porém sem os nós referentes aos depósitos e com a inclusão de arcos de retorno. A inserção destes arcos só é realizada se a seqüência respeita a restrição de comprimento. É utilizada ainda uma restrição que garante que cada seqüência possua apenas um arco de retorno. A solução é composta por um conjunto de circuitos que cobrem todas as viagens (nós) do grafo, no menor custo. A figura

2.2 mostra um exemplo com soluções viáveis e inviáveis, nas quais os arcos de retorno são os tracejados.

- grafo original (sem as garagens)



- solução viável (cada circuito com um arco de retorno)



- solução inviável (o segundo circuito tem dois arcos de retorno)

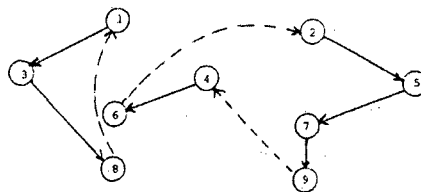


FIGURA 2.2 — Grafos representativos de um VSPLPR. Fonte: L. Bodin et al. (vide [BOD83])

Ferland e Michelon (vide [FER98]) propõem para o VSPMVT três métodos heurísticos: aproximação discreta, procedimento de duas fases e seqüência de problemas de transporte. Apresentam ainda dois métodos exatos baseados no *branch-and-bound*, sendo que um utiliza uma técnica de geração de colunas e *set-partitioning* e o outro, relaxamento nas restrições de janela de tempo.

Os problemas de VSPMVT e VSPMD, de acordo com Mayerle (vide [MAY96]), possuem estrutura mais apropriada para a utilização de modelos de fluxo em rede com múltiplas mercadorias.

### 2.2.6 Considerações finais

A alocação de frota para transporte urbano de passageiros possui vários aspectos que caracterizam o problema.

O custo médio de um ônibus é de aproximadamente sessenta mil dólares, sendo que os modelos mais utilizados variam na faixa de quarenta mil e os biarticulados podem chegar a duzentos e cinquenta mil dólares. Tais valores, apesar de significativos, não representam a maior fatia dos custos totais das empresas, sendo que este posto é ocupado pelo custo com pessoal. No transporte aéreo, por exemplo, o custo associado à frota é muito superior, onde as aeronaves são avaliadas em milhões de dólares e necessitam manutenções mais dispendiosas.

Outro fato relevante a ser considerado é que a minimização de frota através de uma alocação eficiente não é prioridade para os gestores privados do sistema, uma vez que a aquisição de frota está inserida num contexto político, econômico e cultural mais amplo. Assim, busca-se alocar a frota existente, menosprezando possíveis otimizações.

Por outro lado, têm-se os fatores que priorizam a alocação de frota. Como exemplo pode-se citar os paradigmas de dimensionamento dos sistemas de transporte, como os usados na confecção do quadro de horários. Neste processo, uma alocação de frota é previamente realizada, visando basicamente à redução do número de veículos de cada linha. Diversos modelos, técnicas e ferramentas computacionais são utilizados para este dimensionamento, porém



não são objetos deste trabalho. A relevância do exemplo está na evidência da tendência de alocação inicial de frota antes da alocação da tripulação, ou desconsiderando-a. Tal fato é verificado em praticamente todos os *papers* pesquisados, mostrando assim que esta tendência tem fundamentação pelos seguintes motivos:

A alocação prévia da frota estabelece valores para os horários de saída e chegada (*HC* e *HS*), simplificando então a rede resultante;

O número de padrões em potencial é reduzido, diminuindo assim complexidade do problema original.

Vale ressaltar, porém, que esta alocação inicial pode eliminar algumas soluções interessantes, inclusive a ótima.

### **2.3 Alocação de tripulação**

O termo tripulação decorre do fato de que a maioria dos ônibus trabalham com um motorista (condutor) e um cobrador, que é o funcionário responsável pela cobrança das passagens. Assim, quando aparecer o termo “tripulação” ou “condutores”, entenda-se condutores e cobradores, pois suas restrições trabalhistas são as mesmas.

O problema de alocação de tripulação, ou CSP (*Crew Scheduling Problem*), consiste em realizar um conjunto de tarefas (viagens), minimizando uma função objetivo (custo operacional), com um número limitado de condutores,

associados a um ou mais pontos de parada (pontos para troca de condutores).

Basicamente é o mesmo problema do VSP, sendo que:

- Ao invés dos veículos, têm-se condutores;
- A função custo não possui o fator capital, pois não se pode associar um valor a ser depreciado a um condutor;
- Os "depósitos" passam a ser pontos de possíveis trocas de condutores, sendo que nestes pode ocorrer o descanso obrigatório da tripulação.

A estreita interação entre o VSP e o CSP foi exemplificada por Bodin et al. [vide BOD83], onde apresentam um modelo genérico para formulação do CSP, utilizando para isto uma pré-alocação de frota. A figura 2.3 mostra este estreito relacionamento. Inicialmente foi atribuído para três veículos (V1, V2 e V3) uma seqüência de viagens, divididas em partes conforme os pontos de troca dos condutores. As partes são então combinadas de maneira a formar as escalas de trabalho dos condutores (C1, C2, C3 e C4), sendo que neste exemplo as linhas tracejadas referem-se aos intervalos de descanso, ou tempo improdutivo. A viabilidade da combinação destas partes depende de fatores como:

- horário e local de início e fim das partes;
- regras trabalhistas que regem a jornada de trabalho da tripulação, como: duração da jornada, tempo de descanso, horas extras, etc.

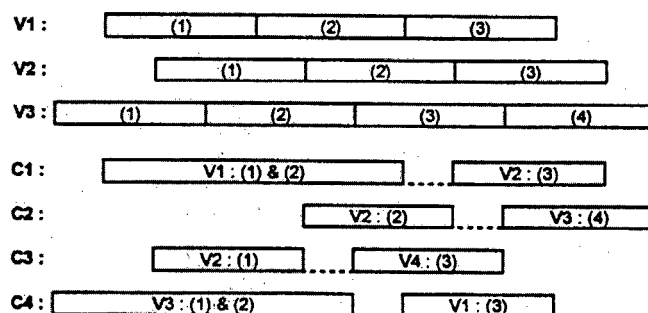


FIGURA 2.3 — Relação entre esquemas de alocação de frota e condutores.  
 Fonte: L. Bodin et al. (vide [BOD83])

Como no VSP, o dado de entrada para o CSP também é o quadro de horários com as respectivas viagens, uma vez que a tarefa de executá-las não muda. Porém, a informação sobre os pontos de possível troca de condutores e o horário de passagem por estes pontos ao longo de uma linha é primordial. O quadro 2.2 exemplifica este detalhamento da informação, onde os pontos de troca dos condutores são assinalados por (\*).

QUADRO 2.2 – Quadro de horários da Line College – Potomac Park. Fonte: L. Bodin et al. (vide [BOD83])

College Park (I)*	Mount Rainer (II)*	North Capitol e New York Ave. (III)	Potomac Park (IV)*
:	:	:	:
09:00	09:20	09:55	10:15
09:15	09:35	10:10	10:30
:	:	:	:
10:20	10:00	09:25	09:05
10:35	10:15	09:40	09:20

Alguns conceitos e nomenclaturas devem ser entendidos antes da descrição do modelo proposto. São eles:

- *Trip* — é a viagem, propriamente dita, realizada pelo mesmo carro, pois caso contrário haverá a penalidade do transbordo para os passageiros.
- *D-trip (driver trip)* — é uma porção da viagem que deve ser executada pelo mesmo condutor e veículo. No caso mais simples, a *d-trip* é formada pela partição de viagens nos pontos de parada. Pode ainda representar o trajeto garagem — ponto inicial/final.

O conjunto de *trips*, ou viagens, é a base para a alocação de veículos, desde que ao final de cada viagem a decisão tomada respeite o movimento do próximo veículo. Já o conjunto de *d-trips* é a base para a alocação de condutores, desde que ao final de cada *d-trip* a decisão tomada respeite o movimento dos próximos condutores. Considerando os dados do quadro 2.2, pode-se ter as seguintes *trips* e *d-trips*:

Trips: [(09:00,I),(10:15,IV)]; [(09:15,I),(10:30,IV)];

[(09:05,IV),(09:20,I)]; [(09:20,IV),(10:35,I)].

D-trips: [(09:00,I),(09:20,II)]; [(09:20,II),(10:15,IV)]; [(09:15,I),(09:35,II)];

[(09:35,II),(10:30,IV)]; [(09:05,IV),(10:00,II)]; [(10:00,II),(10:20,I)];

[(09:20,IV),(10:15,II)]; [(10:15,II),(10:35,I)].

- Partição — seqüência de *d-trips*. Segmento da alocação do condutor, onde este permanece no mesmo veículo;

- Escala, ou Corrida — é a combinação de partições que forma a escala diária de trabalho dos condutores. É caracterizada pelos locais e horários de:
  - início de jornada;
  - início do intervalo de descanso;
  - fim do intervalo de descanso;
  - fim da jornada.

Algumas combinações de partições dependem do deslocamento do condutor desde o ponto final de uma partição até o ponto inicial de outra. A viabilidade da *corrida*, por sua vez, depende da existência de intervalos obrigatórios de descanso, cujas características são firmadas nos acordos trabalhistas (duração do intervalo, quantidade, etc). Intervalos maiores normalmente são mais interessantes para as empresas que possuem demanda de passageiros muito concentrada.

- *Blocos* — consiste da saída do veículo da garagem, uma seqüência de *trips* (uma ou mais *d-trips*), e retorno à garagem. Um veículo pode executar um ou mais blocos numa escala diária.

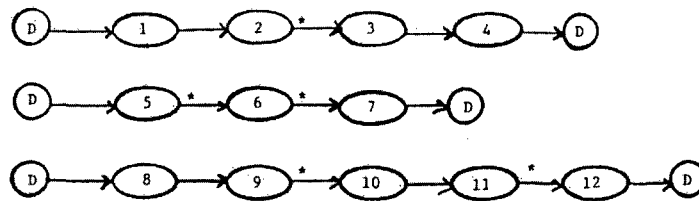
A solução do CSP, segundo Bodin et al. [vide BOD83] é mostrada na figura 2.4 e consiste basicamente em:

- formação dos blocos e sua alocação destes aos veículos (VSP);

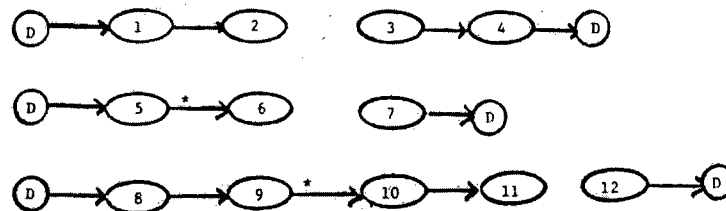
- fragmentação dos blocos em partições;
- combinação das partições de maneira a formar as escalas, e alocação destas aos condutores.

O conjunto das escalas caracteriza uma escala de trabalho diária para uma empresa. Normalmente são utilizadas três escalas diferentes: uma serve para os dias úteis, outra para os sábados e outra para domingos e feriados.

- Blocos



- Partições



- Escalas

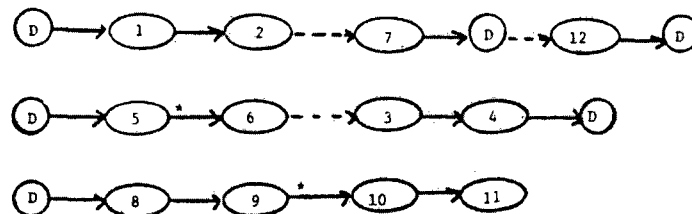


FIGURA 2.4 – Formação dos blocos, partições e escalas. Fonte: L. Bodin et al. (vide [BOD83])

A distribuição destas escalas aos condutores caracteriza um *crew rostering problem*, podendo seguir ou não critérios específicos. Na Europa, por exemplo,

existe o conceito de que todos os condutores têm que ter carga de trabalho e remuneração equivalentes. Nos Estados Unidos esta distribuição segue parâmetros internos das empresas, buscando sempre manter o funcionário na mesma rota, a fim de possibilitar um maior desempenho dele e do veículo. No Brasil esta distribuição normalmente não segue critérios fixos, sendo que na maioria dos casos esta tarefa fica a cargo da fiscalização (operação) das empresas.

### **2.3.1 Modelos e algoritmos de alocação de tripulação**

#### **2.3.1.1 Run cutting**

Os *schedullers* manuais, ou práticos, utilizam um conjunto de heurísticas denominado *run cutting* para confeccionar as escalas que hoje movimentam a grande maioria dos sistemas urbanos de transporte de passageiros. Este método consiste em "cortar" partições de uma alocação prévia de frota e executar um método de combinação visando à formação de escalas (escalas). Na década de 70, os primeiros sistemas computacionais a serem desenvolvidos utilizaram este método, como o RUCUS, que se apresentou pouco flexível e de difícil manuseio. Hildyard e Wallis [vide [HIL81]] apresentaram na década de 80 uma versão refinada do RUCUS, que supera estes problemas, porém sendo necessário ainda um grande esforço de pesquisa para implementação, devido às particularidades de cada empresa ou sistema.

### 2.3.1.2 Local fixo

O modelo de alocação de trabalhadores em um local fixo foi proposto inicialmente por Bennet e Potts (vide [BEN68]). Caracteriza-se pela concentração da mão-de-obra em um local fixo e por um histograma que descreve a necessidade de trabalhadores em cada período, conforme a figura 2.5. Desconsidera com isto o tempo gasto no deslocamento entre o local de chegada da viagem  $i$  e o de saída da viagem  $j$ . A inclusão de restrições extras pode limitar a quantidade de escalas com determinadas características, como, por exemplo, as que possuem horas extras.

Este modelo é utilizado para análises rápidas do impacto causado por mudanças nas regras trabalhistas ou no perfil da demanda (vide [BOD81]).

Lessard (vide [LES81]) utilizou esta formulação para determinação de um perfil prévio da alocação dos condutores.

Outros autores também utilizaram este modelo para desenvolvimento de seus trabalhos, como, por exemplo, Bodin et al. (vide [BOD91] e [BOD92]).

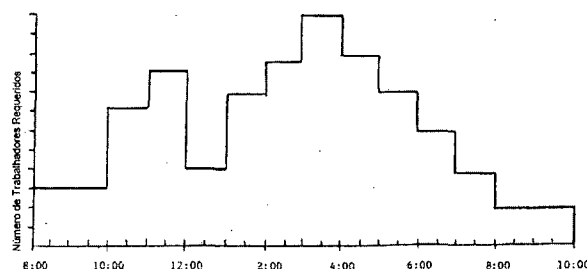


FIGURA 2.5 – Histograma de trabalhadores requeridos por tempo. Fonte: L. Bodin et al. (vide [BOD83])



### 2.3.1.3 Modelo teórico

Bodin et al. (vide [BOD83]) propõem um modelo que utiliza um conjunto hierárquico de restrições. No primeiro nível estão as restrições características de um problema de fluxo em rede ((2.6), (2.7), (2.8), (2.9), (2.10) e (2.11)), com as quais são definidos os CWPs (*Continuous crew Working Period*), que equivalem aos partições. No segundo nível de restrições os CWPs são agrupados em FWSs (*Full Work Schedules*), que correspondem às escalas, normalmente contendo de um a quatro CWPs. Cada FWS em potencial corresponde a um padrão  $l$ , sendo que o conjunto destes padrões é denominado  $\rho$ .

Os autores definem um grafo onde os nós são representados por  $d$ -trips, e o conjunto de arcos é subdividido em dois, os de ligação e os de retorno. O modelo proposto então pode ser formulado da seguinte maneira:

$$\text{Min} \left( \sum_{l \in \rho} c_l z_l + \sum_{(i,j) \in A_1} c_{ij} x_{ij} + \sum_{(i,j) \in A_2} c_{ij} y_{ij} \right) \quad (2.5)$$

s.a.:

$$\sum_{i:(i,j) \in A_1} x_{ij} + \sum_{i:(i,j) \in A_2} y_{ij} - \sum_{i:(i,j) \in A_1} x_{ji} + \sum_{i:(i,j) \in A_2} y_{ji} = 0 \quad \forall j \in N \quad (2.6)$$

$$\sum_{i:(i,j) \in A_1} x_{ij} + \sum_{i:(i,j) \in A_2} y_{ij} = 1 \quad \forall j \in N \quad (2.7)$$

$$\sum_{i:(i,j) \in A_1 \cap C} x_{ij} + \sum_{i:(i,j) \in A_2 \cap C} y_{ij} \leq |C| - 1 \quad \forall j \in N \quad (2.8)$$

$$\sum_{l \in \rho(i,j)} z_l + y_{ij} = 0 \quad \forall (i,j) \in A_2 \quad (2.9)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A_1 \quad (2.10)$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A_2 \quad (2.11)$$

$$z_l \in \{0,1\} \quad \forall l \in \rho \quad (2.12)$$

Onde:

- $A_1$  é o conjunto de arcos  $(i, j)$  tal que as  $d$ -trip  $i$  e  $j$  possam ser seqüenciadas dentro do mesmo CWP;
- $A_2$  é o conjunto de arcos de retorno  $(j, i)$ ;
- $z_l = 1$  se o padrão  $l$  é utilizado e  $z_l = 0$  caso contrário;
- $x_{ij} = 1$  se a  $d$ -trip  $j$  é subsequente da  $d$ -trip  $i$  e  $x_{ij} = 0$  caso contrário;
- $y_{ij} = 1$  se um CWP inicia na  $d$ -trip  $i$  e termina na  $d$ -trip  $j$  e  $y_{ij} = 0$  caso contrário;
- A restrição (2.8) garante a presença de apenas um arco de retorno em cada sequenciamento.
- As restrições (2.9), (2.12) garantem que todos os CWPs sejam cobertos pelos FWSs;
- $c_l$  representa os custos com alocação da tripulação e  $c_{ij}$ , os custos com alocação dos veículos.

A resolução do modelo proposto é inviável devido ao tamanho que ele adquire na prática. O número de padrões  $l$  cresce na proporção  $n^{2k}$ , onde  $n$  é o número de *d-trips* e  $k$  é o número de CWP's existentes nos FWSs. Neste sentido os autores sugerem a utilização de técnicas que possibilitem a subdivisão ou simplificação do problema.

### 2.3.1.4 Cobertura e Partição de conjuntos

Dentre os modelos mais utilizados na alocação de tripulação está o problema de cobertura ou partição de conjuntos (SCP - *Set Covering Problem* e SPP - *Set Partitioning Problem*). Utiliza os conceitos de blocos e partições, sendo que a combinação destes é denominada *duty*, a qual equivale às escalas. Consiste em encontrar o conjunto de *duties* de menor custo que cubram todas as partições. Pode ser formulado como se segue: (SCP)

$$\text{Min } \sum_{j=1}^n c_j x_j \quad (2.13)$$

s.a.

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad \text{para } i = 1, 2, \dots, M, \text{ número de partições,} \quad (2.14)$$

$$x_j \in \{0, 1\} \quad \text{para } j = 1, 2, \dots, N, \text{ número de } \textit{duties}. \quad (2.15)$$

onde:

$c_j$  = o custo da *duty*  $j$

$$a_{ij} = \begin{cases} 1 & \text{se a } \textit{duty} \ j \text{ contém o piece } \ i, \\ 0 & \text{caso contrário.} \end{cases} \quad x_j = \begin{cases} 1 & \text{se } \textit{duty} \ j \text{ faz parte da solução,} \\ 0 & \text{caso contrário.} \end{cases}$$

O SPP apresenta a mesma formulação, trocando apenas o sinal de desigualdade pelo de igualdade. Isto significa que no SCP cada partição deve ser coberta no mínimo por uma *duty*, enquanto no SPP elas são cobertas apenas por uma *duty*.

Ward (vide [WAR81]) utilizou uma decomposição, na qual classifica as escalas de acordo com os lugares que suas partições ocupam no perfil da demanda, o qual representa a variação da demanda de passageiros ao longo do dia, conforme a figura 2.6. Resolveu em seguida uma série de problemas SPP, em que são geradas escalas conforme seu horário de término. Porém, não garante a otimalidade, uma vez que as colunas não são consideradas simultaneamente.

Parker e Smith (vide [PAR81]) utilizaram o SCP com a inclusão de todas as viagens no perfil de serviço, conforme a figura 2.6. Restringiram ainda o número de colunas a serem consideradas através dos seguintes recursos:

- limites superiores e inferiores no tamanho das partições;
- limites superiores e inferiores no tamanho dos intervalos;
- restrição aos tempos de início e fim das partições conforme a classificação das escalas pelo perfil de serviço.

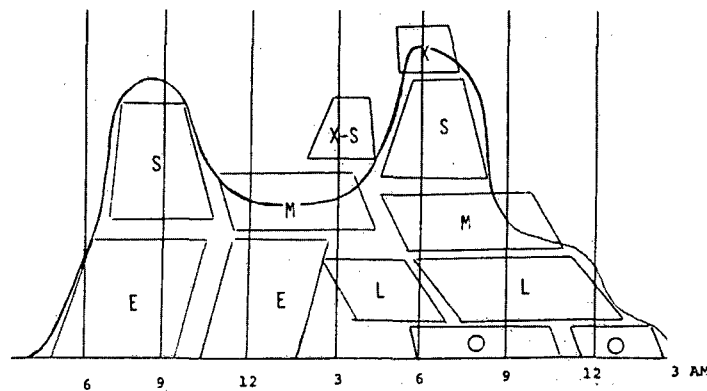


FIGURA 2.6 – Perfil da demanda e decomposição por tipo de escalas. Fonte: L. Bodin et al. (vide [BOD83])

Técnicas de programação linear, em conjunto com *branch end bound*, foram utilizadas nas resoluções do SPP propostas por Heurgon (vide [HEU72] e [HEU75]), Marsten (vide [MAR74]) e Ryan e Foster (vide [RYA81]) e nas resoluções de SCP desenvolvidas por WREN et al. (vide [WRE85]).

Mitra e Darby-Dowman (vide [MIT85]) utilizaram uma função de penalidades sobre um SCP modificado que permite a sobre cobertura e não cobertura de viagens.

Beasley (vide [BEA87]) combinou em um algoritmo técnicas de programação linear com testes de redução, heurísticas duais e otimização por subgradiente, buscando com isto a resolução do SCP.

Os algoritmos genéticos foram aplicados inicialmente no SCP e SPP por Beasley e Chu (vide [BEA94] e [CHU95]). Estes algoritmos foram testados com problemas provenientes da OR-library (vide [BEA90]), sendo que os de SCP apresentaram os melhores resultados, além do fato de que este algoritmo

demanda pouco tempo de desenvolvimento, se comparado com modelos do tipo do IMPAC.

Mayerle (vide [MAY96]) propôs um modelo dividido em três etapas:

- grafo para geração de partições solucionado através do problema de caminhos mínimos;
- procedimento de enumeração para gerar escalas alternativas que produz um conjunto de possíveis escalas de trabalho com seu respectivo custo e partições cobertas;
- problema de cobertura de conjuntos (SCP), no qual aplica um algoritmo genético.

O autor abrange ainda em seu modelo o problema de alocação de carga semanal de trabalho, no qual utiliza dois critérios para a sua distribuição: a equalização da carga de trabalho e a equalização dos salários pagos aos condutores.

#### **2.3.1.5 VSP e CSP simultâneos**

Bali et al. (vide [BAL83]) desenvolveram uma heurística para a resolução simultânea do problema de alocação de frota e tripulação utilizando um problema de *matching*, que é um SPP com no máximo dois zeros por coluna.

Consiste basicamente de três fases:

- construção de partições através de um procedimento de *matching*;

- melhoria das partições através de técnicas de divisão e recombinação;
- agrupamento das partições utilizando também *matching*.

#### **2.3.1.6 HASTUS**

Lessard et al. (vide [LES81]), utilizando uma alocação prévia de frota, propõem um modelo composto pelas seguintes partes:

- relaxamento do problema de maneira a gerar uma fragmentação dos blocos em partições. Para isto utiliza o modelo de alocação de condutores em local fixo, proposto por Bennet e Potts (vide [BEN68]). Este procedimento é denominado HASTUS-Macro;
- classificação das partições com base nos dados do perfil da demanda utilizando um algoritmo de atribuição e um procedimento heurístico;
- resolução do problema de *matching* através de heurística;
- Utilização de heurísticas de melhoramento das soluções.

#### **2.3.2 Considerações finais**

A formulação do problema de alocação de tripulação (geração de escalas) apresenta nos trabalhos mais recentes uma tendência de utilização do modelo de cobertura de conjunto. Este modelo deve enumerar todos os possíveis padrões de combinação, o que torna sua resolução explícita impossível. Dentre

os trabalhos pesquisados constata-se uma busca por procedimentos heurísticos que possam reduzir o tamanho do problema e assim gerar soluções aproximadas em tempos computacionais aceitáveis. Mayerle (vide [MAY96]) e Bodin et al. (vide [BOD83]), por exemplo, sugerem que o problema original seja partido em subproblemas menores. Os procedimentos heurísticos desenvolvidos focam-se na decomposição do problema ou na redução do número de colunas geradas. Porém, como se trata de heurísticas, estas técnicas de redução das colunas geradas nunca são totalmente detalhadas.

Destarte, a proposta do modelo descrito a seguir é a de gerar um conjunto de escalas viáveis reduzido, visando a otimização pelo problema de cobertura de conjuntos.



## CAPÍTULO III

### 3. TÉCNICA DE SOLUÇÃO

#### 3.1 Considerações iniciais

A técnica proposta para a geração de um conjunto reduzido de escalas viáveis é constituída basicamente por três etapas. Na primeira cria-se um conjunto de partições a partir de uma alocação prévia de frota. Na segunda, estas partições são combinadas de maneira a gerar todas as escalas viáveis, não importando a sua quantidade. Na terceira etapa as escalas são avaliadas de forma qualitativa e filtradas através de critérios determinados, formando assim um conjunto menor que possibilita a execução do problema de cobertura de conjuntos.

#### 3.2 Alocação prévia da frota

A utilização de uma alocação prévia de frota proporciona uma série de vantagens na alocação de tripulação, conforme apresentado no capítulo II. Neste sentido, buscou-se nos trabalhos já desenvolvidos um modelo que gerasse a alocação de frota desejada, utilizando dados reais e distintos. Os *softwares* comerciais disponíveis nesta área são todos estrangeiros, com custos desconhecidos e eficiências questionáveis. A implementação de um modelo proposto na literatura, por sua vez, pode demandar tempo e esforços não mensuráveis, não garantindo ainda a viabilidade do resultado.

Assim, optou-se pelo trabalho desenvolvido por Tramontim (vide [TRA01]), que além de ser bastante atual e utilizar uma modelagem inovadora (Busca Local Dirigida), disponibiliza um código computacional já desenvolvido e testado. A alocação de frota fornecida por este modelo é apresentada na forma de uma tabela, conforme o quadro 3.1. Cada linha da tabela é uma viagem, sendo que elas são ordenadas de forma crescente através dos horários de início.

QUADRO 3.1 – Alocação prévia da frota. Fonte: M. Tramontin (vide [TRA01])

DIA	GRUPO	VEICULO	SAIDAMIN	CHEGADA	ORIGEM	DESTINO
DIA ÚTIL	A	4305	207	247	CAPOEIRAS	CENTRO
DIA ÚTIL	A	4305	248	308	CENTRO	CENTRO
DIA ÚTIL	A	4305	310	370	CENTRO	CENTRO
DIA ÚTIL	A	4305	370	415	CENTRO	CENTRO
DIA ÚTIL	A	4305	415	460	CENTRO	CENTRO
DIA ÚTIL	A	4305	460	500	CENTRO	CENTRO
:	:	:	:	:	:	:

As colunas da tabela representam as seguintes informações:

- DIA — Período típico da viagem. São os dias nos quais as alocações são válidas. No modelo proposto, estes períodos são três: dias úteis; sábados; domingos e feriados;
- GRUPO — Grupo ao qual pertence o veículo. Subdivisão da frota ou das linhas que agrupa os veículos, ou as viagens com as mesmas particularidades;
- VEÍCULO — Prefixo do veículo alocado a uma viagem;
- SAIDAMIN e CHEGADA — Horário de início e fim de cada viagem, respectivamente. São contados a partir de um horário estabelecido como ZERO e expressos em minutos;

- ORIGEM e DESTINO — Local de início e fim de cada viagem.

A utilização de um horário como ponto ZERO tem como única finalidade facilitar a manipulação e processamento dos dados, uma vez que estes passam a ser do tipo numeral inteiro e de base decimal. O estabelecimento deste horário ZERO depende das características operacionais de cada frota, sendo normalmente utilizado alguma hora cheia anterior à saída do primeiro ônibus.

A introdução do campo GRUPO possibilita atender a uma exigência operacional que restringe a combinação de viagens de determinadas linhas, mais especificamente, as com patamares tarifários diferentes. A alocação de frota deve considerar esta restrição, relacionando então o veículo a um grupo. Assim, as partições são formadas a partir do veículo, e as escalas combinam as partições daquele grupo. Por exemplo, se a linha Capoeiras pode intercalar viagens com a linha Abraão, então as partições referentes aos veículos alocados para estas linhas podem ser combinadas entre si para formar as escalas.

Esta abordagem proporciona um aumento do número de escalas geradas, uma vez que permite um maior número de combinações entre as viagens.

### **3.3 Caracterização das regras trabalhistas**

Depois de estabelecida a alocação de frota, é necessário levantar os parâmetros que vão reger toda a alocação da tripulação.

A relação de trabalho empregado-empregador, no Brasil, é balizada pelas leis trabalhistas estabelecidas na CLT. Porém algumas categorias, como, a dos condutores do transporte coletivo urbano, necessitam de condições de trabalho especiais, pois sua atividade está relacionada à segurança dos passageiros.

Assim, nas negociações trabalhistas que ocorrem entre o sindicato dos trabalhadores e o patronal, uma série de regras são estabelecidas, como por exemplo:

- Jornada – quantidade diária de horas de trabalho, com remuneração normal, que cada condutor pode executar;
- Horas extras permitidas – quantidade máxima de horas além da jornada que pode ser alocada à tripulação;
- Valor das horas extras – custo da hora extra, sendo representado por um percentual em relação ao custo da hora normal;
- Intervalo – tempo de descanso obrigatório para a tripulação. Pode ser dividido em mais de uma parada, possuir tempo fixo ou variável e apresentar ou não restrições do horário de início.

Dentre estas regras, algumas são relacionadas à construção de partições e outras, à geração de escalas, conforme mostrado a seguir.

### 3.4 Construção das Partições

Uma vez estabelecidas a alocação de frota e as regras trabalhistas, pode-se então iniciar a geração de uma solução viável para a alocação de tripulação. A primeira etapa é a construção das partições, ou fragmentação de blocos, a qual consiste em combinar as viagens ordenadas na alocação de frota, respeitando as restrições elencadas no tópico 3.4.1..

A construção de partições é uma tarefa comum a praticamente todos os trabalhos abordados no capítulo II, sendo apresentadas várias técnicas, como o caminho dos mínimos custos (vide [LES81]), utilização de limites superiores e inferiores (vide [PAR81]), entre outros.

O modelo em questão visa gerar o maior número possível de partições viáveis, ao contrário de outros trabalhos que limitam esta criação, buscando diminuir o tamanho do problema. Para isto, utiliza uma heurística à qual é incrementada a técnica de *Flow Formulation*, ou Formulação de Fluxo, proposta por Lessard (vide[LES81]).

Alguns conceitos empregados na construção das partições necessitam ser elencados para o completo entendimento da técnica, sendo que entre parênteses está a denominação utilizada para cada um deles no desenvolvimento do algoritmo:

### a) Tamanho das Partições

Refere-se ao tamanho, medido em minutos, que cada partição pode ter. É caracterizado por um limite inferior ( $li$ ) e limite superior ( $ls$ ). Se o  $li$  for muito pequeno, pode ocorrer que uma partição seja formada por apenas uma viagem, o que aumenta enormemente número destas. Já o  $ls$ , no máximo, pode ser igual à jornada de trabalho, senão nenhuma tripulação poderá executá-lo. Estes parâmetros afetam o resultado final do problema, pois alteram a quantidade e perfil das partições formadas. Não possuem, no entanto, um valor padrão, devendo então ser estimados. Os ensaios mostrados no capítulo seguinte fornecem dados referentes às alterações no resultado final conforme as variações destes parâmetros, orientando, ainda, a escolha do melhor perfil.

### b) Relaxamento dos limites

Na etapa de construção das escalas, a combinação das partições pode causar a não cobertura de algumas viagens. Neste contexto, os valores de  $li$  e  $ls$  são então relaxados, visando com isto gerar partições com um novo padrão de dimensionamento para cobrir as viagens restantes.

O relaxamento utiliza a variável  $loop$  para ajustar o limite inferior e superior da seguinte maneira:

$$li_2 = li_1 (1 - loop_i / 100) \quad (3.1)$$

$$ls_2 = ls_1 (1 + loop_s / 100) \quad (3.2)$$

Considerando que ambos os *loops* sejam iguais a 10%, se  $li = 100$  e  $ls = 200$ , implica dizer que no primeiro relaxamento  $li = 90$  e  $ls = 220$ , no segundo relaxamento  $li = 81$ ,  $ls = 242$ , e assim por diante, até não restar nenhuma viagem descoberta.

### 3.4.1 Restrições

Depois de estabelecido os valores dos parâmetros supramencionados, a construção das partições consiste basicamente em combinar as viagens ordenadas na alocação de frota, respeitando as seguintes restrições:

#### 1) Dia típico e veículo

As viagens devem pertencer ao mesmo dia típico ( $di$ ) e ao mesmo veículo ( $ve$ ), ou seja:

$$di_i = di_j \quad i e j \in N \quad (3.3)$$

$$ve_i = ve_j \quad i e j \in N \quad (3.4)$$

#### 2) Local e horário

A alocação prévia de frota, ao promover a otimização do uso dos veículos, assegura que o local de saída ( $LS_j$ ) da viagem  $j$  é o mesmo que o local de chegada ( $LC_i$ ) da viagem  $i$ :

$$LC_i = LS_j \quad i e j \in N \quad (3.5)$$

Garante ainda que o horário de saída de  $j$  ( $HS_j$ ) é maior ou igual ao horário de chegada da viagem  $i$  ( $HC_i$ ) mais o tempo de deslocamento entre estes pontos

$$HS_j \geq HC_i + T_{desloc} \quad i, j \in N \quad (3.6)$$

Com isto, tem-se a certeza da viabilidade das partições criadas a partir de um veículo.

### 3) Tamanho da Partição

A duração de uma partição ( $durap$ ) não pode ser menor que o limite inferior ( $li$ ) e nem maior que o limite superior ( $ls$ );

$$li \leq durap \leq ls \quad (3.7)$$

A  $durap$  é medida em minutos e equivale à diferença entre o horário de chegada da última viagem e o horário de saída da primeira viagem que compõem a partição, ou seja,  $durap = HC_j - HS_i$ , com  $i, j \in N$ .

Os efeitos da variação dos valores de  $li$  e  $ls$  na quantidade de partições geradas e no resultado final da alocação de tripulação são mostrados no capítulo IV.

### 4) Formulação de fluxo

As possíveis combinações de viagens podem formar partições que não são viáveis do ponto de vista de geração de escalas, ou seja, partições que não podem ser combinadas com nenhuma outra. Assim, além das restrições



relatadas acima, é acrescentada ao modelo uma técnica que visa impedir a formação deste tipo de partições. Trata-se da *flow formulation*, ou formulação de fluxo, proposta por Lessard (vide [LES81]) no intuito de particionar, de forma mais eficiente, os blocos criados no seu modelo. Sua lógica baseia-se em não gerar partições cujo horário de início não coincide com nenhum horário de chegada de outra partição, uma vez que nunca existirá uma partição que anteceda esta.

No modelo apresentado por Lessard a restrição é resumida a:

$$\sum_{i \in T_p} y_{ik} - \sum_{j \in T_p} y_{kj} = b_k^p \quad \forall p \in \forall k \in T_p$$

onde:

$$- \quad b_k^p = \begin{cases} -1 & \text{se } k \text{ é o horário de saída do bloco } p; \\ 1 & \text{se } k \text{ é o horário de chegada do bloco } p; \\ 0 & \text{caso contrário} \end{cases}$$

-  $T_p$  é o conjunto de todos os horários de saída e chegada das partições

A figura 3.1 ilustra o conceito da formulação de fluxo, onde os arcos representam as partições viáveis, neste exemplo, restringidos a duas horas de duração. É considerado ainda que todos os pontos podem ser pontos de descanso.

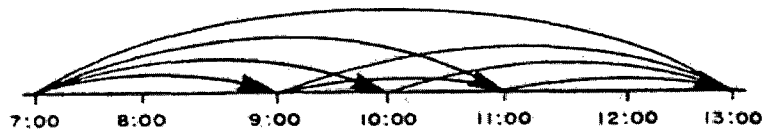


FIGURA 3.1 – Formulação de Fluxo. Fonte: R. Lessard et al. (vide [LES81])

As partições viáveis são:

7:00 - 9:00;    7:00 - 10:00;    7:00 - 11:00;    7:00 - 13:00;  
 9:00 - 11:00;    9:00 - 13:00;    10:00 - 13:00;    11:00 - 13:00.

Neste modelo, cada partição criada possui dois valores distintos que são:  $saída[np]$  e  $chegada[np]$ . Representam, respectivamente, a primeira e a última viagem que compõem a partição. A formulação de fluxo é caracterizada pela variável  $ff[i]$ , onde  $i$  é uma viagem da tabela de horários, sendo que:  $ff[i] = i-1$

Após a construção da primeira partição e estabelecida a  $chegada[1]$ , só se poderá criar uma outra partição a partir da viagem  $i$  se  $ff[i]$  for igual à chegada do última partição, ou seja:

$$ff[i] = chegada[np-1] \quad (3.8)$$

QUADRO 3.2 – Tabela de horários. Fonte: Empresa Estrela Transp. Coletivos

VIAGEM	DIA	GRUPO	VEICULO	SAÍDA(MIN)	CHEGADA	ORIGEM	DESTINO
1	DIA ÚTIL	A	4305	207	247	CAPOEIRAS	CENTRO
2	DIA ÚTIL	A	4305	248	308	CENTRO	CENTRO
3	DIA ÚTIL	A	4305	310	370	CENTRO	CENTRO
4	DIA ÚTIL	A	4305	370	415	CENTRO	CENTRO

Considerando a tabela do quadro 3.2, pode-se exemplificar a formulação como se segue: A partição número 1, conforme determinado o limite superior e inferior, é composta pelas viagens 1, 2 e 3, assim:

$$p[1] = 1 \ 2 \ 3; \quad saída[1] = 1; \quad chegada[1] = 3.$$

Se a duração da partição já atingiu seu limite, ou seja, nenhuma outra viagem poderá ser acrescentada a ela, a segunda partição a ser criada ( $np = 2$ ) obrigatoriamente deve começar com a viagem 4, pois  $ff [4] = 4 - 1 = 3$  e a  $chegada[np-1] = chegada[1] = 3$ , atendendo assim a restrição 3.6.

Uma característica peculiar deste modelo consiste no fato de que se busca gerar o maior número possível de partições viáveis, enquanto outros autores restringem esta geração. Tal particularidade deve-se ao fato de que a restrição que visa facilitar a otimização do problema de cobertura de conjuntos não está no número de partições geradas e sim no filtro aplicado às escalas geradas, conforme mostrado a seguir.

### 3.5 Geração das escalas

A construção das escalas diárias de trabalho consiste basicamente em agrupar as partições viáveis, anteriormente criadas, seguindo algumas regras. Trata-se da alocação da tripulação propriamente dita, onde as escalas formadas, que serão designadas à tripulação, devem satisfazer às restrições relacionadas à legislação trabalhistas e às características operacionais que envolvem cada problema.

A geração destas escalas é obtida através de um método heurístico que combina as partições conforme as seguintes restrições:

### 1) Dia típico e grupo

Considerando que  $Np = \{p_1, p_2, \dots, p_n\}$  seja o conjunto das partições geradas, o sequenciamento de duas partições só é possível se estas pertencerem ao mesmo dia típico ( $di$ ) e ao mesmo grupo de veículos ( $gr$ ), ou seja:

$$di_{p_n} = di_{p_m} \quad p_n \text{ e } p_m \in Np \quad (3.9)$$

$$gr_{p_n} = gr_{p_m} \quad p_n \text{ e } p_m \in Np \quad (3.10)$$

### 2) Horário

O horário de saída da partição a ser seqüenciada deve ser sempre maior que o horário de chegada da partição anterior, isto é:

$$HS_{p_n} \geq HC_{p_m} \quad p_n \text{ e } p_m \in Np \quad (3.11)$$

### 3) Local

O local de saída e chegada das partições a serem combinadas deve coincidir, ou seja:

$$LS_{p_n} = LC_{p_m} \quad p_n \text{ e } p_m \in Np \quad (3.12)$$

Os parâmetros estabelecidos na legislação trabalhista como: duração da jornada de trabalho (*jornada*), intervalo obrigatório (*intervalo*), e quantidade de

horas extras permitidas (*hextra*) são os sujeitos das restrições a seguir. Foram descritos na tópico 3.5 e são fundamentais para a confecção das escalas.

#### 4) Intervalo

O intervalo obrigatório pode possuir um valor mínimo (*intervmin*) e um valor máximo (*intervmax*), conforme cada problema. Em casos onde o intervalo possui um valor fixo, os intervalos máximos e mínimos são iguais. Neste modelo não se utiliza a restrição que visa determinar como o intervalo deve ser alocado na escala, ou seja, quando deve ser seu início. Tal posição decorre da premissa de que o tamanho máximo utilizado para a criação das partições (*Is*) deve ser aproximadamente igual ao meio da jornada de trabalho, sendo que este é o padrão mais utilizado para se alocar o intervalo. Para as escalas com apenas uma partição o intervalo é considerado igual a zero. Para escalas com duas partições a duração do intervalo (*duraint*) é a diferença entre o início da segunda partição e final da primeira, ou seja,  $duraint = HS_{p_{m+1}} - HC_{p_m}$ , com  $p_m \in Np$ . O *duraint* calculado tem que ser maior que o intervalo mínimo exigido (*intervmin*), caso contrário a escala é inviável. Porém, o *duraint* pode ser maior que o intervalo máximo permitido (*intervmax*), sendo que neste caso o intervalo considerado será o *intervmax*, pois o que excede este valor é considerado e remunerado como horas trabalhadas. Assim a restrição se resume a:

$$duraint \geq intervmin \quad (3.13)$$

#### v) Duração da escala

A duração de uma escala (*duraesc*) é medida em minutos e equivale à diferença entre o horário de chegada da última partição e o horário de saída da primeira partição que a compõe, ou seja,  $duraesc = HC_{p_n} - HS_{p_m}$  com  $p_n$  e  $p_m \in N_p$ .

A *duraesc* deve ser sempre menor ou igual à soma da jornada de trabalho com o intervalo obrigatório e a quantidade de horas extras permitidas, isto é:

$$duraesc \leq (jornada + intervalo + hextra) \quad (3.14)$$

As escalas geradas seguem o mesmo padrão de apresentação proposto para as partições. São indexadas e representadas pela seqüência dos índices de cada viagem que as compõem, como por exemplo:  $e[1] = 5 \ 6 \ 7 \ - \ 14 \ 15$ , ou seja, a escala número 1 cobre as viagens 5, 6, 7, 14 e 15.

### 3.6 Seleção das escalas

A geração de escalas, neste modelo, visa criar o maior número possível de combinações, não possuindo assim restrições com este propósito. Porém, conforme os parâmetros de geração, a quantidade de escalas viáveis fica muito grande, sendo este o fator proibitivo para a otimização através do problema de cobertura de conjuntos. Neste sentido, foi introduzida no modelo uma rotina que seleciona escalas conforme alguns índices elencados. Tais índices são calculados para cada escala gerada e possuem a função de permitir sua análise qualitativa. São eles:

- Horas extras pagas (*hextrapg*) – quantidade de horas extras alocadas em cada escala. Sua quantidade é multiplicada pelo fator de acréscimo sobre a hora normal (*hextra*), como mostrado a seguir:

$$Hextrapg = (duraesc - intervalo - jornada) * hextra \quad (3.15)$$

- Horas trabalhadas (*htrab*) – são as horas em que a tripulação efetivamente trabalhou durante uma escala de trabalho. É o somatório da duração de cada viagem que compõe a escala;
- Horas pagas (*hpg*) – é o somatório das horas normais e extras trabalhadas. Caso não haja horas extras e a quantidade de horas normais seja menor que a jornada-padrão, então *hpg* é a própria jornada;
- Eficiência (*ef*) – é a hora trabalhada dividida pelas horas pagas. É um indicador de como a escala utiliza a tripulação. Quando *ef* apresenta seu valor máximo (100%), indica que se paga exatamente o que foi trabalhado, ou seja, a escala é eficiente. Baixos valores de *ef* significam que a escala possui uma alocação com intervalos muito longos, ou partições muito pequenas, onde efetivamente se trabalha menos e se recebe a remuneração completa;

$$ef = htrab / hpg \quad (3.16)$$

- Quantidade de coberturas por viagens (*quantvi*) – indica quantas vezes cada viagem foi coberta, considerando o conjunto de escalas geradas.

A seleção utilizada neste modelo filtra as escalas conforme os seguintes parâmetros: eficiência desejada e quantidade de cobertura por viagem.

O critério de eficiência desejada baseia-se no índice de eficiência ( $ef$ ) para reduzir a quantidade de escalas através da eliminação das escalas de baixa eficácia. A opção de valores elevados de  $ef$  ( $>90\%$ ) reduz o tamanho do conjunto final de escalas. A influência deste parâmetro no resultado final da alocação é demonstrada no capítulo seguinte.

Determinadas viagens podem ficar descobertas conforme o valor estabelecido para o filtro da eficiência. Tal situação deve-se ao fato de que certas viagens são cobertas apenas por uma escala e esta possui eficiência abaixo do estabelecido. Assim, a “quantidade de cobertura por viagens” refere-se à mínima quantidade de coberturas exigidas para cada viagem, no conjunto selecionado de escalas. Desta maneira, o filtro prioriza a eficiência e garante a cobertura de todas as viagens, mesmo que seja através de escalas pouco eficientes, garantindo assim a viabilidade dos conjuntos gerados.



## CAPÍTULO IV

### 4. IMPLEMENTAÇÃO E RESULTADOS COMPUTACIONAIS

#### 4.1 Considerações iniciais

As particularidades da técnica proposta só podem realmente ser mensuradas através de sua implementação, sendo que para verificar sua eficiência é necessária a comparação com situações reais. Neste sentido desenvolveu-se um algoritmo utilizando Delphi 5, que é uma evolução do compilador Pascal, dispondo de recursos de orientação a objeto. Tal característica foi utilizada apenas no relacionamento com a base de dados e na confecção da interface, sendo que o código foi escrito de forma procedural. A base de dados se resume a uma tabela no formato *Excel* ou similar.

Os ensaios foram executados em um computador com processador de 800MHz, 64Mb de memória e sistema operacional *Windows 98*.

#### 4.2 Estrutura dos ensaios

Os ensaios realizados objetivaram a avaliação de aspectos distintos da técnica, sendo então divididos em três tipos: O primeiro consiste na análise dos efeitos que a variação de alguns parâmetros causa na solução final. No segundo ensaio, diversos conjuntos de escalas criados pela técnica são processados no algoritmo de alocação de tripulação desenvolvido por Mayerle (vide [MAY96]), mostrando assim os diferentes resultados alcançados. Por fim, o terceiro

ensaio compara uma situação real e operante com uma solução obtida através do algoritmo de Mayerle juntamente com a técnica proposta.

### **4.3 Dados**

Os dados utilizados para desenvolvimento dos dois primeiros tipos de ensaios são dados reais e provenientes da alocação prévia de frota desenvolvida por Tramontin (vide [TRA01]), os quais se referem as 685 viagens realizadas pela empresa Transportes Coletivos Estrela, com 54 veículos, em 13 linhas do sistema municipal de Florianópolis. Os custos utilizados na alocação da frota são fictícios, conforme considerado por Tramontin, porém respeitando-se a proporcionalidade entre eles. Já os dados para o terceiro ensaio representam a atual operação executada pela empresa Ribeironense Transportes Coletivos, também do sistema de Florianópolis.

### **4.4 Ensaio I – Parâmetros de geração e seleção das escalas**

A geração das escalas depende da existência de um conjunto viável de partições e de alguns parâmetros trabalhistas citados no capítulo anterior. Estes parâmetros apresentam valores diferenciados para cada município ou região, sendo que no caso em que não há convenção trabalhista definindo os mesmos, valem os estabelecidos pela Consolidação das Leis Trabalhistas (CLT). Em todos os ensaios a seguir foi utilizado o padrão vigente na região metropolitana de Florianópolis, que é: Jornada de trabalho de 400 minutos; Intervalo mínimo de 20 e máximo de 60 minutos; e 2 horas extras permitidas.

Os parâmetros trabalhistas não são objeto de análise destes ensaios, permanecendo assim constante em todos eles. O foco é o conjunto viável de partições, cuja confecção necessita de três parâmetros básicos, que são: limite inferior ( $li$ ), limite superior ( $ls$ ) e percentual para o relaxamento ( $loop$ ), conforme descrito a seguir.

#### 4.4.1 Parâmetros de geração das Partições

##### 4.4.1.1 Limite inferior e superior

Os valores de  $li$  e  $ls$  influem diretamente no perfil e na quantidade das partições a serem criadas, pois estabelecem o seu tamanho mínimo e máximo. Quanto menor a faixa limite preestabelecida, menor será o número de partições que se enquadram nesta e, conseqüentemente, maior será a quantidade de viagens não cobertas na primeira iteração.

O primeiro ensaio impôs uma variação uniforme aos limites de geração, sendo que as quantidades de partições e escalas obtidas são mostradas na tabela 4.1. Foi estabelecido um valor fixo para o relaxamento de  $li$  e de  $ls$  ( $loop=90\%$ ), visando assim avaliar somente os efeitos da variação do tamanho das partições no resultado final.

$li$ (min)	$ls$ (min)	Quant. Partições	Quant. Escalas Geradas	tempo (s)
180	220	398	3888	< 1
160	240	561	3839	< 1
140	260	983	21894	4
120	280	1365	52968	18

TABELA 4.1 – Variação uniforme de  $li$  e  $ls$

Outro tipo de variação testada foi a diferenciada, onde se mantém fixo um dos limites e se modifica o valor do outro, sendo que os resultados são mostrados na tabela 4.2.

<i>li</i> (min)	<i>ls</i> (min)	Quant. Peças	Quant. Escala Geradas	tempo (s)
100	140	421	7316	1
100	160	525	22048	1
100	180	745	63144	21
140	220	647	11041	1
120	220	882	34727	7
100	220	1139	126509	197

TABELA 4.2 – Variação diferenciada de *li* e *ls*

A variação diferenciada possibilita uma maior flexibilidade no perfil das partições geradas, proporcionando assim uma maior diversidade de conjuntos e conseqüentemente melhor adequação aos problemas distintos.

#### 4.4.1.2 Relaxamento dos limites

Depois de gerado o primeiro conjunto de escalas, é realizado uma rotina de verificação de viagens não cobertas. No caso de alguma ser detectada, os valores de *li* e *ls* são então relaxados, possibilitando assim a criação de partições menores e maiores que visam cobrir estas viagens restantes.

Neste ensaio fixou-se um limite inicial para as partições, sendo que em seguida foram variados igualmente os valores dos *loops* de *li* e *ls*, obtendo-se os resultados mostrados na tabela 4.3.

<i>li</i>	<i>ls</i>	<i>loop</i>	<i>Pieces</i>	Escala Geradas	tempo
140	220	10%	565	10643	8
140	220	20%	565	10645	4
140	220	30%	566	10646	3
140	220	40%	565	10651	2
140	220	50%	569	10658	2
140	220	60%	575	10672	2
140	220	70%	594	10761	2
140	220	80%	615	10869	2
140	220	90%	647	11041	1
140	220	100%	15714	25865	9

TABELA 4.3 – Variação do parâmetro de relaxamento dos limites

Existe, porém, a necessidade de se variar distintamente os valores de *li* e *ls*, como no exemplo utilizado, onde as regras trabalhistas não estabelecem um tempo para início do intervalo obrigatório. Assim, torna-se interessante manter o valor de *ls* constante e igual à metade da jornada de trabalho, pois desta maneira direciona-se a alocação do intervalo para este ponto. Neste sentido realizou-se um ensaio onde o *loop* de *li* foi sendo alterado em 10% e o de *ls* fixo em 0%, ou seja, sem relaxamento. A tabela 4.4 mostra os resultados obtidos.

<i>li</i>	<i>ls</i>	<i>loop</i>	<i>Pieces</i>	Escala Geradas	tempo
140	220	10%	566	10645	8
140	220	20%	566	10647	3
140	220	30%	566	10647	3
140	220	40%	566	10652	1
140	220	50%	570	10660	1

TABELA 4.4 – Variação diferenciada do parâmetro de relaxamento dos limites

Observa-se que para estes padrões utilizados, a não variação de um dos limites não proporciona grande alteração nas quantidades de partições e escalas geradas.

As conclusões sobre os melhores parâmetros de geração dependem da análise das soluções de alocação proporcionadas por estes diferentes conjuntos de escalas, conforme descrito a seguir.

#### **4.4.2 Parâmetros de seleção das escalas**

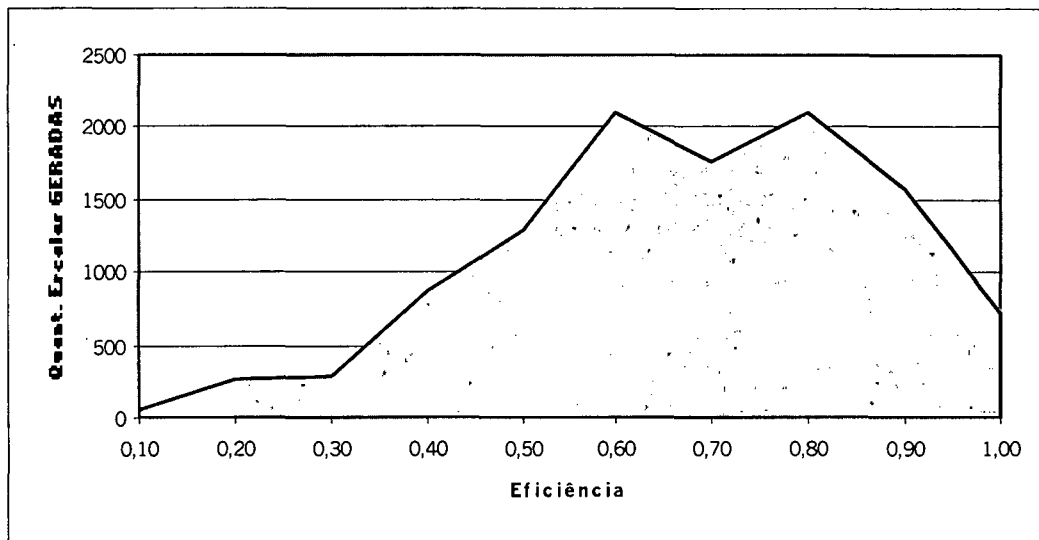
Uma vez geradas as escalas, a técnica propõe a seleção de um conjunto mais reduzido, baseada em dois parâmetros distintos: a eficiência da escala e a quantidade de coberturas por viagens. As variações nos valores destes parâmetros proporcionam diferentes efeitos no resultado final da seleção, conforme descrito a seguir.

##### **4.4.2.1 Eficiência desejada**

O conjunto de partições determina não só a quantidade de escalas geradas como também o seu perfil. A figura 4.5 mostra como as faixas de eficiência estão distribuídas dentro de dois conjuntos distintos de escalas.

A figura 4.5a apresenta 13% de suas escalas abaixo de 40% de eficiência, enquanto a figura 4.5b apresenta somente 8%. Com isto é possível selecionar um grupo mais reduzido de escalas com eficiências acima de um valor mínimo estabelecido. Quanto menor este valor, mais escalas entrarão na seleção, sendo que no exemplo "a" apenas 13 escalas têm eficiência abaixo de 10%.

a)  $l_i = 140$  e  $l_s = 220$



b)  $l_i = 120$  e  $l_s = 220$

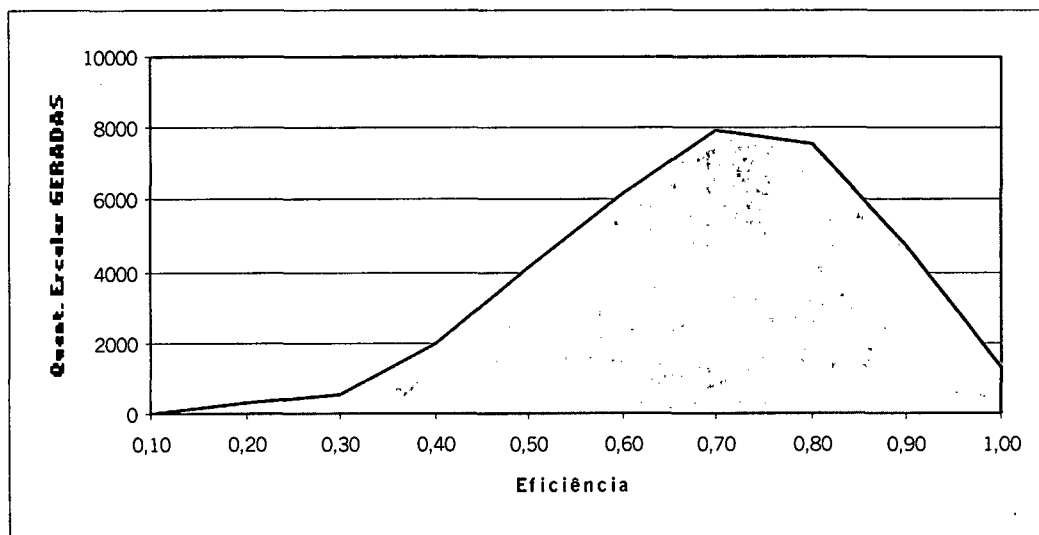


FIGURA 4.5 – Distribuição das escalas geradas a partir da eficiência

#### 4.4.2.2 Quantidade de coberturas por viagem

A seleção pela eficiência pode ocasionar a não cobertura de determinadas viagens durante a seleção. Neste sentido o parâmetro de quantidade mínima de coberturas por viagem tem por finalidade garantir que todas elas sejam

cobertas, independentemente de estarem alocadas em escalas pouco eficientes. Porém, não garante que as viagens sejam cobertas o número de vezes solicitado, pois podem existir viagens que são cobertas apenas uma vez.

Utilizando o mesmo conjunto de escalas do ensaio anterior ( $l_i = 140$  min;  $l_s = 220$  min;  $loop = 90\%$ ; partições = 647 e escalas geradas = 11041), estabeleceram-se três valores para a quantidade mínima de coberturas, sendo que para cada um deles foram realizados ensaios com diferentes valores de eficiência para seleção. A tabela 4.6 mostra os resultados da variação do parâmetro sobre as quantidades de escalas filtradas.

Quant. Coberturas	Eficiência (%)	Escalas Filtradas	Acréscimo de Escalas
1	40	10052	-
2	40	10081	29
4	40	10141	60
1	70	5534	-
2	70	5587	53
4	70	5688	101
1	99	299	-
2	99	463	164
4	99	806	343

TABELA 4.5 – Variação da quantidade de coberturas por viagem

Os efeitos da variação na quantidade de coberturas dependem da eficiência ajustada, pois quanto maior a eficiência desejada, menor é o número de escalas filtradas e, portanto, maior é o número de viagens não cobertas. Assim, quando a cobertura solicitada passa de 1 para 2 na eficiência de 40%, 29 escalas são selecionadas a mais, enquanto na eficiência de 99% são 164.



## 4.5 Ensaio II – Aplicação de cobertura de conjuntos

O conjunto de escalas viáveis possibilita a alocação de tripulação através do modelo de cobertura de conjunto. Neste sentido foi utilizado o algoritmo desenvolvido por Mayerle (vide [MAY96]), discutido no capítulo II, sendo executado com diferentes conjuntos de escalas obtidos pela técnica proposta. O ensaio foi dividido em duas diferentes etapas, como descrito a seguir:

### 4.5.1 Mesmo conjunto de escalas

A primeira parte deste ensaio consiste em avaliar os efeitos causados na alocação final do algoritmo de cobertura de conjuntos. Para isto, utiliza como base um único conjunto de escalas geradas, do qual são selecionados vários conjuntos distintos através da variação dos parâmetros de seleção. O conjunto base é o mesmo empregado nos ensaios anteriores, ou seja,  $l_i = 140$  min;  $l_s = 220$  min;  $loop = 90\%$ ; partições = 647 e escalas geradas = 11041.

Inicialmente fixou-se um valor para a eficiência desejada ( $ef$ ), variando então a quantidade de coberturas por viagem (*coberturas*). A tabela 4.6 mostra os resultados destas situações sobre a alocação no algoritmo de Mayerle, que é representado na tabela 4.6 pelas últimas quatro colunas.

Ef. (%)	Coberturas	Escalas Filtradas	tempo (s)	Iterações	tempo (s)	Custo	Duties
99	1	299	8	3000	8	69112	144
99	2	463	14	9800	30	60843	129
99	4	806	8	19300	41	56944	124
70	1	5534	32	31600	67	56353	124
70	2	5587	39	28900	61	54946	121
70	4	5688	32	36900	76	54212	120
50	1	9006	141	36700	77	52410	118
50	2	9048	147	36600	76	51845	117
50	4	9120	161	43500	88	51338	116
40	1	10052	96	46200	94	52140	116
40	2	10081	100	39800	81	51388	116
40	4	10141	89	40400	83	51375	116

TABELA 4.6 – Alocação conforme variação nos parâmetros de seleção.

A coluna *ruties* refere-se às quantidades finais de escalas de trabalho utilizadas na alocação. Elas não representam, necessariamente, a quantidade de condutores necessários, pois conforme determinadas regras trabalhistas, um condutor pode realizar mais de uma escala.

O melhor resultado (menor custo e menor quantidade de *duties*) alcançado está em destaque na tabela 4.6, sendo obtido com uma eficiência intermediária (50%) e com a maior quantidade de coberturas por viagem. Isto significa que acrescentando ao conjunto escalas com eficiência entre 40 e 50%, não ocorre melhoria na solução. Observa-se ainda que para a mesma eficiência, o aumento de coberturas por viagem proporciona uma redução no custo da alocação final.

#### 4.5.2 Conjuntos distintos de escalas

A segunda parte deste ensaio consiste em verificar como diferentes conjuntos de escalas geradas influem na alocação final. Para isto foram geradas escalas com diferentes tamanhos de partições, em seguida aplicado os mesmos

padrões de seleção e então executada a alocação utilizando o algoritmo de Mayerle. A variável de relaxamento utilizada foi a mesma para  $li$  e  $ls$ , sendo igual a 90%. A tabela 4.7 ilustra os resultados obtidos.

a)  $li = 140$ ;  $ls = 220$ ; escalas geradas = 11041:

Ef (%)	Coberturas	SELEÇÃO		ALOCAÇÃO			
		Escalas Filtradas	tempo (s)	Iterações	tempo (s)	Custo	Duties
50	4	9120	161	43500	87,88	51338	116
10	4	11033	2	37900	93,81	51872	116

b)  $li = 120$ ;  $ls = 220$ ; escalas geradas = 34727:

Ef (%)	Coberturas	SELEÇÃO		ALOCAÇÃO			
		Escalas Filtradas	tempo (s)	Iterações	tempo (s)	Custo	Duties
50	4	30197	51	40200	87,44	50854	110
10	4	34724	61	40500	87,27	49929	109

c)  $li = 100$ ;  $ls = 220$ ; escalas geradas = 126509:

Ef (%)	Coberturas	SELEÇÃO		ALOCAÇÃO			
		Escalas Filtradas	tempo (s)	Iterações	tempo (s)	Custo	Duties
50	4	113563	976	69500	157,48	48415	104
10	4	126506	1020	68900	160,21	48344	105

TABELA 4.7 – Alocação de tripulação conforme os diferentes conjuntos de escalas filtradas

Os melhores resultados foram obtidos com os maiores conjuntos de escalas, porém sendo despendido com estes um maior tempo de processamento.

Conclui-se, assim, que sempre se deve buscar o maior conjunto de escalas possível, sendo que no exemplo utilizado, o tempo computacional tornou proibitiva a geração de mais de 120 mil escalas, sendo este então o fator limitador deste algoritmo.

#### 4.6 Ensaio III – Análise comparativa da técnica proposta

Todos os ensaios anteriores visavam avaliar o comportamento da solução final conforme a variação dos parâmetros envolvidos. O ensaio que se segue, por sua vez, objetiva verificar a eficácia da técnica proposta, através da comparação com um conjunto de escalas que estejam operando em situações reais.

As escalas utilizadas referem-se à atual operação da empresa Ribeironense, cujas características são as seguintes:

- ✓ Número de linhas: 17
- ✓ Número de viagens por dia (dias úteis): 770
- ✓ Frota disponível: 86
- ✓ Regras trabalhistas: Convenção SETUF/SINDIMOC Florianópolis
- ✓ Quant. de *duties* (escalas operantes): 153
- ✓ Custos das escalas (homem/hora): **1.519**

Visando direcionar a alocação do intervalo obrigatório próximo ao meio da jornada de trabalho, estabeleceu-se como tamanho máximo para as partições (*ls*) 220 minutos, não variando este valor durante as iterações (*loop ls* = 0%).

Inicialmente foram gerados alguns conjuntos de escalas para verificar o comportamento da geração para este problema específico. Constatou-se um grande número de partições com apenas uma viagem, caracterizando a existência de viagens longas (> 180min), pois quanto menor forem estas, maior será a quantidade necessária delas para formar uma partição. Assim, optou-se por uma geração com as seguintes características:

<i>li</i> (min)	<i>ls</i> (min)	<i>loop li</i>	<i>loop ls</i>	partições	Escalas geradas	tempo (s)
105	220	10%	0%	682	69.098	510

A seleção das escalas, por sua vez, ficou com se segue:

Coberturas por viagem	Eficiência desejada	Escalãs filtradas	tempo (s)
4	10%	69.098	301

O processamento deste conjunto de escalas despendeu, no algoritmo de cobertura de conjuntos, 92.000 iterações e 497 segundos, sendo que os resultados são mostrados na tabela 4.2, onde ainda é realizado um

comparativo com as atuais escalas da Ribeironense. Os custos são representados em homem/hora.

	OBTIDOS	RIBEIRONENSE
CUSTOS (homem/h)	1.487	1.519
Quant. de <i>DUTIES</i>	192	153

Mesmo considerando uma maior quantidade de *duties*, o resultado é bastante satisfatório, pois representa uma redução real de 2,2% no custo diário da empresa, que equivale a aproximadamente R\$70.000,00 por ano, considerando a diferença de 32 horas/homem por dia. Como as *duties* não representam, necessariamente, a quantidade de condutores, o número final de motoristas necessários para operacionalizar as escalas geradas pode ser equiparado ao atualmente utilizado na empresa através de uma boa distribuição de escalas.

Vale ressaltar que a Ribeironense já utiliza uma ferramenta computacional para a confecção das escalas, além do que as características de sua operação são as piores em termos de alocação, pois as linhas são longas e de baixa frequência.

Portanto, o resultado obtido com a implementação da técnica comprova ser eficiente, existindo ainda a possibilidade de melhorar a solução através do refinamento dos ajustes na alocação de frota e parâmetros operacionais.

## CAPÍTULO V

### 5. CONCLUSÕES E RECOMENDAÇÕES

#### 5.1 Conclusões

A técnica proposta neste trabalho possibilita transformar qualquer alocação de frota do transporte urbano de passageiros em dados de entrada adequados para a alocação de tripulação formulada como problema de cobertura de conjuntos. Com isto, pode-se:

Otimizar o processo de alocação de tripulação através da seleção do conjunto de escalas mais adequado para cada problema;

- Utilizar diferentes métodos de alocação de frota, possibilitando inclusive um comparativo entre eles;
- Utilizar diferentes métodos de alocação de tripulação, desde que formulados como problemas de cobertura de conjuntos, possibilitando assim análises de desempenho mediante situações reais e distintas;
- Trabalhar uma gama maior de problemas.

Complementarmente, foi apresentado um descritivo do comportamento da técnica, no qual se revela a influência dos parâmetros de geração e seleção, do conjunto de escalas, no resultado final da otimização. Tal descritivo busca promover o conhecimento do problema, visando com isto possibilitar ajustes.

mais eficazes, de maneira que esta técnica possa ser bem utilizada em situações distintas. Assim, recomenda-se que, no início da resolução de um problema, realize-se teste para verificar o comportamento da geração e seleção das escalas, para só então determinar o conjunto mais adequado.

Por fim, um ensaio comparativo pode comprovar a aplicabilidade da técnica em casos reais, sendo que o resultado obtido ratifica a eficiência do método diante das escalas atualmente utilizadas em casos reais.

Outro aspecto detectado é a possibilidade de se verificar o impacto que mudanças trabalhistas, como uma possível redução de jornada de trabalho, podem trazer nos custos de um sistema de transporte. Com isto é possível fundamentar tecnicamente uma negociação trabalhista, focando assim soluções mais eficientes e rápidas, o que pode evitar aumentos abusivos das tarifas e até mesmo greves da categoria.

## **5.2 Recomendações**

O código desenvolvido apresenta um tempo de processamento razoável para geração de conjuntos com aproximadamente cento e vinte mil colunas, considerando o equipamento utilizado. Porém, a geração de conjuntos maiores demanda tempos proibitivos, não permitindo assim a realização de ensaios para verificar seus efeitos na alocação final. Assim se recomenda um aperfeiçoamento do algoritmo de maneira a gerar maiores conjuntos de escalas com tempos aceitáveis. Sugere-se, neste sentido, o incremento de técnicas de

processamento que tornem o algoritmo mais rápido, como por exemplo, a utilização de ponteiros ao invés de matrizes.

Depois de gerado o primeiro conjunto de escalas, o processo é novamente realizado em cima exclusivamente das viagens não cobertas. Poderia-se modificar o algoritmo de forma que na segunda iteração fossem geradas partições considerando todas as viagens e não somente as não cobertas. Com isto o número de escalas geradas cresceria bastante proporcionando conjuntos com diferentes perfis.



## REFERÊNCIAS BIBLIOGRAFIA

- [BAI00] BAITA, F. et al. (2000). "A comparison of different solution approaches to the vehicle scheduling problem in a practical case". *Computers & Operations Research*, 27 (13), 1249-1269.
- [BAL80] BALL, M. (1980). "A comparison of relaxations and heuristics for certain crew and vehicle scheduling". Presented at National ORSA/TIMS Meeting, Washington DC.
- [BAL83] BALL, M.; BODIN, L.; DIAL, R. (1983). "A matching based heuristic for scheduling mass transit crews and vehicles", *Transportation Science*, 17 (1), 4-31.
- [BEA87] BEASLEY, J. E. (1987). "An algorithm for set covering problems", *European Journal of Operations Research* 31, 85-93.
- [BEA90] BEASLEY, J. E. (1990). "OR-Library: distributing test-problems by electronic mail", *Journal of Operations Research Society* 41(11), pp. 1069-1072.
- [BEA94] BEASLEY, J. E.; CHU, P. C. (1994). "A Genetic Algorithm for Set Covering Problem", *The Management School – Imperial College*.
- [BEN68] BENNET, B.; POTTS, R. (1968). "Rotating roster for a transit system". *Transportation Science* 2(1), 14-34.

- [BOD78] BODIN, L.; ROSENFEILD, D.; KYDES, A. (1978). "UCOST. A micro approach to a transit planning problem". *J. Urban Anal.* 5 (1), 47-69.
- [BOD81] BODIN, L.; ROSENFEILD, D.; KYDES, A. (1981). "Scheduling and estimation techniques for transportation planning". *Computing Ops. Res.* 8, 25-38.
- [BOD83] BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M. (1983). "Routing and scheduling of vehicles and crews – The state of the art," *International Journal of Computers and Operations Research*, Vol 10, N°2, 63-211.
- [CHU95] CHU, P. C.; BEASLEY, J. E. (1995). "A Genetic Algorithm for Set Partitioning Problem", *The Management School – Imperial College*.
- [DAN54] DANZIG, G; FULKERSON, D. R. (1954). "minimizing the number of tankers to meet a fixed schedule", *Naval Res. Logistics Quart.*, Vol 1, 217-222.
- [FER98] FERLAND, J. A.; MICHELON, P. (1988). "The vehicle scheduling problem with multiple vehicle types". *Journal of the Operational Research Society*, 39 (6), 577-583.
- [FOR62] FORD, L. R.; FULKERSON, D. R. (1962). *Flows in Networks*. Princeton University Press.

- [FRE99] FRELING, R.; PAIXÃO, J. M. P.; WAGELMANS, A. P. M. (1999). "Models and algorithms for single depot vehicle scheduling". Rotterdam Institute for Business Economic Studies, Erasmus Universiteit Rotterdam.
- [HEU72] HEURGON, E. (1972). "Um problème de recouvrement: l'habillage des horaires d'une ligne d'autobus", *RAIRO* 6, 13-29.
- [HEU75] HEURGON, E. (1975). "Preparing Duty Roster for Bus Routes by Computers", in D. R. Bergmann and L. Bodin, (eds), "Preprint Workshop on Automated Technique for Scheduling of Vehicle Operators for Urban Public Transportation Services", Chicago
- [HIL81] HILDYARD, P.; WALLIS, N. (1981). "Advances in computer assisted runcuting in North America", in: A. Wren, Eds. *Computer Scheduling of Public Transport*, 183-192. North-Holland, Amsterdam.
- [LEN81] LENSTRA, J.; KAN, Rinnooy (1981). "Complexity of vehile rounting and scheduling problems", *Networks*, 11 (2), 221-227.
- [LEN81] LENSTRA, J; KAN, R. (1981). "Complexity of vehicle routing and scheduling problems", *Networks*, 11 (2), 221-227.
- [LES81] LESSARD, R.; ROSSEAU, J.; DUPUIS, D. (1981). "Hastus I: a mathematical programming approach to the bus driver scheduling problem", in: A. Wren, Eds. *Computer Scheduling of Public*. North-Holland, Amsterdam.

- [MAR74] MARSTEN, R. E. (1974). "An algorithm for large set partitioning problems", *Management Science*, 20(5), 774-787.
- [MAY96] MAYERLE, S. F. (1996). "Um sistema de apoio à decisão para o planejamento operacional de empresas de transporte rodoviário urbano de passageiros". 176f. Tese (Doutorado em Engenharia de Produção)-Programa de Pós-Graduação em Engenharia de Produção, UFSC, Florianópolis.
- [MIT85] MITRA, G.; DARBY-DOWMAN, K. (1985). "CRU-SHED: A Computer Based Bus Crew Scheduling System Using Integer Programming", in: J. M. Rosseau (ed.), *Computer Scheduling of Public Transport 2*.
- [PAR81] PARKER, M.; SMITH, B. (1981). "Two approaches to computer crew scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling", in: A. Wren, Eds. *Computer Scheduling of Public*, 259-268.
- [RYA81] RYAN, D. M.; FOSTER, B. A. (1981). "An Integer Programming Approach to Scheduling", in: A. Wren, Eds. *Computer Scheduling of Public*. North-Holland, Amsterdam.
- [TRA01] TRAMONTIN, M. P. (2001). "Resolução do problema de alocação de veículos — Comparação de algumas técnicas heurísticas". Tese de mestrado, Universidade Federal de Santa Catarina.
- [VOU97] VOUDOURIS, C. (1998). "Guided local search: na illustrative example in function optimisation". *BT Technology Journal*, 16 (3), 46-50.

[WARD81] WARD, R.; DURANT, P.; HALLMAN, A. (1981). "A Problem Decomposition Approach to Scheduling the Drivers and Crews of Mass Transit Systems", in: A. Wren, Eds. *Computer Scheduling of Public*. North-Holland, Amsterdam.

[WRE85] WREN, A.; SMITH, B. M.; MILLER, A. J. (1985). "Complementary Approach to Crew Scheduling", in: J. M. Rosseau (ed.), *Computer Scheduling of Public Transport 2*.

**APÊNDICE: código do algoritmo implementado**

unit Run\_Final;

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics,  
Controls, Forms, Dialogs,
```

```
Grids, StdCtrls, DBGrids, ComCtrls, Db, DBTables,  
Buttons, ExtCtrls;
```

```
type
```

```
reg = record
```

```
dia:string[12];  
grupo:string[2];  
veiculo:integer;  
saida:integer;  
chegada:integer;  
origem:string[20];  
destino:string[20]  
end;
```

```
reg1 = record
```

```
grupo:string[20];  
np1:integer;  
p1:string[20];  
np2:integer;  
p2:string[20];  
np3:integer;  
p3:string[20];  
interv1:integer;  
interv2:integer;  
intervpg:integer;  
hextrapg:integer;  
efic:real;  
end;
```

```
reg2 = record
```

```
viagem:integer;  
escala:integer;  
eficiencia:real {currency};  
end;
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
Form1: TForm1;
```

```
eficl,perchextra,ct30,ct31,ct32,ct33,ct34,vi,k,ct15,ct16,  
ct17,ct18,ct19,w,nvi,i,np,np1,ne,ne1,nef,ve,ff,ff3,ct,ct1,  
ct2,ct3,ct4,ct5,ct6,ct7,ct8,ct9,ct10,ct11,ct12,ct13,ct14,c  
t20,ct21,hextrapg,hextra,intervalo,duraint1,duraint2,int  
ervmin,intervmax,jornada,duraesc,htrab,hpg,vi0:intege  
r;
```

```
DT,di,gr:string;
```

```
tp1,tp2: tdatetime;
```

```
durap:extended; {durap=duração da partição; li=limite  
inferior; ls=limite superior}
```

```
efic,ef,loopli,loops,li,ls:real;
```

```
chegada: array [0..1000000] of integer; {matriz dos  
indices das chegadas das partições}
```

```
saida: array [1..1000000]of integer; {matriz dos  
indices das saidas das partições}
```

```
hchegada: array [1..1000000]of integer; {matriz dos  
horários das chegadas das partições}
```

```
hsaida: array [1..1000000]of integer; {matriz dos  
horários das saidas das partições}
```

```
cheintmin: array [1..1000000]of integer; {matriz dos  
horários das chegadas acrescidos do intervalo}
```

```
cheintmax: array [1..1000000]of integer; {matriz dos  
horários das chegadas acrescidos do intervalo}
```

```
viagem: array [1..1000000] of reg; {matriz das viagens  
a serem realizadas}
```

```
w1: array [1..1000000] of integer;
```

```
escala: array [1..2000000] of reg1;
```

```
escalaf: array [1..2000000] of reg2;
```

```
escalaf1: array [1..2000000] of reg2;
```

```
table1m: array [1..2000000] of reg2;
```

```
p: array [1..1000000] of string[150]; {matriz das  
partições obtidos}
```

```
htrabpi: array [1..2000000] of integer;
```

```
e: array [1..2000000] of string[150]; {matriz das  
partições obtidos}
```

```
custo: array [1..2000000] of variant;
```

```
nefef: array [1..2000000] of integer;
```

```
quantvi: array [1..2000000] of integer;
```

```
quantviF: array [1..2000000] of integer;
```

```
implementation
```

```
{ $R *.DFM } procedure TForm1.FormCreate(Sender:  
TObject);
```

```
BEGIN
```

```
//Limpando a Table1 e a Query1
```

```
with Table1 do
```

```
begin
```

```
Active:= False;
```

```
EmptyTable;
```

```

Active:= True;
end;
query1.close;
query1.open;
//Cabeçalhos dos Stringrids
with StGEfic do
begin
Cells[0,0]:= 'Viagem';
Cells[1,0]:= 'Escala';
Cells[2,0]:= 'Eficiência';
end;
with StGEscGeradas do
begin
Cells[0,0]:= 'Escala';
Cells[1,0]:= 'Grupo';
Cells[2,0]:= 'Primeira Partição';
Cells[3,0]:= 'Segunda Partição';
Cells[4,0]:= 'Terceira Partição';
Cells[5,0]:= 'Int1';
Cells[6,0]:= 'Int2';
Cells[7,0]:= 'IntPG';
Cells[8,0]:= 'Hextras';
Cells[9,0]:= 'Eficiência';
Cells[10,0]:= 'Custo';
end;
with StGQuantvi do
begin
Cells[0,0]:= 'Viagem';
Cells[1,0]:= 'Coberturas';
end;
with StGPartição do
begin
Cells[0,0]:= 'Partição';
Cells[1,0]:= 'Viagens Cobertas';
Cells[2,0]:= 'Htrab';
end;
with StGFiltro1 do
begin
Cells[0,0]:= 'Filtro 1';
Cells[1,0]:= 'Viagem';
Cells[2,0]:= 'Escala';
Cells[3,0]:= 'Eficiência';
end;
with StGFiltro2 do
begin
Cells[0,0]:= 'Filtro 2';
Cells[1,0]:= 'Escala';

```

```

Cells[2,0]:= 'Custo';
Cells[3,0]:= 'Viagens Cobertas';
end;
END;

procedure TForm1.BtCalculaClick(Sender: TObject);
BEGIN
//Inicio da Lógica

//Carregando o Dia Típico
DT:=ComboxDiaT.text;
TBResultado.open;
TBResultado.filter:='DIA = '+''''+DT+'''';
TBResultado.filtered:=true;
TBResultado.first;
nvi:=0;
while not TBResultado.eof do
begin
nvi:=nvi+1;
viagem[nvi].dia:=TBResultadoDIA.value;

viagem[nvi].veiculo:=TBResultadoVEICULO.asinteger;
viagem[nvi].grupo:=TBResultadoGRUPO.value;
viagem[nvi].saida:=TBResultadoSAIDAMIN.asinteger;

viagem[nvi].chegada:=TBResultadoCHEGADA.asinteger;
viagem[nvi].origem:=TBResultadoORIGEM.value;
viagem[nvi].destino:=TBResultadoDESTINO.value;
TbResultado.Next;
end;
TBResultado.first;
tp1:=time;

//Carregando os dados dos Edits
li:=strtoint(edLi.text);
ls:=strtoint(edLs.text);
loopli:=strtoint(edLoopli.text);
loopls:=strtoint(edLoopls.text);
if loopli >= 100 then
loopli:=99;
if loopls <= 0 then
loopls:=1;
jornada:=strtoint(edjornada.text);
intervmin:=strtoint(edintervmin.text);
intervmax:=strtoint(edintervmax.text);
hextra:=strtoint(edhextra.text);

```



```

perchextra:= strtoint(edperc.text);

//Início da Lógica
di:=viagem[1].dia;
ve:=viagem[1].veiculo;
chegada[0]:=0;
np:=1;
ne:=1;
w:=0;
ct21:=1;
ct6:=1;
ct12:=1;
ct11:=1;
ct13:=1;
ct14:=1;
for ct:= 1 to nvi do
  quantvi[ct]:=0;
for ct:=1 to nvi do
  w1[ct]:=0;

while w=0 do
  begin
  //Limpando o StringGrid das viagens não cobertas
  for ct:= 1 to nvi do
    begin
    vi0:=0;
    with StGV10 do
      begin
      Cells[0,ct]:= "";
      Cells[1,ct]:= "";
      Cells[2,ct]:= "";
      Cells[3,ct]:= "";
      Cells[4,ct]:= "";
      Cells[5,ct]:= "";
      Cells[6,ct]:= "";
      Cells[7,ct]:= "";
      end;
    end;

    for i:=1 to nvi do {No loop do while as viagens são
    zeradas, assim i:=1 é zero}
      begin
      //Aplicando a flow formulation
      ff:=i-1;
      if ve <> viagem[i].veiculo then
        ff:=chegada[np-1];
      for ct:=1 to np do
        begin
          if (ff=chegada[ct-1]) or (ff3=chegada[np-1]) then
            begin
            //Comparando o numero do veiculo
            if ve <> viagem[i].veiculo then
              ve:=viagem[i].veiculo;
              for ct1:=i to nvi do
                begin
                  if ve <> viagem[ct1].veiculo then
                    break;
                //Construindo os partições
                durap:=viagem[ct1].chegada- viagem[i].saida;
                if (durap >= li) and (durap <= ls) then
                  begin
                    p[np]:=p[np]+' '+inttostr(ct1);
                    chegada[np]:=ct1;
                    saida[np]:=i;
                    htrabpi[np]:=htrabpi[np] + (viagem[ct1].chegada-
                    viagem[ct1].saida);
                    np:=np+1;
                    p[np]:=p[np-1];
                  end
                  else
                    p[np]:=p[np]+' '+inttostr(ct1);
                    htrabpi[np]:=htrabpi[np] + (viagem[ct1].chegada-
                    viagem[ct1].saida);
                  end;
                //Fim do loop de ct1
                p[np]:= "";
                htrabpi[np]:=0;
                //Fim do loop modo3 da ff
                for ct2:= ct21 to np do
                  begin
                    if saida[ct2]=i then
                      break
                    else
                      ff3:=chegada[ct2-1];
                    end;
                  break;
                //Para o loop de ct
                end;
                //Fim do if da flow formulation
              end;
            //Fim do loop de ct
            end;
            //Fim do loop de i

            //Partições resultados
            for ct:=ct21 to np-1 do

```

```

begin
hchegada[ct]:=viagem[chegada[ct]].chegada;
hsaida[ct]:=viagem[saida[ct]].saida;
cheintmin[ct]:=hchegada[ct]+intervmin;
cheintmax[ct]:=hchegada[ct]+intervmax;
with StGPartição do
begin
//Cells[0,ct]:= floattostr(li)+ ' '+ floattostr(loop);
Cells[0,ct]:= inttostr(ct);
Cells[1,ct]:= p[ct];
Cells[2,ct]:= inttostr(htrabpi[ct]);
end;
end;

//Construindo as Escalas
gr:=viagem[ct21].grupo;
for ct:=ct21 to np-1 do
begin
if gr <> viagem[saida[ct]].grupo then
gr:=viagem[saida[ct]].grupo;
duraesc:= hchegada[ct]-hsaida[ct];
duraint1:=0;
duraint2:=0;
intervalo:=0;
hextrapg:=(duraesc-jornada-intervaiio);
if hextrapg < 0 then
hextrapg:=0;
if (duraesc <= (jornada+intervalo+hextra)) then
begin
htrab:=htrabpi[ct];
hpg:=jornada+hextrapg;
ef:=htrab/hpg;
escala[ne].grupo:= gr;
escala[ne].np1:= ct;
escala[ne].p1:= p[ct];
escala[ne].np2:= 0;
escala[ne].p2:= "";
escala[ne].np3:= 0;
escala[ne].p3:= "";
escala[ne].interv1:= duraint1;
escala[ne].interv2:= duraint2;
escala[ne].intervpg:= intervalo;
escala[ne].hextrapg:= Hextrapg;
escala[ne].efic:= ef;
custo[ne]:= jornada + hextrapg*(1+(perchextra/10));
e[ne]:=p[ct]+' '+p[ct1];
//richedit2.lines.add(inttostr(ne)+' '+p[ct]);
end;
end;

ne:=ne+1;
end;
//Loop do 2º partição
for ct1:=ct+1 to np-1 do
begin
if gr <> viagem[chegada[ct1]].grupo then
break;
duraesc:= hchegada[ct1]-hsaida[ct1];
duraint1:=hsaida[ct1]-hchegada[ct1];
if duraint1>intervmax then
intervalo:= intervmax
else
intervalo:=duraint1;
Hextrapg:=duraesc-jornada-intervalo;
if hextrapg < 0 then
hextrapg:=0;
duraint2:=0;
if (hsaida[ct1] >= cheintmin[ct1]) then
begin
if (duraesc <= (jornada+intervalo+hextra)) then
begin
if viagem[chegada[ct1]].destino =
viagem[saida[ct1]].origem then
begin
hpg:=jornada+hextrapg;
htrab:=htrabpi[ct] + htrabpi[ct1];
ef:=htrab/hpg;
escala[ne].grupo:= gr;
escala[ne].np1:= ct;
escala[ne].p1:= p[ct];
escala[ne].np2:= ct1;
escala[ne].p2:= p[ct1];
escala[ne].np3:= 0;
escala[ne].p3:= "";
escala[ne].interv1:= duraint1;
escala[ne].interv2:= duraint2;
escala[ne].intervpg:= intervalo;
escala[ne].hextrapg:= Hextrapg;
escala[ne].efic:= ef;
custo[ne]:= jornada + hextrapg*(1+(perchextra/10));
e[ne]:=p[ct]+' '+p[ct1];
//richedit2.lines.add(inttostr(ne)+' '+p[ct]+' '+p[ct1]);
end;
end;
ne:=ne+1;
end;
//Loop do 3º partição
for ct2:=(ct1+1) to (np-1) do

```

```

begin
if gr <> viagem[chegada[ct2]].grupo then
break;
duraesc:= hchegada[ct2]-hsaida[ct];
duraint2:=hsaida[ct2]-hchegada[ct1];
if duraint1 > duraint2 then
intervalo:= duraint1
else
intervalo:= duraint2;
if intervalo>intervmax then
intervalo:=intervmax;
Hextrapg:=duraesc-jornada-intervalo;
if hextrapg < 0 then
hextrapg:=0;
if (hsaida[ct2] >= hchegada[ct1]) then
begin
if (duraesc <= (jornada+intervalo+hextra)) then
begin
if viagem[chegada[ct1]].destino =
viagem[saida[ct2]].origem then
begin
hpg:=jomada+hextrapg;
htrab:=htrabpi[ct] + htrabpi[ct1] + htrabpi[ct2];
ef:=htrab/hpg;
escaia[ne].grupo:= gr;
escaia[ne].np1:= ct;
escala[ne].p1:= p[ct];
escala[ne].np2:= ct1;
escala[ne].p2:= p[ct1];
escala[ne].np3:= ct2;
escala[ne].p3:= p[ct2];
escala[ne].interv1:= duraint1;
escaia[ne].interv2:= duraint2;
escala[ne].intervpg:= intervalo;
escala[ne].hextrapg:= Hextrapg;
escala[ne].efic:= ef;
custo[ne]:= jornada +
hextrapg*(1+(perchextra/100));
e[ne]:=p[ct]+' '+p[ct1]+' '+p[ct2];
//richedit2.lines.add(inttostr(ne)+'
'+p[ct1]+' '+p[ct2]);
ne:=ne+1;
end;
// fim do if de origem destino
end;
//fim do if duraesc no ct2
end;
//fim do if hsaida no ct2
end;

end;
//fim do loop ct2
end;
//fim do if duraesc no ct1
end;
//fim do if hsaida no ct1
end;
//fim do lop do ct1
end;
//fim do loop do ct

//Contando a quantidade de cobertura por viagem
for ct:= ct12 to ne-1 do
begin
for ct3:= saida[escala[ct].np1] to
chegada[escala[ct].np1] do
begin
quantvi[ct3]:=quantvi[ct3]+1;
Table1m[ct6].viagem:= ct3;
Table1m[ct6].escala:= ct;
Table1m[ct6].eficiencia:= escala[ct].efic;
{with StGEfic do
begin
cells[0,ct6]:=inttostr(Table1m[ct6].viagem);
cells[1,ct6]:=inttostr(Table1m[ct6].escala);
cells[2,ct6]:=currtostr(Table1m[ct6].eficiencia);
end;}
ct6:=ct6+1;
end;
if escala[ct].np2 <> 0 then
begin
for ct4:= saida[escala[ct].np2] to
chegada[escaia[ct].np2] do
begin
quantvi[ct4]:=quantvi[ct4]+1;
Table1m[ct6].viagem:= ct4;
Table1m[ct6].escala:= ct;
Table1m[ct6].eficiencia:= escala[ct].efic;
{with StGEfic do
begin
cells[0,ct6]:=inttostr(Table1m[ct6].viagem);
cells[1,ct6]:=inttostr(Table1m[ct6].escaia);
cells[2,ct6]:=currtostr(Tabie1m[ct6].eficiencia);
end;}
ct6:=ct6+1;
end;
if escala[ct].np3 <> 0 then
begin

```

```

for ct5:= saida[escala[ct].np3] to chegada[escala[ct].np3] do
begin
quantvi[ct5]:=quantvi[ct5]+1;
Table1m[ct6].viagem:= ct5;
Table1m[ct6].escala:= ct;
Table1m[ct6].eficiencia:= escala[ct].efic;
{with StGEfic do
begin
cells[0,ct6]:=inttostr(Table1m[ct6].viagem);
cells[1,ct6]:=inttostr(Table1m[ct6].escala);
cells[2,ct6]:=currtostr(Table1m[ct6].eficiencia);
end;}
ct6:=ct6+1;
end;
end;
//If do np3
end;
//If do np2
end;

ct21:=np;
ct12:=ne;

//Verificando as viagens não cobertas
ct7:=1;
for ct4:= 1 to nvi do
begin
if quantvi[ct4]=0 then
begin
vi0:=vi0+1;
w1[ct4]:=0;
{with StGVi0 do
begin
Cells[0,ct7]:= inttostr(ct4);
Cells[1,ct7]:= viagem[ct4].dia;
Cells[2,ct7]:= inttostr(viagem[ct4].veiculo);
Cells[3,ct7]:= viagem[ct4].grupo;
Cells[4,ct7]:= inttostr(viagem[ct4].saida);
Cells[5,ct7]:= inttostr(viagem[ct4].chegada);
Cells[6,ct7]:= viagem[ct4].origem;
Cells[7,ct7]:= viagem[ct4].destino;
end;}
ct7:=ct7+1;
end
else
begin
w1[ct4]:=1;
viagem[ct4].dia:= "";
viagem[ct4].veiculo:=0;
viagem[ct4].grupo:= "";
viagem[ct4].saida:=0;
viagem[ct4].chegada:=0;
viagem[ct4].origem:= "";
viagem[ct4].destino:= "";
end;
end;

//Chamando o loop p/ as viagens não cobertas
for ct9:=1 to nvi do
begin
if w1[ct9]=0 then
begin
w:=0;
li:=li*(1-(loopli/100));
ls:=ls*(1+(loopls/100));
break;
end
else
w:=1;
end;
end;
// Fim do while

//Registrando a quantidade de coberturas de cada
viagem pelas Escalas Geradas
for i:= 1 to nvi do
begin
with StGQuantvi do
begin
Cells[0,i]:= inttostr(i);
Cells[1,i]:= inttostr(quantvi[i]);
end;
end;

//Registrando as quantidades finais
for ct:= 1 to ne-1 do
begin
//richedit2.lines.add(escala[ct].efic);
with StGEscGeradas do
begin
Cells[0,ct]:= inttostr(ct);
Cells[1,ct]:= escala[ct].grupo;
Cells[2,ct]:= escala[ct].p1;
Cells[3,ct]:= escala[ct].p2;
Cells[4,ct]:= escala[ct].p3;

```

```

Cells[5,ct]:= inttostr(escala[ct].interv1);
Cells[6,ct]:= inttostr(escala[ct].interv2);
Cells[7,ct]:= inttostr(escala[ct].intervpg);
Cells[8,ct]:= inttostr(escala[ct].hextrapg);
Cells[9,ct]:= floattostr(escala[ct].efic);
Cells[10,ct]:= inttostr(custo[ct]);
end;
end;
Edvi0.text:=inttostr(vi0);
Ednp.text:= inttostr(np-1);
Edne.text:=inttostr(ne-1);
Ednvi.text:= inttostr(nvi);
tp2:=time;
edtp.text:=timetostr(tp2-tp1);
//if savedialog2.execute then
  //richedit2.lines.SaveToFile(savedialog2.filename);
END;

//Filtrando as Escalas
procedure TForm1.BtFiltrarEscClick(Sender: TObject);
begin
tp1:=time;
eficl:=strtoint(edefic.text);
efic:=eficl/100;
k:=strtoint(editk.text);

//Selecionando apenas as de Eficiência >= a efic
desejada ou as K coberturas mais eficientes
ct17:=0;
for ct15:= 1 to ne-1 do
begin
//Eficiencia
if escala[ct15].efic >= efic then
begin
ct17:=ct17+1;
escalaf1[ct17].escala:= ct15;
for ct30:= saida[escala[ct15].np1] to
chegada[escala[ct15].np1] do
begin
nefef[ct30]:=nefef[ct30]+1;
end;
for ct31:= saida[escala[ct15].np2] to
chegada[escala[ct15].np2] do

```

```

begin
nefef[ct31]:=nefef[ct31]+1;
end;
for ct32:= saida[escala[ct15].np3] to
chegada[escala[ct15].np3] do
begin
nefef[ct32]:=nefef[ct32]+1;
end;
end;
end;

//Verificando dentre as escalas filtradas quais viagens
foram cobertas menos vezes que o k
for ct33:= 1 to nvi do
begin
with StGEfic do
begin
cells[0,ct33]:=inttostr(ct33);
cells[1,ct33]:=inttostr(nefef[ct33]);
//cells[2,ct33]:=inttostr(escalaf1[ct33].escala);
end;
if nefef[ct33]<k then
//if quantviF[ct33]<k then
begin
for ct:= 1 to ct6-1 do
begin
if Table1m[ct].viagem = ct33 then
begin
if Table1m[ct].eficiencia < efic then
begin
Table1.append;
Table1Viagem.value:= Table1m[ct].viagem;
Table1Escala.value:= Table1m[ct].escala;
Table1Eficiencia.value:= Table1m[ct].eficiencia;
Table1.post;
end;
end;
end;
end;
end;
query1.close;
query1.open;
Query1.first;
ct10:=0;
while not Query1.eof do
begin
ct10:=ct10+1;
escalaf[ct10].viagem:= Query1Viagem.asinteger;

```

```

escalaf[ct10].escala:= Query1Escala.asinteger;
escalaf[ct10].eficiencia:= Query1Eficiencia.value;
query1.next;
end;
Query1.first;

vi:=escalaf[1].viagem;
for ct15:= 1 to ct10 do
begin
if quantvi[escalaf[ct15].viagem] = 1 then {Quando
a viagem é coberta apenas uma vez}
begin
ct17:=ct17+1;
nefef[escalaf[ct15].viagem]:=1;
escaiaf1[ct17].escala:=
escalaf[ct15].escala;
vi:=escalaf[ct15+1].viagem;
end;
if escalaf[ct15].viagem = vi then
begin
if nefef[escalaf[ct15].viagem] = k then
begin
if escalaf[ct15].viagem <>
escalaf[ct15+1].viagem then
vi:=escalaf[ct15+1].viagem;
end
else
begin
ct17:=ct17+1;
nefef[escalaf[ct15].viagem]:=
nefef[escalaf[ct15].viagem] + 1;
escaiaf1[ct17].escala:=
escalaf[ct15].escala;
vi:=escalaf[ct15+1].viagem;
end;
end; {Fim do else if (if vi)}
end; {Fim do for de ct15}

//Não registrando as repetidas
ct34:=0;
nef:=0;
for ct18:= 1 to ne-1 do
begin
for ct19:= 1 to ct17 do
begin
if escalaf1[ct19].escala = ct18 then
begin
ct34:=ct34+1;
nef:=nef+1;

```

```

with StGFiltro2 do
begin
Cells[0,ct34]:= inttostr(ct34);
Cells[1,ct34]:= inttostr(ct18);
Cells[2,ct34]:= inttostr(custo[ct18]);
Cells[3,ct34]:= e[ct18];
end;
for ct30:= saida[escala[ct18].np1] to
chegada[escala[ct18].np1] do
begin
quantviF[ct30]:=quantviF[ct30]+1;
end;
for ct31:= saida[escala[ct18].np2] to
chegada[escala[ct18].np2] do
begin
quantviF[ct31]:=quantviF[ct31]+1;
end;
for ct32:= saida[escala[ct18].np3] to
chegada[escala[ct18].np3] do
begin
quantviF[ct32]:=quantviF[ct32]+1;
end;
break;
end;
end;
end;
end;
//Registrando os dados finais
{for ct:= 1 to ct17 do
begin
with StGFiltro1 do
begin
Cells[0,ct]:= inttostr(ct);
Cells[1,ct]:= inttostr(escalaf1[ct].viagem);
Cells[2,ct]:= inttostr(escalaf1[ct].escala);
Cells[3,ct]:= currtostr(escalaf1[ct].eficiencia);
end;
end;}

for ct:= 1 to nvi do
begin
with StGFiltro1 do
begin
Cells[0,ct]:= inttostr(ct);
Cells[1,ct]:= inttostr(quantviF[ct]);
end;
end;

```

```

Ednef.text:=inttostr(nef);
tp2:=time;
edtp2.text:=timetostr(tp2-tp1);
ct:=1;
end;

//Salvando as Escalas Filtradas
procedure TForm1.BtSalvarClick(Sender: TObject);
begin
richedit1.lines.add('LINHAS '+inttostr(nvi));
richedit1.lines.add('COLUNAS '+inttostr(nef));
richedit1.lines.add('DADOS');
for ct:= 1 to ct34 do
begin
with StGFiltro2 do
begin
richedit1.lines.add(inttostr(ct)+' '+Cells[2,ct]+' '+Cells[3,ct]);
end;
end;
if savedialog1.execute then
richedit1.lines.SaveToFile(savedialog1.filename);
end;

//Resetando parâmetros das escalas geradas
procedure TForm1.BtReset1Click(Sender: TObject);
begin
for ct:= 1 to ne-1 do
begin
p[ct]:="";
htrabpi[ct]:=0;
escala[ct].grupo:= gr;
escala[ct].np1:= ct;
escala[ct].p1:= p[ct];
escala[ct].np2:= ct1;
escala[ct].p2:= p[ct1];
escala[ct].np3:= ct2;
escala[ct].p3:= p[ct2];
escala[ct].interv1:= duraint1;
escala[ct].interv2:= duraint2;
escala[ct].intervpg:= intervalo;
escala[ct].hextrapg:= Hextrapg;
escala[ct].efic:= ef;
custo[ct]:= jornada + hextrapg*(1+(perchextra/100));
e[ct]:=p[ct]+' '+p[ct1]+' '+p[ct2];

```

```

ne:=0;
Table1m[ct].viagem:= 0;
Table1m[ct].escala:= 0;
Table1m[ct].eficiencia:= 0;
with StGPartição do
begin
Cells[0,ct]:= "";
Cells[1,ct]:= "";
Cells[2,ct]:= "";
end;
with StGQuantvi do
begin
Cells[0,ct]:= "";
Cells[1,ct]:= "";
end;
with StGEscGeradas do
begin
Cells[0,ct]:= "";
Cells[1,ct]:= "";
Cells[2,ct]:= "";
Cells[3,ct]:= "";
Cells[4,ct]:= "";
Cells[5,ct]:= "";
Cells[6,ct]:= "";
Cells[7,ct]:= "";
Cells[8,ct]:= "";
Cells[9,ct]:= "";
Cells[10,ct]:= "";
end;
end;
Edvi0.text:="";
Ednp.text:= "";
Edne.text:="";
edtp.text:="";
end;

```

```

//Resetando parâmetros das escalas filtradas
procedure TForm1.BtReset2Click(Sender: TObject);
begin
nef:=0;
Ednef.text:="";
edtp2.text:="";
richedit1.Lines.Clear;
escalaf[ct].viagem:= 0;
escalaf[ct].escala:= 0;
escalaf[ct].eficiencia:= 0;

```

```
escalaf1[ct].viagem:= 0;
escalaf1[ct].escala:= 0;
escalaf1[ct].eficiencia:= 0;
for ct:= 1 to ct34 do
begin
nedef[ct]:=0;
with StGFiltro1 do
begin
Cells[0,ct]:= "";
Cells[1,ct]:= "";
Cells[2,ct]:= "";
Cells[3,ct]:= "";
end;
with StGFiltro2 do
begin
Cells[0,ct]:= "";
Cells[1,ct]:= "";
Cells[2,ct]:= "";
Cells[3,ct]:= "";
end;
end;
with Table1 do
begin
Active:= False;
EmptyTable;
Active:= True;
end;
query1.close;
query1.open;
end;

end.
```