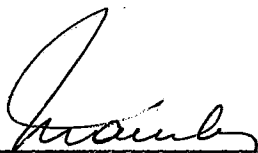


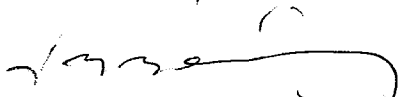
CONTRIBUIÇÕES À SOLUÇÃO DE PROBLEMAS DE PERCURSOS E SEQUENCIAMENTO DE VEÍCULOS

LUIZ SATORU OCHI

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSARIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIENCIAS
EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

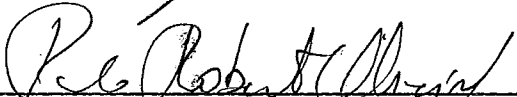
Aprovada por:


Prof. NELSON MACULAN FILHO, D. Habil.
(Presidente)


Prof. JOSÉ MÁRIO MARTÍNEZ, D. Sc.


Prof. MÁRIO VEIGA FERRAZ PEREIRA, D. Sc.


Prof. CELSO CARNEIRO RIBEIRO, D. Ing.


Prof. PAULO ROBERTO OLIVEIRA, D. Ing.

RIO DE JANEIRO - BRASIL

OUTUBRO 1989

OCHI, LUIS SATORU

Contribuições à Solução de Problemas de Percurso e Sequenciamento de Veículos [Rio de Janeiro], 1989.

VII, 489 p. 29.7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 1989).

Tese - Universidade Federal de Rio de Janeiro, COPPE.

1. Problemas de Roteamento de Veículos, Programação Linear Inteira, Relaxações Lagrangeanas, Otimização Combinatória.

I. COPPE/UFRJ II. Título (Serie).

Cii)

À Gatinha e ao Daniel

Aos meus pais.

AGRADECIMENTOS

Desejo expressar os meus agradecimentos às pessoas que me auxiliaram na elaboração deste trabalho, embora reconheça a impossibilidade de atribuir a cada um a importância de sua colaboração.

Mas em particular, gostaria de agradecer ao Prof. Nelson Maculan pela sua eficaz orientação e ao acesso às inúmeras referências bibliográficas propiciando-me a oportunidade de prosseguir e concluir o presente trabalho.

Ao Roberto Becker pelas informações iniciais do problema aplicativo (FP).

Aos colegas do departamento de computação da Universidade Federal Fluminense pelo apoio e compreensão.

À Clarice e ao Evandro pela edição do presente texto.

Finalmente à CAPES e ao CNPq pelo auxílio prestado.

... é muito difícil descobrir uma teoria que descreva todo o universo. Por isso divide-se o problema em diversas partes e inventam-se inúmeras teorias parciais....

STEPHEN W. HAWKING

UMA BREVE HISTÓRIA DO TEMPO

RESUMO DA TESE APRESENTADA À COPPE/UFRJ COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS (D. Sc.)

CONTRIBUIÇÕES À SOLUÇÃO DE PROBLEMAS DE PERCURSOS E SEQUENCIAMENTO DE VEÍCULOS

LUIZ SATORU OCHI

OUTUBRO 1989

ORIENTADOR: NELSON MACULAN FILHO

PROGRAMA DE ENGENHARIA DE SISTEMAS E COMPUTAÇÃO

O problema de percursos e sequenciamento de veículos é atualmente um dos problemas de maior sucesso dentro da Programação Matemática, evidenciado pelo vasto número de publicações sobre o assunto encontrados na literatura.

A origem deste trabalho está ligado a uma proposta de otimizar rotas de navios petroleiros em longo curso com restrições do tipo "time-windows".

Propomos novos procedimentos exatos e aproximados para este problema, envolvendo formulações, relaxações, heurísticas e uma nova técnica de otimização exata.

ABSTRACT OF THESIS PRESENTED TO COPPE/UFRJ AS PARTIAL
FULFILLMENT OF REQUIREMENTS FOR DEGREE OF DOCTOR OF SCIENCE
(D.Sc.)

CONTRIBUTIONS TO THE SOLUTION OF THE ROUTING
AND SCHEDULING OF VEHICLES PROBLEMS

LUIZ SATORU OCHI

OCTOBER 1989

CHAIRMAN: NELSON MACULAN FILHO

DEPARTMENT OF SYSTEMS AND COMPUTING ENGINEERING

Nowadays, the problem of routing and scheduling of vehicles is one of the most distinguished in Mathematical Programming, as we can see from many publications found in specific literature.

The origin of this research is related to the routes optimization for petroleum ship in long haul with time-windows constraints.

We suggest new exact and non-optimal procedures to this problem, involving formulations, relaxations, heuristics and a new exact optimization technique.

ÍNDICE

	Página
CAPÍTULO I: INTRODUÇÃO.....	001
CAPÍTULO II: A TEORIA E AS TÉCNICAS DO PROBLEMA DO ROTEAMENTO E SCHEDULING DE VEÍCULOS : A LITERATURA EXISTENTE.....	011
CAPÍTULO III: O PROBLEMA APLICATIVO (FP).....	208
CAPÍTULO IV: FORMULAÇÕES MATEMÁTICAS PARA O PROBLEMA (FP).....	216
CAPÍTULO V: GERAÇÃO DE LIMITES INFERIORES : NOVAS RELAXAÇÕES E DECOMPOSIÇÕES LAGRANGEANAS.....	266
CAPÍTULO VI: GERAÇÃO DE LIMITE SUPERIOR: UMA NOVA GENERALIZAÇÃO DO ALGORITMO DE CLARKE E WRITE (1964).....	342
CAPÍTULO VII: RELAÇÕES ENTRE DUAS FORMULAÇÕES DISTINTAS DO PROBLEMA (FP).....	371
CAPÍTULO VIII: SOLUÇÃO VIA TÉCNICAS EXATAS: UM NOVO ESQUEMA UTILIZANDO GERAÇÃO DE COLUNAS E PROGRAMAÇÃO DINÂMICA.....	388

CAPÍTULO IX: CONCLUSÕES E PROPOSTAS PARA TRABALHOS	
FUTUROS.....	428
BIBLIOGRAFIA.....	436

CAPITULO I - INTRODUÇÃO

O nosso interesse pelos problemas de roteamento de veículos, surgiu a partir de duas constatações: a primeira está relacionada com o planejamento do setor de transportes de empresas, que em muitos casos, ainda adotam técnicas não muito convincentes na programação da sua frota de veículos, o que onera significativamente os custos dos serviços prestados.

O outro aspecto motivador, foi o de que o problema de achar as melhores rotas é de extrema importância e de difícil solução.

Este problema é importante na medida em que afeta o custo final de transporte e é difícil devido ao elevado número de restrições e de variáveis envolvidas.

A origem deste trabalho está ligado a uma proposta de otimizar rotas de navios petroleiros de médio e grande porte em longo curso.

O problema que denotamos por (FP) refere-se à programação de navios da FRONAPE, órgão ligado ao departamento de transportes da PETROBRÁS.

Recentemente, *BECKER (1987)* descreveu com detalhes o problema (FP), propondo uma solução através de um método de

Programação Linear Inteira, formulado como um modelo de recobrimento com restrições adicionais (MRA) e resolvendo-o via programa MPSX-MIP da IBM.

O problema (FP) (veja *BECKER (1987)*) consiste em um conjunto de contratos de transportes de petróleo e seus derivados para a PETROBRÁS e de minério de ferro para terceiros.

Para o atendimento destes contratos (trips), a FRONAPE dispõe de uma frota de aproximadamente 35 navios heterogêneos entre petroleiros e minero-petroleiros.

Num horizonte de planejamento de curto prazo, (aproximadamente 60 dias), existem cerca de 50 contratos (trips) a serem analisados.

A complexidade deste problema se deve principalmente ao elevado número de restrições e de variáveis envolvidas. Algumas destas restrições se relacionam com os navios, portos de origem e destino de cada trip, tipo de carga de cada trip, velocidade do navio, clima, avaria nos navios, tempo de espera para atracar e/ou carregar e/ou descarregar não previstos, condições do casco do navio, estado da pintura do navio, período de docagem, corrente marítima, tempo limite para concluir um trip, janela de tempo (time-window) para o navio aportar na origem de um trip pronto para operar, restrições de compatibilidade entre navio-porto, navio-carga, navio-rota, navio-bandeira, custos

diferenciados entre dois tipos distintos de navios próprios e afretados, localização atual de cada navio, etc.

Embora o problema de roteamento de veículos seja muito estudado na literatura, com inúmeras aplicações (veja os trabalhos de *BODIN, GOLDEN, ASSAD e BALL (1983)* e *GOLDEN e ASSAD (1988)*), não existem, de nosso conhecimento, muitos trabalhos publicados sobre roteamento de navios com o nível de complexidade do problema (FP).

A maioria absoluta dos trabalhos sobre roteamento se refere a modelos de roteamento sem restrições de tempo e com uma única origem e/ou problemas de roteamento com frota homogênea e/ou problemas de roteamento sem restrições de precedências ou de janelas de tempo.

As aplicações, são na maioria de : problemas de roteamento de veículos urbanos (ônibus escolar, ambulâncias, coleta de lixo), planejamento de vôos de aviões, planejamento de tripulações aéreas, o problema *Dial-a-Ride*, etc.

São raras as aplicações publicadas sobre roteamento de navios, e destes, a maioria se refere ao transporte por cabotagem, efetuado com pequenas embarcações.

Uma exceção é o trabalho desenvolvido recentemente por *BROWN, GRAVES e RONEN (1987)*, (PBGR) onde foi analisado um problema de roteamento de navios petroleiros com restrições

de tempo e com diferentes velocidades para cada navio.

Os autores formularam o problema como um modelo de particionamento elástico (elastic set partitioning model) determinando todas as rotas viáveis numa primeira etapa e obtendo a seguir um conjunto de rotas ótimas, uma para cada navio.

Devido às características próprias do problema, os autores colocam a frota como sendo homogênea, o que reduz a dimensão e as dificuldades do problema.

Embora possua as características do problema (FP), o (PBGR) foi estudado somente para uma frota homogênea usando um procedimento semelhante ao de *BECKER (1987)*, gerando inicialmente o conjunto de rotas viáveis para posteriormente escolher um conjunto de rotas ótimas através de um modelo de recobrimento com restrições adicionais.

A diferença entre eles é que no (PBGR) são analisados também os efeitos de soluções degeneradas e instabilidades numéricas nos modelos de particionamento, propondo modelos generalizados (elásticos) que procuram reduzir estas inconveniências.

Não existem, de nosso conhecimento, trabalhos sobre roteamento de navios do porte do (PBGR) ou (FP) que utilizem técnicas de relaxação e/ou decomposição lagrangeana para viabilizar o uso de técnicas exatas para a solução do

problema.

Para um melhor entendimento deste trabalho, achamos conveniente definir algumas notações e definições preliminares.

Os problemas de percursos de veículos são classificados neste trabalho em três classes:

-O Problema de Roteamento de Veículos (PRV)

-O Problema do Scheduling de Veículos (PSV)

-O Problema de Roteamento e Scheduling de Veículos com restrições time-windows (PRSTW)

O problema de roteamento de veículos (PRV) é um problema totalmente espacial, sem restrições adicionais de tempo e/ou de precedências e tem como objetivo gerar um conjunto de rotas de custo mínimo.

Uma rota por sua vez é definida como uma sequência de tarefas ordenadas no espaço a serem efetuadas por um veículo.

O (PSV) é definido como um problema de roteamento com restrições de tempos fixos e tem como objetivo gerar um conjunto de schedules de custo mínimo.

Um *schedule* é definido como uma sequência de atendimentos ordenados no espaço e no tempo efetuado por um veículo.

Finalmente o (PRSTW) é o problema de roteamento e scheduling combinados com restrições adicionais de janelas de tempo (*time-window*), dentro do qual uma tarefa deve ser iniciada ou concluída.

A característica deste último tipo de problemas é a presença de *variáveis de tempo* numa formulação matemática.

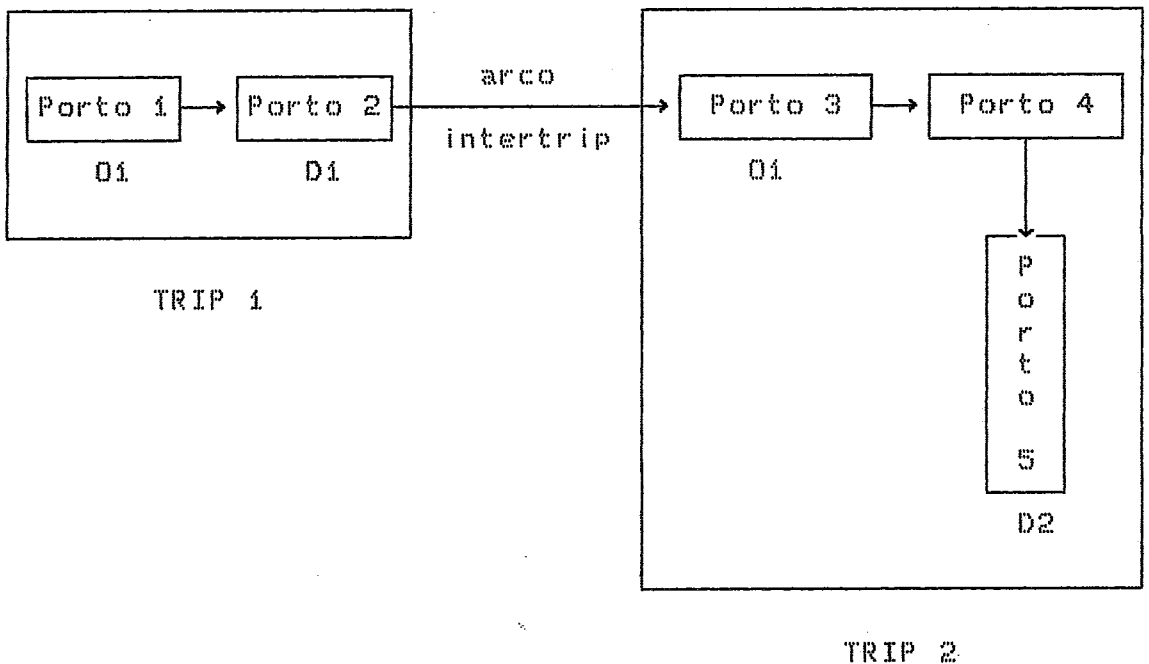
Tanto uma rota como um *schedule* são formados por uma sequência ordenada de *trips* definido por uma ou mais tarefas do problema.

No problema (FP), um *trip i* equivale a um contrato a ser efetuado, normalmente descrito por um local de carregamento O_i (porto origem), e um local de descarregamento D_i (porto de destino) um tempo e um custo de percorrer o arco (O_i, D_i) , um tempo limite para concluir o *trip i* e uma janela de tempo (*time-window*) $[a_i, b_i]$ (com $a_i > 0$ e $a_i < b_i$), dentro do qual os navios alocados ao *trip* devem aportar na sua origem O_i , prontos para operarem.

O trecho caracterizado entre o destino do *trip i* e a origem do *trip j*, subsequente ao *trip i*, é definido como sendo um *arco intertrip* (uma viagem ociosa).

A partir das definições dos trips, estamos em condições de passarmos da rede física original composta pelos nós (portos) e arcos (distância direta entre dois portos) para uma rede transformada (não física) formada pelos trips (novos nós) e pelos arcos intertrips (novos arcos).

Este procedimento que passamos a adotar nas formulações e nos algoritmos reduz sensivelmente as dimensões do problema (FP), como mostra o exemplo dado a seguir:



EXEMPLO DA REDUÇÃO DA REDE TRANSFORMADA

Na figura anterior, a rede original possui 5 nós e 4 arcos e a rede transformada 2 nós e 1 arco.

No desenvolvimento deste trabalho, inicialmente, sintetizamos o atual estado da arte do problema de roteamento e scheduling de veículos e no capítulo 2,

descrevemos resumidamente o problema aplicativo (FP).

Nos capítulos subsequentes, propomos uma teoria e novos algoritmos exatos e aproximados para resolver o problema (FP).

No capítulo 3, propomos sete formulações matemáticas distintas para o problema (FP) acompanhados de suas interpretações em relação ao modelo real.

No capítulo 4, desenvolvemos novas relaxações lagrangeanas e decomposições lagrangeanas para algumas das formulações do capítulo 3 analisamos a sua viabilidade e aplicabilidade bem como a qualidade dos limites inferiores gerados para o problema original.

No capítulo seguinte, apresentamos um novo algoritmo heurístico, baseado nos conceitos de "economia" proposto originalmente por *DANTZG e RAMSER (1959)* e posteriormente por *CLARKE e WRIGHT (1963)*.

A heurística estabelece um novo critério para medir economias num problema que envolve parâmetros conflitantes como custos e tempos, propondo um procedimento de sintetizações à partir do descarte de navios afretados no mercado.

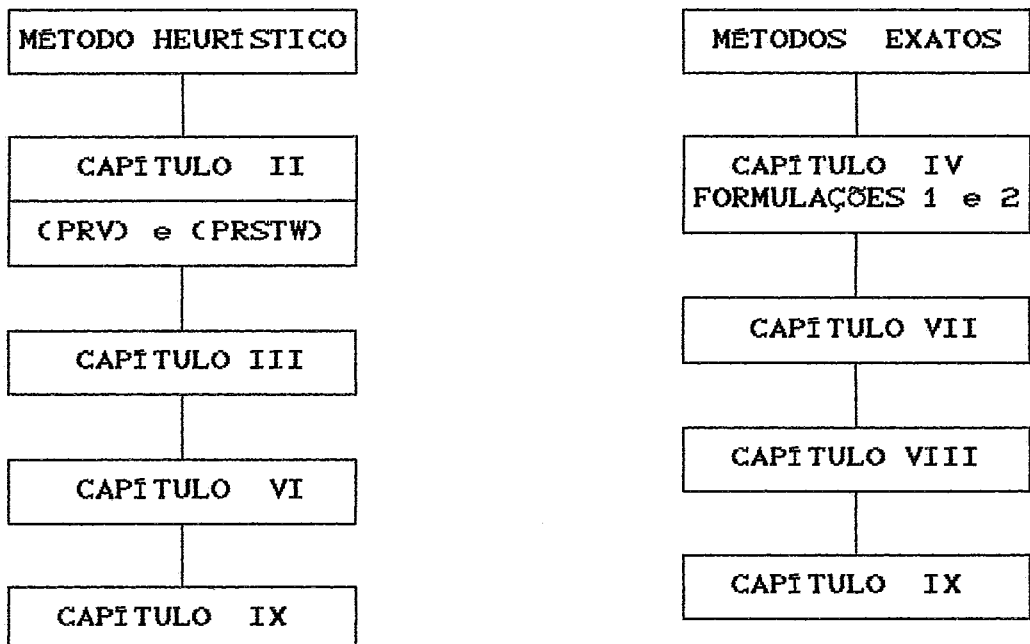
No capítulo 6, apresentamos um estudo teórico comparando a qualidade do limite inferior gerado entre

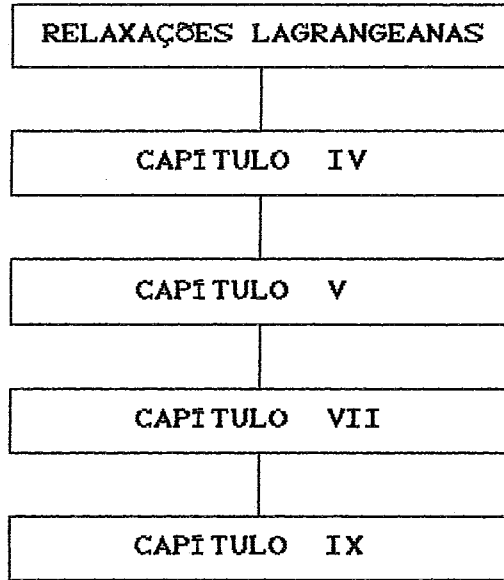
relaxações de duas formulações distintas do problema (FP).

No capítulo 7, propomos um procedimento exato para resolver o problema (FP) baseado num novo esquema de geração de colunas para o problema contínuo, onde o subproblema de geração de colunas é descrito como um conjunto de problemas de caminho mínimo com restrições *time-window* e estes resolvidos através de métodos de programação dinâmica, generalizando o algoritmo desenvolvido por *DESROSIERS, SOUMIS e DESROCHERS (1983)*.

Finalmente, o capítulo 8 apresenta as conclusões e propostas de trabalhos futuros.

Para os leitores com interesse mais específico em : Método Heurístico, Métodos Exatos ou Relaxações Lagrangeanas podemos sugerir a seguinte sequência :





CAPÍTULO II

A TEORIA E AS TÉCNICAS DO PROBLEMA DE ROTEAMENTO
E SCHEDULING DE VEÍCULOS: A LITERATURA EXISTENTE

INTRODUÇÃO

O problema de percurso de veículos é classificado neste trabalho em três classes:

O problema de roteamento de veículos (PRV)

O problema de scheduling de veículos (PSV)

O problema de roteamento e scheduling de veículos com restrições do tipo time-windows (PRSTW).

O (PRV) é um problema totalmente espacial, sem restrições adicionais de tempo e/ou de precedências e tem como objetivo gerar um conjunto de rotas de custo mínimo.

Uma rota por sua vez é definida como uma *seqüência de tarefas ordenadas no espaço* a serem efetuadas por um veículo.

O (PSV) é definido como um problema de roteamento com restrições de tempos fixos; por exemplo, horário de chegada ou de saída de um veículo, tempo máximo para concluir uma tarefa ou uma rota, tempo máximo de espera para

atendimentos, etc..

O objetivo do (PSV) é o de gerar um conjunto de *schedules* de custo mínimo.

Um *schedule* é definido como uma *sequência de atendimentos ordenados no espaço e no tempo*, efetuados por um veículo.

Finalmente os (PRSTW) agrupam os problemas de roteamento e *scheduling* combinados com restrições adicionais de janelas de tempo *time-windows*, dentro dos quais tarefas devem ser iniciadas ou concluídas.

A característica deste tipo de problemas é a presença de *variáveis de tempo*.

O objetivo do (PRSTW) é o de gerar um conjunto de *schedules* de custo mínimo, respeitando as restrições de espaço e de tempo.

A literatura nos mostra que das classes de roteamento de veículos, os (PRSTW) são os mais complexos devido à presença de restrições do tipo *time-windows* que dentre outras dificuldades por ela originadas, está o fato de não podermos mais efetuar um sequenciamento prévio das tarefas a serem executadas sem fixarmos um ponto de cada *time-window*.

De fato, desconhecendo o ponto t_i do *time-window* $[a_i, b_i]$ de uma tarefa i , não podemos predizer o conjunto de candidatos viáveis a suceder a tarefa i em um *schedule*; já que para cada $t_i \in [a_i, b_i]$ podemos gerar um novo conjunto de candidatos a suceder a tarefa i .

Neste capítulo, introduzimos uma sinopse sobre boa parte da literatura existente sobre as três classes de problemas de roteamento de veículos: (PRV), (PSV) e (PRSTW), apresentada numa ordem cronológica.

II.1) O PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)

Apresentamos nesta seção o problema básico de otimização de percursos no espaço, efetuados por uma frota de veículos; também conhecido na literatura como o problema de roteamento puro de veículos ou simplesmente roteamento de veículos (PRV).

O (PRV) é um problema espacial, isto é, ele não incorpora nenhuma restrição de tempo (*scheduling*) e/ou de precedências, e tem como objetivo gerar um conjunto de rotas viáveis a um custo global mínimo.

O (PVR) clássico é descrito por uma rede com uma única origem, de onde cada veículo da frota deve partir com quantidades pré-estabelecidas de um determinado produto, para atender às demandas dos clientes alocados à sua rota, retornando ao final da jornada ao seu ponto de origem.

Esta versão foi inicialmente estudada por *DANTZIG e RAMSER (1959)*, para um problema de distribuição de gasolina.

A partir deste trabalho pioneiro, surgiram centenas de artigos com diversas abordagens para este problema, chegando aos anos atuais como uma das áreas de maior importância dentro da Pesquisa Operacional.

Dentre os problemas de percursos de veículos mais populares podemos citar os seguintes:

P1 - O PROBLEMA DO CAIXEIRO VIAJANTE (PCV)

P2 - O PROBLEMA DO CARTEIRO CHINÊS (PCC)

P3 - O PROBLEMA DOS M-CAIXEIROS VIAJANTES (PMCV)

P4 - O PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)

P5 - O PROBLEMA DE ROTEAMENTO-LOCALIZAÇÃO (PRL)

Listamos a seguir os principais resultados para cada um dos problemas acima, encontrados na literatura, com maior ênfase para o (PCV) e o (PRV).

P1 - O PROBLEMA DO CAIXEIRO VIAJANTE (PCV)

Este é, sem sombra de dúvidas, o mais conhecido dentre todos os problemas de roteamento de veículos.

O (PCV) é um caso particular do (PRV) consistindo de um único veículo (*caixeiro viajante*) com capacidade ilimitada (ou suficiente para atender todos os nós da rede e retornar à sua origem).

Considere uma rede com n nós (cidades), sendo $n-1$ nós destinos e uma origem n_0 . O (PCV) consiste em determinar uma sequência de visitas que o viajante deve efetuar, de modo a passar por cada uma das $n-1$ cidades uma única vez, iniciando e finalizando a viagem na origem n_0 , percorrendo a menor distância total possível.

O nascimento do (PCV) foi pesquisado por vários autores, dentre eles *HOFFMAN e WOLFE (1985)* no livro *O PROBLEMA DO CAIXEIRO VIAJANTE* editado por *LAWLER, LENSTRA, RINNOOY KAN e SHMOYS (1985)*, do qual nos baseamos para sintetizar o surgimento deste problema.

HOFFMAN e WOLFE (1985) acreditam que o nascimento do (PCV) na matemática tenha ocorrido entre os anos de 1931 a 1932, mas eles observaram que há centenas de anos atrás,

EULER (1759) e VANDERMONDE (1771) já discutiam o problema dos movimentos alternativos do cavalo num jogo de xadrez, observando que estes movimentos podem gerar ciclos hamiltonianos na rede formada pelo tabuleiro de xadrez.

Isto pode ser visto como um princípio do problema de gerar um ciclo hamiltoniano de distância mínima (PCV).

Quanto ao termo *caixeiro viajante*, *HOFFMAN e WOLFE (1985)* citam uma pesquisa desenvolvida por *FLOOD (1955)* que se refere a um livro alemão editado em (1832) intitulado *O caixeiro viajante : como deveria ser e como proceder para obter lucros e ser bem sucedido nos negócios: por um viajante veterano*. O autor deste livro já sugeria a obtenção de boas rotas evitando visitas repetidas a uma mesma cidade.

Na vasta literatura existente sobre o (PCV) em nível de *survey* podemos sugerir, dentre outros, os trabalhos de: *BELLMORE e NEMHAUSER (1974)*, *HELD, HOFFMAN, JOHNSON e WOLFE (1984)* *CHRISTOFIDES (1973)*, *BODIN, GOLDEN, ASSAD, BALL (1983)*; além do livro sobre o (PCV) editado por *LAWLER, LENSTRA, RINNDROY KAN e SHMOYS (1986)*.

Devido à sua complexidade ser não polinomial o (PCV) é NP-completo (veja *KARP (1972)*; *GAREY e JOHNSON (1979)* e *PAPADIMITRIOU (1977)*), muitos dos (PCV) são resolvidos aproximadamente utilizando-se de procedimentos heurísticos.

Na solução via técnicas exatas, normalmente o (PCV) é formulado inicialmente como um Problema de Programação Linear Inteira (PPLI).

Descrevemos, a seguir, algumas formulações existentes para o (PCV) descritos como (PPLI).

D) FORMULAÇÃO CLÁSSICA DO (PCV)

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2.1)$$

$$\text{S.A. } \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (2.3)$$

$$x = (x_{ij}) \in S \quad (2.4)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, n, \quad j = 1, \dots, n \quad (2.5)$$

onde S são restrições que evitam a formação de subrotas na solução do (PCV).

As restrições (2.1), (2.2) e (2.3) descrevem um problema de alocação linear (PAL), enquanto (2.4) são restrições para evitar a formação de sub-rotas (rotas

desconexas) na solução do (PCV) e finalmente (2.5) são as variáveis binárias zero-um do problema.

Existem na literatura várias formulações para o conjunto S de (2.4) e dentre eles, podemos citar os três mais conhecidos:

$$S_1 = \{(x_{ij}) \text{ tal que } \sum_{i \in P} \sum_{j \notin P} x_{ij} \geq 1, \forall \text{ subconjunto próprio}$$

$$\text{e não-vazio } P \subset N = \{1, 2, \dots, n\}\} \quad (2.6)$$

$$S_2 = \{(x_{ij}) \text{ tal que } \sum_{i \in P} \sum_{j \in P} x_{ij} \leq |P| - 1, \forall \emptyset \neq P \subseteq$$

$$\subseteq \{2, 3, \dots, n\}\} \quad (2.7)$$

$$S_3 = \{(x_{ij}) \text{ tal que } z_i - z_j + nx_{ij} \leq n - 1, 2 \leq i \neq j \leq n,$$

$$\text{para algum } z_i \in \mathbb{R}\} \quad (2.8)$$

Cada um dos conjuntos S_i , $i = 1, 2, 3$ acima, tem por objetivo evitar a formação de rotas desconexas na solução do (PCV), S_1 por exemplo, nos diz que todos os pontos de um subconjunto $P \subseteq N$ deve ser ligado aos pontos de $N-P$.

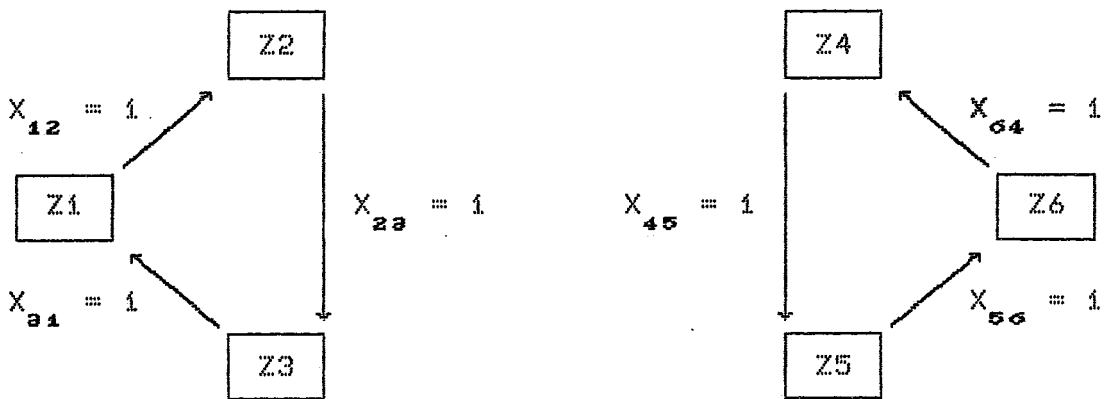
S_2 requer que uma solução do (PCV) não deve conter ciclos excluindo a origem.

Finalmente, S_3 é um procedimento de também evitar a

formação de sub-rotas, desenvolvido por MILLER, TUCKER e ZEMLIN (1960) e ao contrário de S_1 e S_2 exige uma atenção maior para a sua compreensão.

Assim suponha o seguinte exemplo de solução desconexa.

Aplicando S_3 a esta solução desconexa, teremos:



$$\begin{aligned}
 & z_4 - z_5 + 6 x_{45} \leq 5 \\
 + & z_5 - z_6 + 6 x_{56} \leq 5 \\
 & z_6 - z_4 + 6 x_{64} \leq 5
 \end{aligned}$$

$$18 \leq 15 \quad (\text{absurdo})$$

Note que na formulação (PCV1) descrito por (2.1) a (2.5), podemos permutar as restrições (2.2) e (2.3) pelas restrições:

$$\sum_{j=2}^n x_{1j} = 2 \quad (2.9)$$

$$\sum_{j=1}^n x_{ij} + \sum_{k=1}^n x_{ki} = 2, \quad i = 1, \dots, n \quad (2.10)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} = n \quad (2.11)$$

As restrições (2.9) e (2.10) nos dizem que cada nó(cidade) deve ter grau (número de arcos incidentes) igual a 2 enquanto (2.11) garante um total de n arcos na solução do (PCV).

Deste modo, se substituirmos (2.2) e (2.3) por (2.9), (2.10) e (2.11), obtemos uma nova formulação (PCV2).

ii) (PCV2)

$$\left\{ \begin{array}{l} \text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \end{array} \right. \quad (2.12)$$

$$\left\{ \begin{array}{l} \text{S.A.} \quad \sum_{j=2}^n x_{1j} = 2 \end{array} \right. \quad (2.13)$$

$$\left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} + \sum_{k=1}^n x_{ki} = 2, \quad i = 1, \dots, n \end{array} \right. \quad (2.14)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^n \sum_{j=1}^n x_{ij} = n \end{array} \right. \quad (2.15)$$

$$\left\{ \begin{array}{l} X = (x_{ij}) \in S \end{array} \right. \quad (2.16)$$

$$\left\{ \begin{array}{l} x_{ij} \in \{0, 1\}, \quad i = 1, \dots, n \text{ e } j = 1, \dots, n \end{array} \right. \quad (2.17)$$

III) (PCV) COM VARIÁVEIS DE FLUXO

Esta formulação foi desenvolvida por *GAVISH e GRAVES (1978)* e tem como novidade a introdução de variáveis de fluxo y_{ij} .

A interpretação dos fluxos $y_{ij} \geq 0, \forall i, \forall j$ pode ser assim entendida: supomos inicialmente o problema de despachar $n-1$ unidades de fluxo da origem n_0 , onde cada um dos $n-1$ nós destinos tem demanda unitária.

$$\text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2.18)$$

$$\text{S.A.} \quad \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (2.19)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (2.20)$$

$$\sum_{j=1}^n y_{ij} - \sum_{j=1}^n y_{ji} = -1, \quad i = 2, \dots, n \quad (2.21)$$

$$y_{ij} \leq M x_{ij}, \quad i = 1, \dots, n \text{ e } j = 1, \dots, n \quad (2.22)$$

$$x_{ij} \in (0, 1), \quad y_{ij} \geq 0 \quad i = 1, \dots, n \text{ e } j = 1, \dots, n \quad (2.23)$$

onde $M \geq n - 1$.

Os conjuntos das restrições (2.18), (2.19) e (2.20) descreve o problema de alocação linear (PAL) e (2.21) nos diz que em cada nó destino i , deve ficar exatamente uma unidade de fluxo.

Além disso, note que (2.21) proíbe a formação de sub-rotas na solução do (PCV).

As restrições (2.22) exigem que se o arco (i, j) não for usado ($x_{ij} = 0$) então o fluxo neste arco deve ser nulo ($y_{ij} = 0$).

Finalmente (2.23) são as variáveis de decisão x_{ij} e as variáveis de fluxo $y_{ij} \geq 0$ do problema.

IV) (PCV4)

No caso simétrico, podemos formular o (PCV) com variáveis arco (x_i) ao invés de variáveis do tipo nó-arco (x_{ij}).

$$\left. \begin{array}{l} \text{Minimizar} \\ \sum_{i=1}^n c_i x_i \end{array} \right\} \quad (2.24)$$

$$\left. \begin{array}{l} \text{S.A.} \\ \sum_{i=1}^n x_i = n \end{array} \right\} \quad (2.25)$$

$$\sum_{i \in k_t} x_i \geq 2, \forall k_t \equiv (P_t, \bar{P}_t), \text{ com } P_t < N \quad (2.26)$$

$$\text{e } \bar{P}_t = N - P_t$$

$$x_i \in \{0, 1\}, i = 1, \dots, n \quad (2.27)$$

$$\text{onde } x_i = \begin{cases} 1, & \text{se o arco } i \text{ for utilizado} \\ 0, & \text{c/c.} \end{cases}$$

e c_i = custo de percorrer o arco i

Uma formulação alternativa para o (PCV4) é obtida ao substituirmos as restrições (2.25) e (2.26) pelas restrições equivalentes:

$$\sum_{i \in k_t} x_i \geq 1, \forall k_t \equiv (P_t, \bar{P}_t), \text{ com } P_t < N \quad (2.28)$$

e

$$\sum_{i \in A_j} x_i = 2, j = 1, 2, \dots, n \quad (2.29)$$

v) FORMULAÇÃO (PCV5)

$$\text{Minimizar } \sum_{i=1}^n c_i x_i \quad (2.30)$$

$$\text{S.A. } \sum_{i \in k_t} x_i \geq 1, \forall k_t \equiv (P_t, \bar{P}_t) \text{ com } P_t < N \quad (2.31)$$

$$\sum_{A_j} x_i = 2, \quad j = 1, \dots, n \quad (2.32)$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, n \quad (2.33)$$

vi) O (PCV) - PERIÓDICO: (PCV6)

Esta é uma versão do (PCV), onde se utilizam os n períodos de tempo.

Considere c_{ijt} = custo de percorrer diretamente o trecho entre i e j no período de tempo t . ~~de de tempo t.~~

$$x_{ijt} = \begin{cases} 1, & \text{se o arco } (i, j) \text{ é usado no período } t. \\ 0, & \text{c/c.} \end{cases}$$

$$\text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^n c_{ijt} x_{ijt} \quad (2.34)$$

$$\sum_{j=1}^n \sum_{t=1}^n x_{ijt} = 1, \quad i = 1, \dots, n \quad (2.35)$$

$$\sum_{i=1}^n \sum_{t=1}^n x_{ijt} = 1, \quad j = 1, \dots, n \quad (2.36)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ijt} = 1, \quad t = 1, \dots, n \quad (2.37)$$

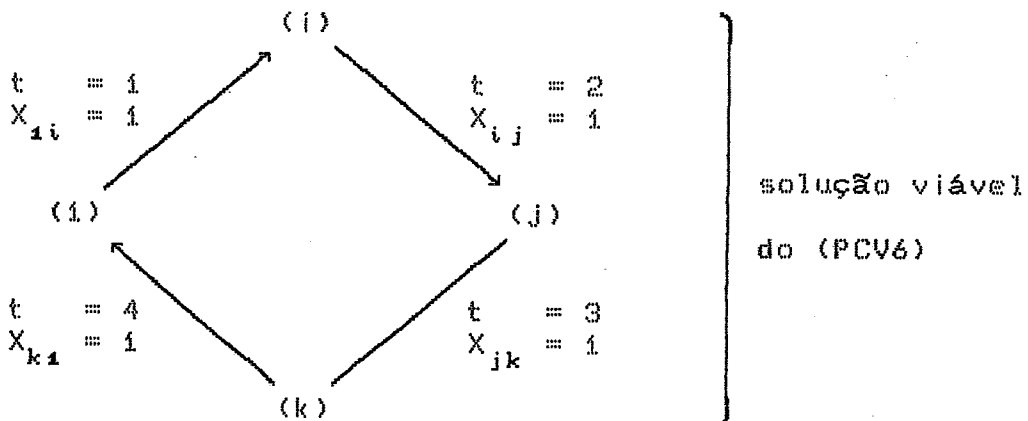
$$\sum_{j=1}^n \sum_{t=2}^n t x_{ijt} - \sum_{j=1}^n \sum_{t=1}^n t x_{jtt} = 1, \quad i = 2, \dots, n \quad (2.38)$$

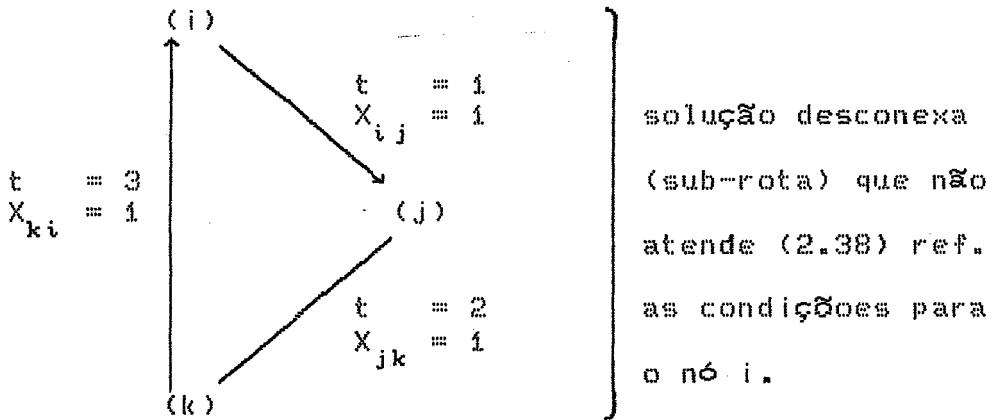
$$x_{ijt} \in \{0, 1\}, \quad i = 1, \dots, n, \quad j = 1, \dots, n \text{ e } t = 1, \dots, n \quad (2.39)$$

Supomos no (PCV6) que a origem do (PCV) é o nó $(i=1)$.

As restrições (2.38) nos dizem que o período de tempo que se chega em i é o i° período de tempo, anterior ao período que se sai do nó i , para $i = 2, 3, \dots, n$.

Isto evita a formulação de sub-rotas, como mostramos no exemplo dado a seguir:





Um resultado muito importante para reduzir o número de restrições do (PCV6), de $4n - 1$ para n , foi mostrado por *FOX, GAVISH e GRAVES (1980)*, com a simples substituição das restrições (2.35), (2.36) e (2.37) do (PCV6) pelas restrições.

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^n x_{ijt} = n \quad (2.40)$$

A prova da equivalência entre (2.35) - (2.37) e (2.40) pode ser encontrada em *GARFINKEL (1985)*.

Uma outra formulação do (PCV) periódico foi desenvolvido por *PICARD e QUEYRANNE (1978)*, para a programação de uma máquina e cuja formulação é uma extensão de uma formulação anteriormente proposta por *HADLEY (1964)*.

Considere $x_{ijt} = \begin{cases} 1, & \text{se existe transição do estado } i \\ & \text{para estado } j \text{ no tempo } t \\ 0, & \text{c/c.} \end{cases}$

$$\begin{aligned} \text{Minimizar} \quad & \sum_{j=1}^n c_{ojo} x_{ojo} + \sum_{t=1}^n \sum_{i=1}^n \sum_{j=1}^n c_{ijt} x_{ijt} + \\ & + \sum_{j=1}^n c_{j,n+1,n} x_{j,n+1,n} \end{aligned} \quad (2.41)$$

S.A.

$$\sum_{j=1}^n x_{ojo} = 1 \text{ onde } x_{ojo} = \sum_{\substack{k=1 \\ k \neq j}}^n x_{jk1}, \quad j = 1, \dots, n \quad (2.42)$$

$$\begin{aligned} \sum_{\substack{i=1 \\ i \neq j}}^n x_{ijt} &= \sum_{\substack{k=1 \\ k \neq j}}^n x_{j,k,t+1}, \quad t = 1, \dots, n-2, \\ & j = 1, \dots, n \end{aligned} \quad (2.43)$$

$$\sum_{\substack{i=1 \\ i \neq j}}^n x_{i,j,n-1} = x_{j,n+1,n}, \quad j = 1, \dots, n \quad (2.44)$$

$$x_{ojo} + \sum_{\substack{i=1 \\ i \neq j}}^n \sum_{t=1}^n x_{ijt} = 1, \quad j = 1, \dots, n \quad (2.45)$$

$$\begin{aligned} x_{ijt} \in (0, 1), \quad i = 1, \dots, n, \quad j = 1, \dots, n, \\ t = 1, \dots, n \end{aligned} \quad (2.46)$$

Listamos a seguir, uma síntese de parte da literatura existente sobre o (PCV), isto é, o (PCV) espacial, *numa ordem cronológica*. Trabalhos sobre o Problema dos

M-Caixeiros viajantes e (PCV) com restrições adicionais de tempo serão mencionadas nos itens posteriores deste capítulo.

Dentre os trabalhos pioneiros sobre o (PCV), podemos citar os artigos de *Dantzig, Fulkerson e Johnson (1964)* e *Flood (1955)*.

Kruskal (1956) analisou a relação existente entre o (PCV) e uma sub-árvore geradora mínima e *Croes (1958)* desenvolveu uma técnica de relaxação para o (PCV), resolvendo inicialmente um problema de alocação linear (PAL) e posteriormente buscando soluções que formem rotas viáveis para o (PCV).

Bellman (1962) introduziu conceitos de Programação Dinâmica para resolver um (PCV) com até 17 cidades. Ainda nesse ano, *Hardgrave e Nemhauser (1962)* mostraram a relação entre o (PCV) e o problema do caminho mais longo (PCML), concluindo que o (PCV) é um caso especial do (PCML).

Duas formulações do (PCV), a clássica e a versão com múltiplas visitas foram analisadas e reduzidas a um (PCML).

Held e Karp (1962) sugeriram a solução de vários problemas, dentre eles o (PCV) via Programação Dinâmica, observando que estes problemas possuem uma estrutura que permite a sua solução por esquemas recursivos do tipo usado

na Programação Dinâmica.

Little, Murty, Sweeney e Kare (1963) desenvolveram uma solução por partes para o (PCV), usando Programação Linear e Branch-and-Bound. Já *Karg e Thompson (1964)* apresentaram um procedimento heurístico para resolver o (PCV). Neste mesmo ano, *Gilmore e Gomory (1964)* descreveram um problema de sequenciamento de tarefas de uma máquina como um (PCV).

An Engineering Approach to the (TSP) é o título de um trabalho que descreve um método proposto por *ROBERTS e B.FLORES (1966)*, adequado, segundo seus autores, para os profissionais *não-matemáticos* e consiste basicamente de uma sequência de operações de forma: gerar circuitos iniciais, melhorar circuito, e determinar sequências de cidades (nós) da solução final.

NICHOLSON (1968) apresentou um novo método aproximado que resolve de forma eficiente o (PCV). Um ano após, surgiu outro *survey* sobre o (PCV), de *ISAAC e E. TURBAN (1969)* que basicamente complementou um *survey* anterior de *BELLMORE e NEMHAUSER (1968)*.

E.BELLMORE e J.C.MALONE (1971), apresentaram dois algoritmos para o (PCV) assimétrico, resolvendo inicialmente um (PAL) e efetuando a seguir, se necessário, estudos paramétricos para gerar uma solução do (PCV).

NICOS CHRISTOFIDES (1972), analisou a qualidade dos limites inferiores obtidos para o (PCV) e apresentou um novo procedimento iterativo para obter limites mais eficientes. Nos vários exemplos numéricos obteve uma aproximação média de 4,7% do valor ótimo para (PCV) - simétrico e 3,8% para (PCV)- não simétrico, isto com um algoritmo cuja complexidade não difere muito dos algoritmos para resolver o (PAL). Um outro algoritmo cuja complexidade é aproximadamente igual ao requerido pelo (PAL) foi desenvolvido por *LAWLER (1971)*, para um caso particular do (PCV) onde $d_{ij} = 0$ para $i \geq j$.

KROLAK, FELTS e ARBLE (1971), propuseram um algoritmo heurístico que usa um tempo computacional muito reduzido, em tempo real para (PCV) com 100 a 200 cidades.

Um dos resultados mais brilhantes sobre o (PCV) foi obtido por *Held e Karp (1971) e (1973)*, ao relacionar o (PCV) com o problema da árvore geradora mínima.

Outro "survey" sobre os procedimentos heurísticos para o (PCV) é apresentado por *Rosenkrantz, Stearns e Lewis (1977)*.

Uma nova heurística para o (PCV) simétrico foi apresentado por *Lin e Kernighan (1973)*, onde o algoritmo, segundo os autores, obteve solução ótima para todos os problemas testes com até 110 cidades, e com um tempo de

execução de aproximadamente n^2 (n é o número de cidades).

Uma versão melhorada do algoritmo anteriormente proposto por *Bellmore e Malone (1971)* foi desenvolvido por *Garfinkel (1973)*.

Uma modificação do algoritmo de *Held e Karp (1973)* para o (PCV) simétrico foi proposta por *Hansen e Krarup (1974)* e, segundo seus autores, é 25 vezes mais rápida que o algoritmo original.

Outra heurística para o (PCV) foi construído por *Wiorowski e McElvain (1975)*. Nesta heurística a prioridade é o tempo de computação necessário para se gerar uma solução aproximada do (PCV).

Golden (1977), sugere um procedimento heurístico que fornece estimativas estatísticas do comprimento da solução do (PCV).

Miliotis (1976) e (1978) resolve o (PCV) via simplex e algoritmos de planos de corte ou branch-and-bound.

Burness e White (1976) apresentam o "problema de localização do caixeiro viajante" que é uma variante do problema clássico de localização. Neste caso, a nova facilidade será o ponto inicial e final do caixeiro viajante.

Uma outra heurística que resolve o (PCV) em tempo polinomial foi proposto por *Christofides (1976)*.

Norback e Love (1977) apresentaram um outro procedimento geométrico para resolver sequencialmente o (PCV).

A prova de que o (PCV) euclidiano é NP-completo, foi demonstrado por *Papadimitriou (1977)* e neste mesmo ano, *Bazaraa e Goode (1977)* publicaram um trabalho sobre o problema dual do (PCV), onde o problema dual é resolvido via métodos do tipo subgradientes e Branch-and-bound.

Smith e Thompson (1977) propuseram outro algoritmo de enumeração implícita do tipo (LIFO) para o (PCV) simétrico usando a relaxação em i -árvore proposto por *Held e Karp (1971)*.

Comparativamente com o algoritmo de *Held e Karp (1971)* este algoritmo, segundo seus autores, se mostrou mais econômico na quantidade de memória requerida. São mostradas experiências computacionais onde os autores exibem exemplos que mostram casos de superioridade sobre o algoritmo original.

Smith, Srinivasan e Thomson (1977) apresentaram três novos algoritmos de enumeração implícita para resolver o

(PCV) assimétrico usando técnicas semelhantes aos de *Eastman (1958)* e de *Bellmore e Malone (1971)* para o problema da eliminação de sub-rotas. São apresentadas experiências computacionais em problemas com até 200 nós (cidades).

Cornuejols e Nemhauser (1978) adaptaram o algoritmo proposto por *Christofides (1976)* obtendo melhoramentos no seu pior caso.

Garfinkel e Gilbert (1978) efetuaram uma análise probabilística para vários (PCV) gerados aleatoriamente e mostraram que o valor ótimo pode ser bem aproximado por uma função beta.

Uma outra relaxação linear do (PCV) foi sugerida por *Svestra (1978)*, onde todas as soluções viáveis são circuitos hamiltonianos.

Cruyssen e Rijckaert (1978) desenvolveram um novo algoritmo heurístico que é uma generalização do método de *Webb (1971)*. Ele é sequencial, no sentido de que a cada passo, o algoritmo gera um nova ligação entre duas cidades, com uma complexidade global da ordem de $O(n^{1.82})$.

Papadimitriou e Steiglitz (1978) mostraram através de vários exemplos, possíveis dificuldades do (PCV) simétrico e assimétrico.

Um (PCV) estocástico onde as distâncias entre cidades não é determinística foi sugerido por *Leipala (1978)*.

Gupta (1978) propôs um novo algoritmo de busca lexicográfica e decisões sequências para a solução do (PRV).

Karp (1979) resolveu um (PCV) usando uma relaxação para transformar o problema original num problema de alocação linear (PAL). Posteriormente usou um procedimento para gerar rotas viáveis para o (PCV). O tempo de execução do algoritmo se mostrou semelhante ao tempo gasto por outros algoritmos de (PAL).

Smith (1980) desenvolveu um outro algoritmo de enumeração implícita (LIFO) baseado no algoritmo de *Held e Karp (1971 e 1973)*. O autor concluiu que em dois aspectos o procedimento de busca usando a relaxação i-árvore é inferior à relaxação do (PCV) num (PAL). O primeiro é o fator tempo de computação que é maior na arborescência mínima e o segundo para o (PCV) assimétrico, onde os limites das duas relaxações são iguais, beneficiando portanto um procedimento mais econômico como a relaxação em (PAL).

Outros resultados computacionais para o (PCV) simétrico de 15 a 318 cidades são mostrados no trabalho de *PADBERG e HONG (1980)*.

Um procedimento clássico para resolver o (PCV) foi

apresentado por *BALAS e CHRISTOFIDES (1981)* que apresentaram um novo algoritmo para a versão simétrica, usando inicialmente relaxações lagrangeanas para converter o (PCV) num (PAL). Limites superiores são obtidos através de heurísticas e finalmente um novo método de Branch and Bound foi usado para obter soluções viáveis.

Uma versão generalizada do (PCV) foi estudada por *LAPORTE, NOBERT e PELLENTIER (1980)*. O trabalho consiste em encontrar o ciclo hamiltoniano de custo mínimo entre n conjuntos de nós (cidades), usando um esquema de eliminação de sub-rotas.

JONKER, DE LEVE, DERVERD e VOLGENANT (1980) desenvolveram um novo algoritmo do tipo *duas rotas*, para o (PCV) simétrico e não simétrico.

KANNELLAKIS e PAPANIMITRIOU (1980) apresentaram uma extensão do algoritmo de busca de *KIM-KERNIGHAN (1973)* para o (PCV) assimétrico.

GOLDEN, BODIN, DOYLE e STEWART (1980) desenvolveram um estudo sobre a qualidade e complexidade dos limites gerados por heurísticas para o (PCV), concluindo que uma solução a 2 ou 3% do ótimo pode ser obtido com um algoritmo de complexidade de $O(n^3)$, onde n é o número de cidades do (PCV).

GROTSCHEL (1980) propôs um algoritmo para o (PCV) de grande porte usando o conhecimento das facetas do poliedro associado ao (PCV).

CROWDER e PADBERG (1980) propuseram um algoritmo exato para o (PCV) simétrico, resolvendo problemas com até 318 cidades. O algoritmo usa planos de cortes e brand-and-bound.

Outra variante do (PCV) clássico foi descrito por *CUTLER (1980)*, denominado *N-Line Planar TSP*. Neste problema os pontos estão sobre N linhas e para N pequeno a solução foi obtida via Programação Dinâmica.

FRIEZE, GALBIATI e MAFFIOLI (1982) apresentaram um estudo sobre o pior caso de várias heurísticas para o (PCV) assimétrico com desigualdade triangular. Os autores também propuseram uma nova heurística de complexidade $O(\lceil \log_2 n \rceil)$ para a versão assimétrica.

Outra versão do (PCV) com o objetivo de encontrar um ciclo de comprimento mínimo passando por cada cidade ao menos uma vez, foi analisada por *CORNUEJOLS, FONLUP e NADDEF (1985)*.

Nesta mesma linha, *COSMADAKIS e PAPANIMITRIOU (1984)* apresentaram outro algoritmo desenvolvido para resolver o (PCV) com poucos nós (5 ou 6), mas que devem ser visitadas centenas e centenas de vezes. O algoritmo é exponencial no

número de cidades mas logarítmica no número de visitas.

Um outro *survey* sobre o (PCV) foi apresentado por *Held, Hoffman, Johnson e Wolpe (1984)*.

A versão do (PCV) com muitas visitas, foi também estudada por *Szwarcfiter (1985)*, onde foi proposto um novo algoritmo inspirado num trabalho de *Dantzig, Fulkerson e Johnson (1954)*.

Um outro método exato usando cortes foi proposto por *Fleischmann (1985)* que resolveu problemas com até 120 cidades.

Laporte, Mercure e Nohert (1985) apresentaram algoritmo exatos para o (PCV)-generalizado e assimétrico com o objetivo de gerar um circuito hamiltoniano mínimo entre n grupos de cidades. O problema foi formulado como um problema de Programação Linear inteira e resolvido via técnicas de relaxação lagrangeana e Branch-and-Bound. Resultados computacionais com até 100 nós e 8 grupos foram apresentados.

Por outro lado, *Korner (1986)* analisou as relações entre várias heurísticas para o (PCV) de minimização e maximização (maior circuito hamiltoniano).

Jeromin e Korner (1985) propuseram heurísticas para

obter limites do (PVC) simétrico generalizado, isto é, o (PCV) com algumas restrições adicionais que dividem as cidades em grupos e cada grupo de cidade deve ser visitado contiguamente.

Padberg e Rinaldi (1987) resolveram um dos maiores exemplos documentados do (PCV) simétrico, usando técnicas de Branch-and-Bound e cortes. Foram resolvidos exemplos com até 532 cidades.

Mais recentemente, *Arthur e Freudeney (1988)* desenvolveram uma técnica de gerar exemplos do (PCV) no qual a rota ótima já é previamente conhecida. Como atrativo este procedimento não estabelece limites de dimensões para o (PCV). Testes empíricos mostraram que os problemas resultantes são tão difíceis quanto os problemas gerados aleatoriamente.

Finalmente *B. Fleischmann (1988)* sugeriu uma nova classe de planos de cortes para o (PCV) simétrico.

P2 O PROBLEMA DO CARTEIRO CHINÊS (PCC)

O (PCC) tem por objetivo cobrir todos os arcos da rede minimizando a distância total percorrida.

Quanto a característica da rede, ela pode ser direcionada, não-direcionada ou mista. Nos dois primeiros casos, o (PCC) possui uma complexidade polinomial, enquanto no caso misto, o (PCC) é NP-árduo.

Entre as várias aplicações possíveis para o (PCC) podemos citar: o problema da coleta do lixo urbano, o problema do percurso de um carteiro, o problema da distribuição de gás em botijões nas residências, etc.

De forma geral, a solução do (PCC) pode ser vista como uma generalização de uma rota de EULER (ciclo Euleriano).

Uma rota de *EULER* é um caminho contínuo, percorrendo cada arco do grafo uma única vez, sendo que a sua existência nos grafos direcionados e não direcionados foi inicialmente mostrada por *EULER (1736)* que afirmou que no caso não direcionado, a condição necessária e suficiente para a existência de uma rota de *EULER* é a de que o grau de cada nó seja par. Assim, se o grafo contém nós com grau ímpar, só podemos cobrir todos os arcos percorrendo alguns arcos mais

de uma vez.

MEI-KO-KWAN (1962) descreveu o problema de minimizar a distância total necessária para cobrir todos os arcos de uma rede contendo nós ímpares (rede não euleriana) e a partir deste trabalho, este problema passou a ser reconhecido como Problema do Carteiro Chinês (PCC), cuja formulação básica na versão não direcionada é a seguinte:

$$\left\{ \begin{array}{l} \text{Minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \end{array} \right. \quad (2.47)$$

$$\left\{ \begin{array}{l} \text{S.A.} \quad \sum_{k=1}^n x_{ki} - \sum_{k=1}^n x_{ik} = 0, \quad i=1, \dots, n \end{array} \right. \quad (2.48)$$

$$\left\{ \begin{array}{l} x_{ij} + x_{ji} \geq 1, \text{ para todo arco } (i,j) \in A \end{array} \right. \quad (2.49)$$

$$\left\{ \begin{array}{l} x_{ij} \geq 0 \text{ e inteiro, } i=1, \dots, n \\ \qquad \qquad \qquad j=1, \dots, n \end{array} \right. \quad (2.50)$$

onde A = conjunto de arcos da rede

x_{ij} = o número de vezes que o arco (i,j) é percorrido

n = número de nós da rede

c_{ij} = custo de percorrer o arco (i,j)

Listamos a seguir, alguns trabalhos sobre o (PCC) encontrados na Literatura; numa ordem cronológica.

Um dos trabalhos pioneiros sobre o (PCC) foi desenvolvido por *BELLMAN e COOKE (1969)*, que apresentaram um algoritmo de programação dinâmica para o (PCC). Os resultados computacionais mostraram um bom desempenho do algoritmo, notadamente em exemplos com dimensões reduzidas.

EDMONDS e JOHNSON (1973) apresentaram uma solução para o (PCC) usando a teoria de Matching. O algoritmo proposto é polinomial no tempo e é uma generalização do algoritmo *b-Matching blossom* desenvolvido por estes mesmos autores.

BELTRAMI e BODIN (1974) desenvolveram um algoritmo exato para o caso direcionado e uma heurística para o caso direcionado e capacitado.

Um novo algoritmo polinomial para o (PCC) não direcionado foi proposto por *CHRISTOFIDES (1974)*.

FREDERICKSON (1979) apresentou vários algoritmos aproximados para a solução de vários problemas de roteamentos, dentre eles, o (PCC) direcionado e misto.

O autor analisa o pior caso de vários algoritmos inclusive o apresentado por *EDMONDS e JOHNSON (1973)*.

STERN e DROR (1979) analisaram o problema do Carteiro Chinês (PCC) com restrições adicionais de comprimento

(tempo) máximo de rotas (M-PCC). São apresentadas aplicações práticas relacionadas com a área de energia.

MINIEKA (1979) descreveu um *survey* sobre o (PCC) totalmente direcionados ou totalmente simétricos.

Além disso, foi proposto uma nova técnica para resolver o (PCC) misto, na presença de nós com grau ímpar.

GOLDEN e WONG (1981) definiram um problema de roteamento com arcos capacitados, conhecido como (PCC) capacitado, gerando várias formulações matemáticas e efetuando análises de complexidade para tais problemas.

Ainda neste arco, *GOLDEN, DeARMON e BAKER (1981)* desenvolveram várias heurísticas para o (PCC) capacitado apresentando ainda alguns resultados computacionais.

MOON (1982) considerou o (PCC) numa rede com as peculiaridades de rede urbana.

Para cada rede $N = (V, E)$, onde E é o conjunto de nós e V o de segmentos de ruas ligando dois nós, os autores analisam o problema de como repetir a passagem por segmentos de ruas de modo a minimizar a distância total percorrida num (PCC).

Muitos autores resolveram este problema como um

problema de *MATCHING*, quando N não é direcionado.

Para os casos direcionados ou mistos, com todos os nós de grau par, costuma-se usar o conceito de fluxo de custo mínimo.

A partir destas observações, o autor explorou as características da sua rede e desenvolveu um novo algoritmo simples que encontra rapidamente uma solução ótima para o (PCC).

PEARNS (1988) analisou dois procedimentos para obter limites inferiores para o (PCC) capacitado, a saber o *Matching Lower Bound (MLB)* e o *Node Scanning Lower Bound (NSLB)* e introduz um novo procedimento que produz limites inferiores muito eficazes.

A qualidade do limite inferior obtido, segundo o seu autor, é igual ou melhor aos obtidos por (MLB) e (NSLB).

P3 O PROBLEMA DOS M CAIXEIROS VIAJANTES (PMCV)

O problema dos M caixeiros viajantes é uma extensão do (PCV) clássico já que agora existem M viajantes para visitar as n cidades, todos saindo e retornando de uma única cidade origem n_0 . Uma restrição adicional no (PMCV) na maioria dos casos exige que todos os M viajantes devem passar ao menos por uma cidade destino, ou seja o número de Caixeiros a ser utilizado é fixo e igual a M.

Todas as restrições do (PCV) são mantidas, isto é, cada cidade deve ser visitada exatamente uma única vez por um único viajante, não existem restrições de capacidade para os M viajantes e o objetivo no (PMCV) é o de determinar M percursos viáveis, sendo uma para cada viajante, de menor distância global possível.

As formulações matemáticas descritas para o (PCV) podem ser estendidas facilmente para o (PMCV), cuja formulação clássica é a seguinte:

$$\left\{ \begin{array}{l} \text{Minimizar} \quad \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \end{array} \right. \quad (2.51)$$

$$\left\{ \begin{array}{l} \text{S.A.} \quad \sum_{i=0}^n x_{ij} = \begin{cases} M, & \text{se } j = 0 \\ 1, & \text{se } j = 1, \dots, n \end{cases} \end{array} \right. \quad (2.52)$$

$$\sum_{j=0}^n x_{ij} = \begin{cases} M, & \text{se } i = 0 \\ 1, & \text{se } i = 1, \dots, n \end{cases} \quad (2.53)$$

$$X = (x_{ij}) \in S \quad (2.54)$$

$$x_{ij} \in \{0, 1\}, \quad i = 0, \dots, n, \quad j = 0, \dots, n \quad (2.55)$$

Na formulação (2.51) - (2.55), as restrições (2.51) a (2.53) descrevem um (PT) se $x_{ij} \geq 0$. Enquanto (2.54) são restrições para eliminar sub-rotas e (2.55) são as variáveis binárias do problema.

A formulação inicial do (PMCV) como problema de Programação Linear Inteira se deve a *MILLER, TUCKER e ZEMLIN (1960)*.

O termo *extensão* ou *generalização* do (PCV), usado para o (PMCV), pode na verdade ser substituído por expressões do tipo *equivalentes*, já que um (PMCV) pode ser convertido num (PCV) e vice-versa, como mostramos a seguir.

No (PMCV) podemos criar $M-1$ cópias da origem $0 = O_1$ que, juntamente com a já existente, formam M origens O_1, O_2, \dots, O_m .

O procedimento é o seguinte: ligar cada uma das M origens $O_i, i = 1, \dots, M$ com os nós destinos, exatamente como na rede original e proibir a ligação entre dois

depósitos (origens). Se o grafo for completo podemos colocar por exemplo, a distância entre duas origens superior a

$$\sum_{i=1}^n \sum_{j=1}^n |d_{i,j}|$$

Um exemplo com $M = 2$ seria da forma:

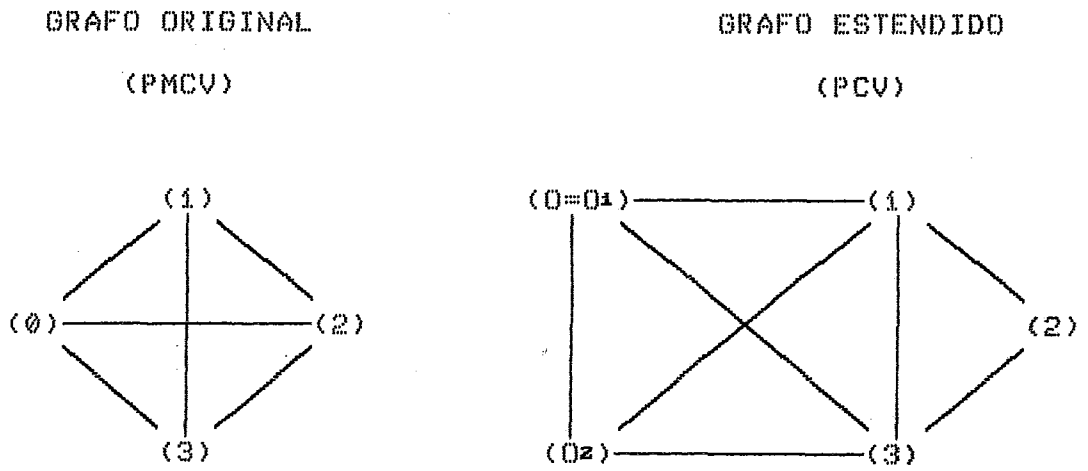


FIGURA 2.1

Uma solução da Figura (2.1) seria:

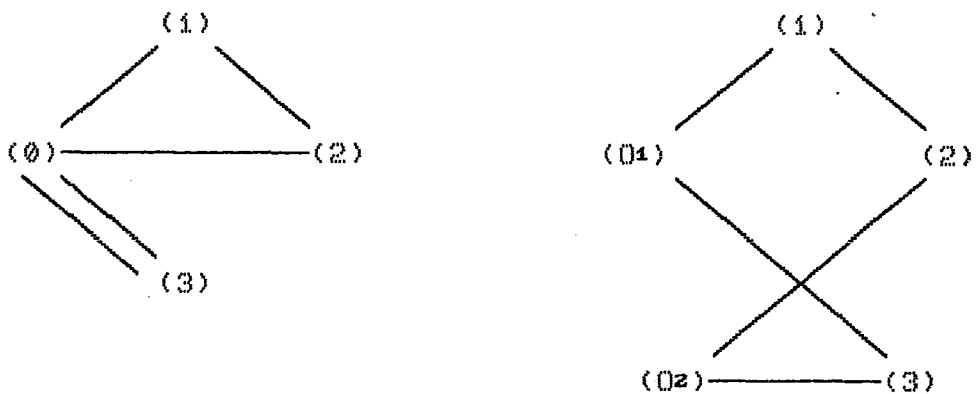


FIGURA 2.2

Na figura 2.2 note que as soluções do (PMCV) e do (PCV) são equivalentes se interpretarmos as origens O_1 e O_2 como sendo cópias uma da outra.

A equivalência do (PCV) com o (PMCV) foi mostrada independentemente por vários autores: *BELLMORE e HONG (1974)*, *ORLOFF (1974)*, (neste último trabalho, foram desenvolvidos outros resultados que foram posteriormente corrigidos por *LENSTRA e RINNOOY KAN (1976)*) e finalmente *SVESTKA e HUCKFELDT (1973)*.

O (PMCV) abrange uma variedade de problemas clássicos, por exemplo, quando $M = 1$, teremos o (PCV), por outro lado, se no (PMCV) acrescentarmos restrições de capacidade para cada viajante, teremos o problema do despacho de veículos que será visto posteriormente neste trabalho.

Sintetizamos a seguir, em ordem cronológica, os principais resultados sobre o (PMCV) espacial encontrados na literatura, enfatizando que os (PMCV) com restrições adicionais de tempo e/ou de precedências serão analisadas no item Problemas de *Scheduling*.

Uma nova formulação para o (PMCV) foi proposta por *SVESTRA e HUCKFELD (1973)* que segundo os seus autores se mostrou superior às outras formulações encontradas na literatura.

A solução foi obtida usando técnicas de Branch-and-Bound, acoplados com algoritmos para gerar limites inferiores e superiores.

A formulação proposta permite que certas sub-rotas satisfaçam o conjunto de restrições, reduzindo com isso, o trabalho de busca. Esta formulação é na verdade uma generalização da apresentada por *MILEER, TUCKER e ZEMLIN (1960)* para o (PCV) e é descrita da forma:

$$\text{Minimizar } \sum_{i=1}^r \sum_{j=1}^r d_{ij} x_{ij} \quad (r = m+n-1) \quad (2.56)$$

$$\text{S.A.} \quad \sum_{i=1}^r x_{ij} = 1, \quad j = 1, \dots, r \quad (2.57)$$

$$\sum_{j=1}^r x_{ij} = 1, \quad i = 1, \dots, r \quad (2.58)$$

$$y_i - y_j + \left\lceil \frac{n}{m} \right\rceil x_{ij} \leq \left\lceil \frac{n}{m} \right\rceil + m - 2, \quad \forall i \neq j \quad (2.59)$$

$$\text{com } i, j \in I_0 = (1, 2, \dots, m), \dots, (n^2 - 3n + 2)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, \forall j \quad (2.60)$$

onde $\lceil x \rceil$ denota o maior inteiro, menor ou igual a x .

As distâncias d_{ij} são definidas a partir das distâncias originais c_{ij} do (PCV) da seguinte forma:

Aumentar a matriz (c_{ij}) com $m-1$ novas linhas e colunas, onde cada nova linha e cada nova coluna é uma cópia da primeira linha e primeira coluna de (c_{ij}) respectivamente, supondo que a primeira (linha e coluna) correspondem a origem do problema do Caixeiro viajante.

oo	cópias Linha 1	= (d_{ij})
cópias coluna 1	(C_{ij})	

FIGURA 2.3

BELLMORE e HONG (1974) mostraram também que o (PMCV) pode ser escrito como um (PCV) extendendo o grafo original.

A idéia básica consiste em criar $M-1$ nós adicionais que representam cópias da única origem do grafo original.

HONG e PADBERG (1977) demonstraram que um (PMCV) simétrico com n cidades é equivalente a um (PCV) simétrico com $n+M+1$ cidades (se retirarmos a exigência de todos os M caixeiros devem visitar cidades) ou $n+M$ (se fixarmos o número M de caixeiros utilizados).

Já *RUSSEL (1977)* apresentou uma nova heurística para o

(PMCV) com restrições adicionais. A heurística é uma extensão do algoritmo anteriormente proposto por *LIN e KERNIGHAN (1973)* para o (PCV) possuindo uma complexidade da ordem de $O(N^{2.2})$, onde N é o número de cidades do problema.

BERENGUER (1979) desenvolveu matrizes de distâncias atualizadas para o (PMCV) através de transformações lineares viáveis. As atualizações são, na verdade, vários critérios para obter as melhores rotas para o (PMCV).

Um caso particular de transformação é a de *CLARKE e WRIGHT (1963)* para o problema do despacho de veículos (veja P4). Após analisar vários critérios, o autor afirma que as únicas viáveis são obtidas adicionando constantes o_i e o_j para as linhas i e colunas j da matriz original de distâncias D , multiplicadas por um escalar λ (λD).

LENSTRA e RINNOOY KAN (1979) esclareceram numa nota, algumas dúvidas de resultados teóricos de *BERENGUER (1979)*.

LAPORTE e NOBERT (1980) analisaram vários problemas de roteamento formulados como um (PMCV) simétrico com M fixo ou variável. Os autores concluíram também que as técnicas encontradas na literatura para transformar um (PMCV) num (PCV) com M fixo não são vantajosas devido à elevação das dimensões e da característica do (PCV) resultante. Um procedimento alternativo foi proposto usando relaxações no (PMCV) e supondo ser M um número variável.

No caso simétrico, a formulação matemática seria da forma:

$$\text{Minimizar } \sum_{i < j} c_{ij} x_{ij} + fM \quad (2.61)$$

$$\text{S.A. } \sum_{j=2}^n x_{1j} = 2M \quad (2.62)$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2, \quad k = 2, \dots, n \quad (2.63)$$

$$\sum_{i < k} x_{ij} \leq |S| - 1, \quad (3 \leq |S| \leq n - 2);$$

$$S \subseteq \{2, \dots, n\} \quad i, j \in S \quad (2.64)$$

$$x_{ij} \in \{0, 1\} \quad \forall 1 < i < j$$

$$\quad \forall j = 2, \dots, n \quad (2.65)$$

$$M \geq 1 \text{ e inteiro (se for variável)} \quad (2.66)$$

Onde o (PMCV) é constituído de $n-1$ cidades destinos e $n=1$ denota a origem.

f é o custo fixo por cada viajante utilizado.

M é o número de caixeiros viajantes utilizados (fixo ou variável).

O problema (2.61) a (2.66) foi resolvido usando uma técnica semelhante ao desenvolvido por *MILLIOTIS (1978)* obtendo um resultado tão bom quanto à solução do (PCV) transformado quando M é variável.

Com M fixo, os autores observaram uma nítida superioridade do seu algoritmo frente aos métodos que transformam o (PMCV) num (PCV), obtendo economias de até 88%.

RAO (1980) demonstrou que a transformação desenvolvida por *BELLMORE e HONG (1974)* para converter n um (PMCV) n em n um (PCV) no caso não simétrico, pode ser estendido também ao caso simétrico, resultando num (PCV) simétrico com apenas $n + M - 1$ cidades.

LAPORTE, NOBERT e MERCURE (1980) estudaram o problema dos m caixeiros viajantes localizados em p diferentes origens ($p \leq m$) o problema foi definido como um (p, m) (PCV), e no caso particular onde $p = 1$, teremos $(1, m)$ (PCV) \equiv (PmCV).

O problema (p, m) (PCV) foi resolvido via relaxações para reduzir o elevado número de restrições existentes neste tipo de problema, mesmo quando o número n de cidades for relativamente pequeno, como mostra a formulação matemática dado a seguir:

NOTAÇÕES:

Considere: $N = \{1, \dots, n\}$ o conjunto de cidades destinos

$K = \{1, \dots, p\}$ o conjunto de origens

X_{ij} = número de vezes que o arco (i, j) é usado por um caixeiro no sentido (i, j) ou (j, i)

X_{kl} não é definido se k e $l \notin K$.

$X_{ij} = 2$ significa que o caminho entre i e j foi usado 2 vezes (de $i \rightarrow j$ e de $j \rightarrow i$) formando uma rota simples.

m_k = número de caixeiros localizados na origem $k \in K$, onde $\sum_{k \in K} m_k = m$

\bar{m}_k = limite superior de m_k , $\forall k \in K$.

$$\left\{ \begin{array}{l} \text{Minimizar} \quad \sum_{i, j \in N} c_{ij} x_{ij} \end{array} \right. \quad (2.67)$$

$$\text{S.A.} \quad \sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2 m_k, \quad k \in K \quad (2.68)$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2, \quad k \in N - K \quad (2.69)$$

$$\sum_{\substack{i, j \in S \\ i < j}} x_{ij} \leq |S| - 1, S \subset N, S \cap K = \emptyset \quad (2.70)$$

$$\sum_{t=1}^{|T|-1} x_{i_t, i_{t+1}} \leq |T| - 2, T \subset N, |T| \geq 3 \quad (2.71)$$

$$\begin{array}{l} i_t, i_{t+1} \in T \\ i_t < i_{t+1} \end{array} \quad |T \cap K| = 2$$

$$\left. \begin{array}{l} x_{ij} \in \{0, 1\} \quad i, j \in N-K \\ \\ i \in K, j \in N-K \end{array} \right\} \quad (2.72)$$

$$1 \leq m_k \leq \bar{m}_k, \quad k \in K \quad (2.73)$$

As restrições (2.68) e (2.69) indicam que o nó k deve ter grau a $2m_k$ se $k \in K$ ou igual a 2 caso contrário.

As condições (2.70) são barreiras para eliminar sub-rotas não contendo nenhuma origem.

(2.71) são restrições de acoplamento de barreiras.

Estas restrições proíbem a ligação de duas origens e assegura que o viajante não visita mais de uma origem.

DISCENZA (1981) apresentou uma nova versão do método desenvolvido anteriormente por *HONG e PADBERG (1977)* para

transformar um (PMCV) num (PCV).

Neste trabalho, o autor afirma que dos quatro nós adicionais gerados na transformação de *HONG e PADBERG (1977)*, dois podem ser eliminados sem maiores prejuízos.

Uma outra opção apresentada pelo autor é substituir os quatro nós adicionais por um par de restrições simples da forma:

$$\left[\begin{array}{l} \sum_{i=1}^m x_{n, n+1} = 1 \\ e \\ x_{n+m-1, n+m} + x_{n, n+m} = 1 \end{array} \right.$$

onde n = número de nós destinos e m = número de viajantes.

Concluindo, *DISCENZA (1981)* mostrou a equivalência das soluções de todas estas formulações.

GARCIA-DIAZ (1985) introduziu uma nova metodologia que representa o (PMCV) como um modelo de circulação numa rede e a seguir, fornece um procedimento para eliminar soluções com sub-rotas.

A solução ótima do problema de circulação pode não ser

viável para o (PMCV) devido a possíveis sub-rotas na solução, desta forma, o autor sugere um procedimento iterativo para reduzir o número de sub-rotas da solução.

TRINH (1987) descreveu dois novos algoritmos exatos para o (PMCV), sendo um baseado no modelo de fluxos de custo mínimo e o outro, usando transformações do (PMCV) para um (PCV) assimétrico. Em ambos os casos, desenvolveu-se técnicas específicas do tipo Branch-and-Bound para sua solução.

P4 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)

O problema clássico de roteamento de veículos (PRV) tem sido intensamente divulgado na literatura nas duas últimas décadas com trabalhos teóricos e aplicativos.

Também conhecido na literatura como o problema do despacho de veículos (PDV) ou como o problema de otimização de percursos de distribuição (POPD); ele se caracteriza como um problema totalmente espacial, isto é, não envolve nenhuma restrição de tempo (scheduling) e/ou de precedência.

O (PRV) é descrito normalmente como um problema de roteamento nos nós de uma rede e cujo objetivo é satisfazer às demandas destes nós através de uma frota de veículos localizados num único (ou vários) depósito(s) de origem por onde cada veículo deve sair e retornar ao final da sua jornada, a um menor custo global possível.

O (PRV) é um nome genérico dado a uma variedade de classes de problemas, envolvendo a visita a um conjunto de clientes através de uma frota de veículos. Um dos casos particulares mais populares do (PRV) é o problema dos M caixeiros Viajantes (PMCV) visto anteriormente em P3.

Salientamos novamente que os problemas de roteamento de veículos envolvendo restrições adicionais de tempo e/ou de precedência serão discutidos nos itens posteriores deste capítulo.

Existem na literatura vários trabalhos que enfocam o *estado da arte* do (PRV), dentre eles podemos citar os seguintes: CELESTINO (1973); CHRISTOFIDES MINGOZZI e TOTH (1979); BODIN e GOLDEN (1981); BODIN, GOLDEN, ASSAD e BALL (1983); CERDEIRA e COELHO (1984); GOLDEN e ASSAD (1986); e o livro editado por GOLDEN e ASSAD (1988).

Apresentamos a seguir, uma sinopse dos principais trabalhos sobre o (PRV) encontrados na literatura *na ordem cronológica*.

O enfoque em alguns casos se torna mais amplo com formulações e/ou resumo das técnicas utilizadas.

Tal procedimento pode ser parcialmente justificado devido a possíveis familiaridades deste com o enfoque dos capítulos posteriores deste trabalho.

Apesar das limitações das técnicas de otimização na época, GARVIN, CRANDALL, JOHN e SPELLMAN (1957) já relatavam aplicações de programação linear na área de produção e distribuição de derivados do petróleo.

Um dos trabalhos pioneiros sobre o (PRV) foi apresentado por *DANTZIG e RAMSER (1959)*.

O modelo apresentado pelos autores, consiste de uma frota homogênea de veículos que devem satisfazer às demandas de n destinos a partir de uma única origem, por onde os veículos devem sair e retornar ao final da sua jornada.

A capacidade dos veículos foi suposta menor que a quantidade total a ser demandada nos n destinos que por sua vez devem ser atendidos numa única visita.

RESUMO DO ALGORITMO DE DANTZIG E RAMSER (1959)

Inicialmente aloca-se cada destino numa viagem separada, obtendo n rotas simples (origem - destino - origem).

Esta solução inicial normalmente é inviável na prática já que o número de veículos é quase sempre menor que o número de destinos.

Para viabilizar o número de veículos necessários, efetua-se a cada estágio r , síntetizações entre duas viagens de modo a reduzir o tamanho da frota. As síntetizações devem respeitar as restrições de capacidade dos veículos, condição

esta descrita pela proibição de sintetizações entre pares de destinos cuja soma das suas demandas supere $\frac{C}{2^{N-r}}$, onde C = capacidade do veículo, N é o número máximo de estágios permitidos e r é o estágio atual.

Assim, garante-se ao final de n-1 estágios que a capacidade do veículo estará no máximo saturada.

As sintetizações entre duas viagens são determinadas à partir da distância economizada s_{ij} ao ligar diretamente o destino i com o destino j, onde

$$s_{ij} = d_{i0} + d_{j0} - d_{ij}$$

Se $s_{ij} < 0 \rightarrow$ vale a pena conectar i com j

Se $s_{ij} \geq 0 \rightarrow$ não vale a pena conectar i com j

BALINSKI e QUANDT (1964), formularam o problema do despacho de veículos como um modelo de particionamento (set partitioning model) da forma:

$$\left\{ \begin{array}{l} \text{Minimizar} \quad \sum_{j=1}^n c_j x_j \quad (2.74) \\ \text{S.A.} \quad \sum_{j=1}^n a_{ij} x_j = 1, \quad i = 1, 2, \dots, m \quad (2.75) \\ x_j \in \{0, 1\}, \quad j = 1, 2, \dots, n \quad (2.76) \end{array} \right.$$

onde $x_j = \begin{cases} 1, & \text{se a rota } j \text{ for usada} \\ 0, & \text{c/c.} \end{cases}$

A matriz do (PP) $A = (a_{ij})$ é uma matriz (zero-um)

onde cada componente $a_{ij} = \begin{cases} 1, & \text{se a atividade } i \text{ pertence a} \\ & \text{rota } j \\ 0, & \text{c/c.} \end{cases}$

e finalmente $c_j =$ custo da rota j

Os autores utilizaram o método de cortes de *GOMORY* (1958) para obter uma solução ótima inteira, e procurou-se comprovar a eficiência deste procedimento através de vários testes computacionais.

Sabemos que a grande dificuldade deste tipo de modelo (particionamento ou recobrimento) aplicado a um problema real é o elevado número de colunas (rotas) viáveis existentes; número que pode crescer exponencialmente com o número de nós destinos.

Assim, na prática costuma-se utilizar apenas uma parte das colunas viáveis do (PP) escolhidas através de algum critério de dominância e/ou pelo descarte de rotas através de informações prévias do problema.

Um dos artigos de maior sucesso sobre o problema do despacho de veículos foi apresentado por *Clarke e Wright* (1964).

Nele, os autores propuseram um algoritmo heurístico construído à partir do algoritmo proposto anteriormente por *Dantzig e Ramser* (1959) para o problema do despacho de uma frota de veículo, a partir de um depósito central.

A idéia do novo algoritmo surgiu a partir da observação de que o método proposto por *Dantzig e Ramser* (1959) se preocupa mais em saturar a capacidade dos veículos do que em minimizar a distância (custo) total necessário para atender todas as demandas.

Assim, a proposta foi a de substituir as restrições de capacidade máxima permitido $\left[\frac{(C)}{2^{N-r}} \right]$ no r -ésimo estágio por uma restrição mais branda, ou seja, permitir sintetizações entre duas viagens desde que não violem a capacidade do veículo (C).

A enorme popularidade do algoritmo de *Clarke e Wright* (1964) se deve também à sua flexibilidade em se adaptar aos diferentes tipos de problemas de roteamento e scheduling de veículos e o único fator negativo a respeito das duas versões é a necessidade de em ambos ter de resolver um (PCV) na construção das matrizes de distâncias (custos) mínimos a cada estágio.

O algoritmo de *Clarke e Wright (1964)* que popularizou a classe de heurísticas do tipo "savings" pode ser resumido da seguinte forma:

Passo 1) Gerar uma solução inicial idêntica à de *Dantzig e Ramser (1959)*.

Passo 2) Se o destino i for conectado diretamente com o destino j , elimina-se um veículo e economiza-se a distância.

$$s_{ij} = d_{io} + d_{oj} - d_{ij}$$

Passo 3) Se $s_{ij} < 0$, a sintetização é viável.

Caso contrário, se $s_{ij} \geq 0$ procura-se outra sintetização, retornando ao Passo 2) até que não reste nenhuma agregação viável a ser feita.

Outra análise sobre procedimentos heurísticos da classe *savings* foi feito por *Gaskell (1967)* para o problema do despacho de veículos.

Tillman e Coehran (1968) apresentaram uma nova heurística para o (PRV) generalizando o algoritmo de *Clarke e Wright (1964)*. A exemplo da versão original, a versão generalizada permite a inclusão de outros tipos de restrições

sem maiores dificuldades.

Christofides e S.Eilon (1969) analisaram um (PRV) onde um conjunto de clientes devem ser atendidos por uma frota de veículos localizados num depósito central.

As restrições são relacionadas com o atendimento das demandas dos clientes, a satisfação das capacidades dos veículos e o atendimento das distâncias máximas para cada rota.

O problema foi resolvido usando três técnicas: "Branch-and-Bound", "savings" e "3-rota ótima".

No método de Branch-and-Bound, um limitante inferior foi gerado através de um método baseado no algoritmo de *Little et al (1963)* que transforma um (PMCV) num (PCV). (Recorde que o (PRV) é uma extensão do (PMCV)).

O método do tipo "savings" utilizado se assemelha com o algoritmo proposto anteriormente por "Clarke e Wright (1964)" e finalmente o método "3-rota ótima" é uma extensão do método de *Lin (1965)* para o (PCV).

Resultados computacionais de dez problemas testes mostraram as dificuldades do primeiro procedimento nos problemas de dimensões elevadas e em linhas gerais, uma superioridade do terceiro procedimento.

No Brasil, *Graciano Sá (1969)* apresentou um trabalho sobre o (PRV) denominado "*Diálogo sobre programação matemática generalizada e o problema do despacho de carga*". Nele, o autor cita três argumentos que o levaram a escrever sobre o assunto. O primeiro motivo é o desconhecimento na época das técnicas de Programação Linear Generalizada e de Programação Inteira pela maioria dos técnicos das empresas. Os outros dois motivos foram a grande difusão do problema na literatura e a proximidade dos objetivos da Programação Linear com os problemas de tomada de decisão dentro de uma empresa.

A exposição foi desenvolvida na forma de um diálogo entre técnicos de uma empresa de navegação mostrando as dificuldades e o potencial das técnicas de Programação Matemática para atacar este tipo de problemas.

Outros procedimentos heurísticos e exatos para o (PRV) foram apresentados por *Pierce (1968)*.

As técnicas exatas são do tipo Branch-and-Bound e as heurísticas, segundo o seu autor, são variantes de uma heurística anteriormente proposta por *Rossmann e Twery (1958)* (artigo não publicado).

Nemhauser (1969) estudou o problema de roteamento de trens entre dois pontos A e B. Este percurso pode ser feito

diretamente (serviço expresso) ou indiretamente, passando por pontos intermediários (serviços locais).

Obviamente o serviço local demanda um tempo bem maior que o expresso.

Finalmente foi desenvolvido um algoritmo para gerar rotas ótimas via programação dinâmica associando os dois tipos de serviços existentes.

Bellmore, Bennington e Lubore (1971) generalizaram o problema de programação de navios petroleiros desenvolvido anteriormente por *Dantzig e Fulkerson (1954)*, formulando-o como um problema de programação linear mista. O problema linear contínuo foi resolvido via SIMPLEX e posteriormente foi usado Branch-and-Bound para obter solução linear mista.

LWERR (1972) realizou um estudo sobre a performance dos principais métodos heurísticos sequenciais da época para o (PRV).

Os resultados computacionais de uma variedade de problemas de roteamento de múltiplas dimensões mostrou de uma forma geral a superioridade daqueles métodos que utilizam o critério de economias elaborado por *DANTZIG e RAMSER (1959)* e por *CLARKE e WRIGHT (1964)*.

Uma variante do (PRV) clássico onde existem múltiplos

terminais e onde as demandas são probabilísticas foi analisado por *TILLMAN e CAIN (1972)*.

Neste trabalho, os autores propuseram novos procedimentos heurísticos que podem ser vistos como extensões do algoritmo de *CLARKE e WRIGHT (1964)*.

Neste mesmo ano, *WREN e HOLLIDAY (1972)* apresentaram um novo algoritmo heurístico para o (PRV) com múltiplos depósitos, com restrições de capacidade para os veículos e distância (tempo) máxima para as rotas.

O procedimento admite uma rota contendo mais de um depósito mas sujeito ao número máximo de veículos para cada depósito.

Os resultados computacionais de alguns problemas testes para o caso particular com um único depósito mostraram a sua superioridade frente ao algoritmo clássico de *CLARKE e WRIGHT (1964)*.

GILLET e MILLER (1974) apresentaram um novo algoritmo do tipo *varredura* para o problema de despacho de veículos a partir de um depósito central.

Os autores observaram que o tempo computacional aumenta proporcionalmente com o quadrado do número médio de atendimentos numa rota para o problema com um número fixo de

demandas.

Comparativamente com os métodos do tipo *economias* propostas por GASKELL (1967) e o proposto por CHRISTOFIDES e EILON (1969), este novo método do tipo *varredura* se mostrou muito superior no tocante à *qualidade* das soluções geradas, embora seja um algoritmo que exige maior tempo computacional.

ORLOFF (1974) descreveu um modelo que estuda uma versão anterior de sua autoria que tratava de um problema de roteamento com um único veículo.

No modelo atual, o problema de roteamento é composto por M veículos, várias origens (depósitos) e um único destino (PRMV).

Dentre as aplicações para este tipo de problemas podemos tomar como exemplo, o problema da coleta do lixo urbano (onde o depósito de descarga é único); o problema de roteamento de ônibus escolar (para uma única escola), etc.

O procedimento que transforma o (PMCV) num (PCV) foi estendido mostrando que o grafo do (PRMV) é equivalente a uma forma modificada do grafo do (PRV) com um único veículo. Com isso, provou-se que os algoritmos existentes para o (PRV) com um único veículo resolvem o (PRMV).

DINKEL, KLEINDORFER, KOCHENBERGER e WONG (1976)

desenvolveram uma formulação do (PCV) com restrições adicionais direcionadas para problemas relacionados com o controle do meio ambiente.

A formulação matemática apresentada, envolve variáveis binárias com até três índices o que o torna um problema de difícil solução usando somente técnicas exatas.

Partindo deste pressuposto, os autores desenvolveram vários esquemas para obter soluções aproximadas do problema, baseados em procedimentos heurísticos da classe *vizinho mais próximo* e *vizinho mais distante*.

FOSTER e RYAN (1976) mostraram novas metodologias para incorporar a partir de um modelo de particionamento outras restrições para uma variedade de problemas de roteamento de veículos.

A solução contínua é obtida via método simplex revisado e à integralidade assegurada adicionando-se várias restrições ao problema de particionamento.

GILLET e JOHNSON (1976) estenderam o algoritmo proposto anteriormente por GILLET e MILLER (1974) introduzindo um novo algoritmo do tipo *varredura* para problemas de despacho de veículos com múltiplos terminais (origens), múltiplos destinos, frota homogênea e rotas com restrições de tamanho

(tempo) máximo.

Nos testes computacionais, o algoritmo se mostrou viável para problemas com muitos terminais e centenas de pontos demanda, e possibilitando a seguinte observação :

Quando o número médio de atendimentos por rota e o número total de atendimentos permanecem aproximadamente fixos, é conveniente aumentar o número de terminais para reduzir o tempo computacional.

HOLMES e PARKER (1976) desenvolveram uma nova extensão do algoritmo de despacho de veículos de *CLARKE e WRIGHT (1964)*.

Estes novos algoritmos desenvolvidos para as versões simétricos e não simétricos permitem melhoramentos das soluções finais através de um procedimento de perturbação controlada pelo analista.

LENSTRA e RINOOY KAN (1976) publicaram uma nota mostrando algumas incorreções nas afirmações feitas por *ORLOFF (1974)* onde são apresentadas novas metodologias para a solução do (PMCV).

MOLE e JAMENSON (1976) desenvolveram um novo método sequencial de geração de rotas para (PRV).

O algoritmo é uma generalização do método das *economias* de CLARKE e WRIGHT (1964) e é basicamente constituído dos seguintes passos dados a seguir:

Inicialmente, o algoritmo analisa a possibilidade de alocar várias tarefas (ao invés de uma) para cada veículo procurando com isso minimizar a capacidade ociosa dos veículos.

No passo seguinte, determina-se a melhor posição da rota para encaixar um cliente ainda não conectado em nenhuma rota.

A seguir, identifica-se o próximo cliente a ser agregado na rota e finalmente no passo seguinte é feita uma análise da viabilidade de um novo sequenciamento dos clientes de cada rota usando métodos do tipo 2-ótimos.

ORLOFF (1976) descreveu um modelo geral para o (PRV) e mostrou que dependendo da característica de cada problema particular, o modelo pode ter a sua dimensão e a sua complexidade reduzidas sensivelmente. Vários casos particulares são citados; como o (PCC), (PCV), (PMCV), o problema do carteiro rural (PCR), etc.

ZIERER, MITCHELL e WRITE (1976) apresentaram formulações matemáticas para os problemas de distribuição de derivados do petróleo da SHELL.

O problema básico de distribuição consiste de três tipos de produtos envolvendo três sistemas de transportes, entre quatro origens dos produtos e um elevado número de pontos intermediários e de pontos de demandas.

Outro trabalho aplicativo do (PRV) foi apresentado por *BRADLEY, BROWN e GRAVES (1977)* que descreveram uma família de problemas de transporte marítimo resolvendo-os como problemas de fluxo de custo mínimo.

Um trabalho sobre a qualidade dos limites gerados pelas heurísticas do tipo "savings" foi desenvolvido por *GOLDEN (1977a)*.

Nele, o autor faz uma avaliação da qualidade das soluções das versões sequencial e concorrente do algoritmo de *CLARKE e WRIGHT (1964)*, procurando apontar o seu pior caso.

A determinação da qualidade da solução de um procedimento heurístico não é uma tarefa muito simples de se verificar.

Na sua análise, *GOLDEN (1977a)* resolveu inicialmente vários problemas testes de dimensões reduzidas para os quais se sabia antecipadamente que o método das *economias* fornecia solução ótima.

Estes resultados exatos são incorporados a outras análises para se chegar a um limite no pior caso.

São apresentados exemplos patológicos indicando que existem situações onde o algoritmo sequencial e concorrente de CLARKE e WRIGHT (1964) gera soluções muito pobres. Por exemplo, no caso sequencial (CWS) gerou-se uma desigualdade do tipo:

$$\frac{V(\text{CWS})}{V(\text{ÓTIMO})} > \frac{4 \log(n) + 5}{21}, \quad \text{onde } n = \text{número de nós}$$

do (PRV) e $V(T)$ representa o valor da solução final do problema T .

Mas o autor reconhece que na prática, estas situações são raras e que em geral os algoritmos do tipo *savings* (*economias*) apresentam performance bastante satisfatórias.

GOLDEN, MAGNANTI, NGUYEN (1977) escreveram um trabalho mostrando a importância do problema de alocação e roteamento de veículos num sistema logístico.

O trabalho tem como objetivo, o de mostrar que com uma estrutura de dados eficiente, pode-se melhorar sensivelmente o desempenho de alguns algoritmos heurísticos encontrados na literatura.

Técnicas heurísticas como as propostas por Clarke e

Wright (1964); Gillet e Miller (1974) e Tyagi (1968) são implementadas com modificações permitindo a solução aproximada de problemas de grande porte num pequeno espaço de tempo (alguns segundos).

Finalizando, é apresentada uma aplicação referente ao problema da distribuição de jornais com cerca de 600 pontos destinos.

Outra variante do (PRV) clássico foi apresentada por Golden e Yee (1979) que desenvolveram vários procedimentos heurísticos para resolver um (PRV) probabilístico, isto é, um (PRV) onde uma frota fixa de veículos atende um conjunto de nós destinos com demandas probabilísticas, à partir de um depósito central.

Os veículos com restrições de capacidade devem eventualmente sair e retornar do depósito.

O problema tem como objetivo gerar um conjunto de rotas com distância total esperada mínima.

As demandas são modeladas através de uma distribuição de probabilidade (por exemplo Poisson) com média λ_i e pressupõe-se que as demandas sejam mutuamente independentes.

MOLE, LIEBMAN e ORLOFF (1977) apresentaram uma versão melhorada de uma heurística anteriormente proposta por

ORLOFF (1974) para o (PRV).

A versão original permitia o descarte de certos tipos de rotas que eventualmente poderiam estar na solução ótima do problema.

Baseado nestas informações, são sugeridas modificações simples no algoritmo original de modo a obter a otimalidade, com maior frequência.

Outro trabalho que enfoca a análise do pior caso de várias heurísticas foi apresentado por FREDERICKSON, HECHT e KIM (1978).

STEIN (1978a) definiu um tipo particular de problemas de roteamento de veículos como o *problema do ônibus* que é na verdade, uma generalização do (PCV) clássico.

Este problema tem como objetivo determinar o tamanho da rota requerida para um único ônibus carregar e descarregar n passageiros de locais aleatórios para locais igualmente aleatórios, dentro de uma região previamente limitada no plano.

O trabalho apresenta alguns resultados interessantes como a de que o comprimento da rota dividido pela raiz quadrada do número n converge quase que seguramente para a raiz quadrada de pouco mais que as duas vezes a área da

região limitada, quando $n \rightarrow \infty$.

MOLE (1979) apresentou um *survey* sobre os problemas de despacho de veículos encontrados na literatura nesta época.

RUSSEL e IGOR (1979) descreveram o problema de alocar pontos de demanda de clientes para os dias de uma semana.

Foi suposto uma certa flexibilidade destes pontos de demandas nas suas alocações, podendo inclusive ocorrer caso de clientes alocáveis a qualquer dia da semana.

Para cada dia da semana é gerado uma nova rede e supõe-se que estas redes tenham em comum as seguintes características: um único depósito central e como objetivo, minimizar a distância (tempo) total raquerida para atender a todas as suas demandas minimizando também o número de veículos utilizados:

O conjunto de restrições do problema pode ser assim resumido:

V = número de veículos disponíveis diariamente

C_k = capacidade de carga do veículo k

D_k = distância (tempo) máximo para cada veículo k sobre qualquer rota.

$Q_{i,d}$ = demanda do nó i no dia d .

U_d = carregamento máximo permitido no dia d .

$d_{i,j}$ = distância (tempo) do nó i ao nó j .

P_j = conjunto de dias da semana permissíveis ao nó j .

RESTRICÇÕES:

R1) A demanda total de nós alocados no dia i não deve ultrapassar U_i .

R2) Respeitar o número máximo (V) de veículos disponíveis por dia.

R3) A demanda total alocada a um veículo k deve ser igual ou menor à capacidade C_k .

R4) A distância (tempo) total percorrida por qualquer veículo k não deve ultrapassar o limite D_k .

R5) Cada ponto de demanda deve ser atendido por um único veículo em cada dia da semana designado.

R6) Quando uma demanda permite seu atendimento somente

em alguns dias da semana (segunda e quinta) e requer dois dias da semana para o seu atendimento, evitar alocações parcialmente inviáveis (segunda e terça) por exemplo.

Para a solução deste problema foram desenvolvidos diversos procedimentos heurísticos da classe *savings* de CLARKE e WRIGHT (1964) e do tipo *N-Tour* de RUSSEL (1963).

LAPORTE e NOBERT (1980) mostraram a semelhança entre os vários tipos de problemas de roteamento de veículos no sentido de que com relaxações adequadas todos podem ser resolvidos como um (PCV).

MORTAGY e ELEITHY (1980) desenvolveram um sistema para identificar os fatores que poderiam influenciar o custo de transportes numa empresa de exploração e transportes de petróleo e seus derivados.

São apresentados : um procedimento analítico e outra técnica simples para gerar o tamanho ótimo da frota de veículos.

CHAPLEAU, FERLAND e ROUSSEAU (1981) introduziram neste trabalho um novo procedimento para gerar rotas de ônibus escolar numa zona urbana densamente povoada. O método proposto é uma variante de uma heurística do tipo *varredura* anteriormente apresentada por DULAC, FERLAND e FORGUES (1980).

Nesta nova versão, se torna possível avaliar o nível de saturação das rotas geradas.

CHRISTOFIDES, MINGOZZI e TOTH (1981a) apresentaram três algoritmos de busca em árvore para o (PRV). Inicialmente é determinado um limite inferior usando métodos do tipo *shortest spanning k - degree center tree (k-DCT)* e *q-rotas*.

O primeiro é o problema da árvore geradora mínima com um grau fixado para cada vértice e o segundo gera com rotas demanda total exatamente igual a q .

Posteriormente são utilizadas técnicas de Branch-and-Bound nos arcos e nas rotas para gerar soluções inteiras.

O problema de roteamento de veículos considerado neste trabalho tem a seguinte formulação matemática como um problema linear inteiro.

$$\text{Minimizar } \sum_{i=0}^N \sum_{j=0}^N (c_{ij} \sum_{k=1}^M x_{ijk}) \quad (2.77)$$

$$\text{S.A. } \sum_{i=0}^N \sum_{k=1}^M x_{ijk} = 1, \quad j = 1, \dots, N \quad (2.78)$$

$$\sum_{i=0}^N x_{ipk} - \sum_{j=0}^N x_{pjk} = 0, \quad k = 1, \dots, M \quad (2.79)$$

$$p = 0, \dots, N$$

$$\sum_{i=1}^N (q_i \sum_{j=0}^N x_{ijk}) \leq Q, \quad k = 1, \dots, M \quad (2.80)$$

$$\text{(PRV)} \quad \sum_{i=0}^N \sum_{j=0}^N c_{ij} x_{ijk} + \sum_{i=1}^N (m_i \sum_{j=0}^N x_{ijk}) \leq T, \quad k = 1, \dots, M \quad (2.81)$$

$$\sum_{j=1}^N x_{0jk} = 1, \quad k = 1, \dots, M \quad (2.82)$$

$$y_i - y_j + N \sum_{k=1}^M x_{ijk} \leq N - 1, \quad (i \neq j) = 1, \dots, N \quad (2.83)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i = 0, \dots, N, \quad \forall j = 0, \dots, N \quad (2.84)$$

$$\forall k = 0, \dots, M$$

$$y_i \quad \text{qualquer } i = 1, \dots, N \quad (2.85)$$

onde $x_{ijk} = \begin{cases} 1, & \text{se o veículo } k \text{ visita } j \text{ logo após visitar } i \\ 0, & \text{c/c.} \end{cases}$

q_i = demanda do cliente i .

m_i = custo requerido por um veículo para descarregar a quantidade q_i para o cliente i .

M = número de veículos homogêneos.

N = número de clientes.

Q = capacidade dos veículos.

T = distância (tempo) máximo permitido para cada rota.

$C_{i,j}$ = custo de se locomover de i para j .

As restrições (2.78) exigem que cada cliente j deve ser visitado exatamente uma única vez enquanto (2.79) nos diz que se um veículo chega ao nó p ele deve necessariamente sair deste nó destino.

As condições (2.80) e (2.81) são as restrições de capacidade e de distância máxima enquanto (2.82) afirma que cada veículo deve ser usado exatamente uma única vez e finalmente (2.83) são as restrições para eliminar soluções desconexas com a origem propostos por MILLER, TUCKER e ZEMLIN (1960).

CHRISTOFIDES, MINGOZZI e TOTH (1981b) enfocaram as dificuldades para resolver um problema de otimização

combinatória exclusivamente por Programação Dinâmica devido ao elevado número de vértices do grafo correspondentes às variáveis de estado. Após estas argumentações, os autores propuseram um procedimento que relaxam as variáveis de estado para produzir um bom limitante inferior para se usar uma técnica do tipo Branch-and-Bound.

Em programação linear inteira, este tipo de relaxação seria equivalente a uma relaxação lagrangeana.

CULLEN, JARVIS e RATLIFF (1981) resolveram uma variedade de problemas de roteamento de veículos formulando-os como um modelo de particionamento (*set partitioning model*).

As implementações são acompanhadas por uma visualização gráfica mostrada na tela do computador e foram testados para problemas com até 100 destinos, resolvidos via simplex com geração de colunas e Branch-and-Bound.

FISHER e JAIKUMAR (1981) apresentaram um novo algoritmo heurístico para a solução do problema de roteamento de veículos localizados num depósito central.

O procedimento de escolha de *clientes para veículos* é obtido através de um problema de alocação linear generalizada com uma função objetivo que fornece aproximadamente os custos de atendimento.

O algoritmo é bem flexível podendo acomodar facilmente restrições adicionais; além de garantir uma solução viável para o problema, caso exista.

A formulação matemática do (PRV) analisado é da forma:

$$\text{Minimizar } \sum_k \sum_j \sum_i c_{ijk} x_{ijk} \quad (2.86)$$

$$\text{S.A. } \sum_i a_i y_{ik} \leq b_k, \quad k = 1, \dots, K \quad (2.87)$$

$$\sum_k y_{ik} = \begin{cases} K, & \text{se } i = 0 \\ 1, & \text{se } i = 1, \dots, n \end{cases} \quad (2.88)$$

$$y_{ik} \in (0, 1), \quad i=0, 1, \dots, n \\ k=1, \dots, K \quad (2.89)$$

$$\sum_i x_{ijk} = y_{jk}, \quad j = 0, \dots, n \\ k = 1, \dots, K \quad (2.90)$$

$$\sum_j x_{ijk} = y_{ik}, \quad i = 0, \dots, n \\ k = 1, \dots, K \quad (2.91)$$

$$\sum_i \sum_j x_{ijk} \leq |S|-1, \quad S \subset \{1, \dots, n\}, \quad 2 \leq |S| \leq n-1 \\ k = 1, \dots, K \quad (2.92)$$

$$x_{ijk} \in (0, 1), \quad i=0, \dots, n, \quad j=0, \dots, n, \quad k=1, \dots, K \quad (2.93)$$

As restrições (2.87) - (2.89) representam o problema da alocação generalizada.

Observe que se fixarmos y_{ik} de modo a satisfazer às condições (2.87) - (2.89), então para um dado $k \in K$ as restrições (2.86), (2.90) - (2.93) definem um (PCV) sobre os clientes alocados para o veículo $k \in K$.

O algoritmo inicialmente aloca clientes para veículos satisfazendo as restrições (2.87) - (2.89) com uma função objetivo que fornece aproximadamente o custo da rota do (PCV) de cada veículo.

A etapa seguinte consiste em obter uma solução completa resolvendo um (PCV) heurístico ou outros algoritmos que forneçam a sequência mais favorável de atendimentos.

LENSTRA e RINNOOY KAN (1981) fizeram um estudo sobre a complexidade dos principais problemas de roteamento de veículos encontrados na Literatura.

O trabalho define a classe dos problemas *NP-árduo* (*NP-hardness*), elabora um resumo sobre os problemas pertencentes a esta classe e finalmente desenvolve um estudo sobre o *pior-caso* de procedimentos inexatos para o (PRV).

Mostramos a seguir duas tabelas desenvolvidas neste

trabalho referenciando a complexidade dos (PRV) e os autores destes resultados.

TABELA 2.1:

Complexidade dos problemas de Roteamento de veículos.

PROBLEMA	COMPLEXIDADE	REFERÊNCIA
(PRV)	NP-árduo	-
(PCV)	NP-árduo	R. M. KARP (1972)
(PCV) direcionado	NP-árduo	R. M. KARP (1972)
(PCC)	$O(v^3)$	EDMONDS e JOHNSON (1973)
(PMCC)	NP-árduo	FREDERICKSON, HECHT, KIM (1978)
(PCC) direcionado	$O(v^3 \log a)$	EDMONDS e KARP (1972)
(PMCC) direcionado	NP-árduo	FREDERICKSON, HECHT, KIM (1978)
(PCC) misto	NP-árduo	PAPA DIMITRIOU (1976)
(PCR)	NP-árduo	LENSTRA e RINNOOY KAN (1976)
(PCR) direcionado	NP-árduo	LENSTRA e RINNOOY KAN (1976)

TABELA 2.2: PIOR-CASO dos algoritmos aproximados do (PRV)

Problema	Algoritmo	Limite Inferior	Limite Superior	Comple-xidade	Referência
	nearest neighbor	$\frac{1}{2} \lceil \log v \rceil + \frac{1}{2}$	$\frac{1}{2} \log(v+1) + \frac{4}{9}$	$O(v^2)$	ROSENKRANTZ, STEARNS e LEWIS (1977)
	sequential CLARKE-WRIGHT	—	$\frac{2 \log v + 5}{7}$ $\frac{4}{21}$	$O(v^2)$	GOLDEN (1977)
	insertion	$\lceil \log v \rceil + 1$	4	$O(v^2)$	ROSENKRANTZ, STEARNS e LEWIS (1977)
(PCV)	nearest insertion	2	2	$O(v^2)$	idem
	cheapest insertion	2	2	$O(v^2 \log v)$	idem
	nearest addition	2	2	$O(v^2)$	idem
	nearest merger	2	2	$O(v^2 \log v)$	idem
	k-optimal p/ $k \leq v/4$	—	2	—	idem
	double spanning tree	2	2	$O(v^2)$	idem
	spanning tree + matching	$\frac{3}{2}$	$\frac{3}{2}$	$O(v^3)$	CHRISTOFIDES, CORNUEJOLS e NEMHAUSER (1978)
(PCV) direc.	repeated assignment	$\lceil \log v \rceil$	—	$O(v^3)$	MAFFIOLI
(PCC) misto	EDMONDS-JOHNSON	2	2	$O(v_a^3 + e_a^2 a + a)$	FREDERICKSON (1979)
	reverted EDMONDS-JOHNSON	2	2	$O(\frac{1}{2} v_a^3 + e_a a + a)$	idem

Problema	Algoritmo	Limite Inferior	Limite Superior	Complexidade	Referência
	mixed strategy	$\frac{5}{3}$	$\frac{3}{2}$	$O(v^{\frac{3}{2}} + e^2 a)$	FREDERICKSON (1979)
(PCR)	spanning tree + matching	$\frac{3}{2}$	$\frac{3}{2}$	$O(v^3 + e)$	<i>idem</i>
(PCHV)	nearest neighbor	$\frac{M \log v + M}{2}$	$\frac{M \log v}{6}$	$O(v^2)$	FREDERICKSON, HETCH e KIM (1978)
	nearest insertion	$2M$	$2M$	$O(v^2)$	<i>idem</i>
	tour splitting	$\frac{5}{2} - \frac{1}{M}$	$\frac{5}{2} - \frac{1}{M}$	$O(v^3)$	<i>idem</i>
(PMCC)	tour splitting	$2 - \frac{1}{M}$	—	$O(v^3)$	<i>idem</i>

na tabela 2.1, e tabela 2.2 temos que:

(PRV) = Problema de Roteamento de Veículos

(PCV) = Problema do Caixeiro Viajante

(PCC) = Problema do Carteiro Chinês

(PMCC) = Problema dos M-Carteiros Chineses

(PCR) = Problema do Carteiro Rural

v = número de vértices do grafo

a = número de arcos direcionados do grafo

e = número de arcos não direcionados do grafo

$\lceil x \rceil$ = menor inteiro igual ou maior que x

M = número de caixeiros (carteiros)

MAGNANTI (1981) apresentou um novo resumo sobre a

Literatura existente sobre o (PRV), enfocando as principais formulações e as técnicas mais utilizadas.

FISHER, GREENFIELD, JAIKUMAR e LESTER (1982)

desenvolveram um sistema para um problema de planejamento da *E.I. DuPont, Inc*, utilizando métodos do (PRV) para reduzir os custos de despacho de produtos em cerca de 15%.

O problema aplicativo possui cerca de 1500 clientes e cerca de 50 rotas.

LAPORTE (1982) descreveu uma nova versão do (PRV) na qual são introduzidos pesos não-negativos para cada cidade a ser visitada por uma frota heterogênea de veículos.

As cidades podem ser visitadas mais de uma vez se assim for conveniente e além disso algumas cidades podem não ser visitadas (peso nulo).

No trabalho, foi proposto um procedimento exato baseado num algoritmo anteriormente desenvolvido por *FLEISCHMANN (1982)*.

McBRIDE (1982) apresentou um novo algoritmo para resolver o problema da coleta de lixo urbano.

O problema define rotas particulares denominadas *lefts* e *U-turns* cujo percurso é mais moroso e mais sujeito a

acidentes de tráfego.

O algoritmo proposto é uma versão modificada do anteriormente desenvolvido por *BODIN e KURSH (1978)* e tem como objetivo básico reduzir as rotas do tipo *left* e *U-turns*.

BARTHOLDI, PLATZMAN, COLLINS e WARDEN (1983) desenvolveram um sistema manual de roteamento de veículos.

O procedimento é uma heurística de fácil manejo e foi elaborado para ser utilizado manualmente, sem a necessidade de um computador.

O algoritmo foi aplicado a problemas de roteamento de até 200 destinos cujas localizações podem se alterar a cada dia, mostrando uma economia de até 13% em relação à solução atualmente utilizada.

LAPORTE, NOBERT E ARPIN (1984) formularam um (FRV) com múltiplos depósitos como um problema de Programação linear inteira (PFLI) descrito a seguir:

$$\text{Minimizar } \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} + \sum_{n \in R} f_r m_r \quad (2.94)$$

$$\text{S.A. } \sum_{i < k} x_{ik} + \sum_{k < j} x_{kj} = 2, \quad k \in N-R \quad (2.95)$$

$$\sum_{i < r} x_{i_r} + \sum_{r < j} x_{r_j} = 2mr, \quad r \in R \quad (2.96)$$

$$\sum_{i \in S} \sum_{j \in S} x_{i_j} \leq |S| - \left\lceil \sum_{k \in S} d_k / D \right\rceil, \quad S \subseteq N-R, \quad |S| \geq 3 \quad (2.97)$$

$$X_{i_1 i_2} + 3 X_{i_2 i_3} + X_{i_3 i_4} \leq 4, \quad i_1, i_4 \in R, \quad i_2, i_3 \in N-R \quad (2.98)$$

$$X_{i_1 i_2} + X_{i_{h-1} i_h} + 2 \sum_{i \in P} \sum_{j \in P} x_{i_j} \leq 2h - 5, \quad h \geq 5, \quad i_1, i_h \in R \quad (2.99)$$

$$i_2, \dots, i_{h-1} \in N-R$$

$$d_{i_2} + \dots + d_{i_{h-1}} \leq D$$

$$\text{onde } P = (i_2, \dots, i_{h-1})$$

$$i \leq ml \leq mr \leq \bar{m}r, \quad r \in R \quad (2.100)$$

$$x_{i_j} \in (0, 1), \quad i \in N-R, \quad j \in N-R \quad (2.101)$$

e

$$x_{i_j} \in (0, 1, 2), \quad \text{para } (i \text{ ou } j) \in R \quad (2.102)$$

onde N = conjunto de nós

d_i = demanda do nó i

D = capacidade dos veículos

$C = (c_{ij})$ = matriz de custos simétricos

$(R \subset N)$ = conjunto dos depósitos

m_r = veículos homogêneos localizados no depósito r

f_r = custo fixo para usar um veículo do depósito r

m_r = limite inferior de m_r

$\overline{m_r}$ = limite superior de m_r

$$\lceil x \rceil = \begin{cases} \text{menor inteiro } \geq x, & \text{se } x > 0 \\ 1, & \text{c/c.} \end{cases}$$

x_{ij} = número de vezes que o trecho entre i e j é usado na solução ótima

$$x_{ij} \text{ não é definido para } \begin{cases} - i \geq j \\ - (i, j) \in R \times R \\ - d_i + d_j > D. \end{cases}$$

As restrições (2.95) e (2.96) especificam o grau de cada nó e (2.97) são restrições para eliminar sub-rotas.

As condições (2.98) e (2.99) são restrições de barreira proibindo as ligações diretas de dois depósitos numa mesma

rota.

Neste trabalho, o (PRV) descrito por (2.94) - (2.102) foi resolvido por etapas.

Inicialmente foi gerado: uma solução viável através de uma heurística e um limite inferior através da solução de uma relaxação do problema.

A solução final é obtida através de testes de viabilidade da solução parcial nas restrições relaxadas.

CERDEIRA e COELHO (1984) escreveram um novo survey sobre os problemas de otimização de percursos de distribuição ou (PRV). Neste trabalho foi feito um levantamento quase completo das diversas formulações matemáticas e as principais técnicas de otimização exatas e aproximadas existentes na Literatura sobre o problema.

CHAPLEAU, FERLAND e ROUSSEAU (1984) desenvolveram uma nova técnica para obter soluções aproximadas do problema de roteamento em arcos capacitados. O procedimento se baseia num método de inserção paralela que gera várias rotas simultaneamente, procurando com isso, obter um melhor balanceamento entre os custos das diversas rotas.

Duas estratégias complementares são consideradas no algoritmo:

i) Dado um arco, determinar em qual rota ele deve ser inserido de modo a minimizar o desvio cometido.

ii) Dado uma rota, determinar qual arco restante deve ser a ele inserido.

O procedimento consiste em passar de uma estratégia para a outra até que as rotas estejam completas.

GOLDEN, ASSAD, LEVY e GHEYSENS (1984) propuseram várias heurísticas do tipo *savings* e do tipo *rotear primeiro e agrupar depois* para se determinar o tamanho "ótimo aproximado" de uma frota heterogênea de veículos num (PRV).

Normalmente o problema de gerar o tamanho ótimo de uma frota de veículos ou é precedido ou em muitos casos é dominado pela decisão de gerar as melhores rotas do (PRV).

A formulação matemática do (PRV) utilizada neste trabalho é de forma:

$$\text{Minimizar } \sum_{k=1}^T f_k \sum_{j=1}^n x_{0jk} + \sum_{k=1}^T \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ijk} \quad (2.103)$$

$$\text{S.A. } \sum_{k=1}^T \sum_{i=0}^n x_{ijk} = 1, \quad j = 1, \dots, n \quad (2.104)$$

$$\sum_{i=0}^n x_{ipk} - \sum_{j=0}^n x_{pjk} = 0, \quad k = 1, \dots, T \quad (2.105)$$

$$p = 1, \dots, n$$

$$r_0 = 0$$

$$r_j - r_i \geq (d_j + a_T) \sum_{k=1}^T x_{ijk} - a_T \quad \begin{matrix} i = 0, \dots, n \\ j = 0, \dots, n \end{matrix} \quad (2.106)$$

$$r_j \leq \sum_{k=1}^T \sum_{i=0}^n a_k x_{ijk}, \quad j = 0, \dots, n \quad (2.107)$$

$$x_{ijk} \in \{0, 1\}, \quad i=0, \dots, n, \quad j=0, \dots, n, \quad k=1, \dots, T \quad (2.108)$$

onde n = número de clientes

T = número de tipos de veículos

a_k = capacidade do veículo do tipo k ($a_1 < a_2 < \dots < a_T$)

f_k = custo fixo por veículo do tipo k utilizado

$$(f_1 < f_2 < \dots < f_T)$$

d_i = demanda do cliente i

c_{ij} = custo de percorrer o trajeto entre i e $j = c_{ji}$

r_i = variável de fluxo da mercadoria associada ao cliente i

$$x_{ijk} = \begin{cases} 1, & \text{se o veículo tipo } k \text{ percorre } (i, j) \\ 0, & \text{c/c.} \end{cases}$$

$(k = 0)$ = depósito

É suposto disponibilidade infinita do número de veículos de cada tipo.

As restrições (2.104) e (2.105) asseguraram que cada cliente é visitado exatamente uma única vez e que se o veículo chega a um nó destino então ele deve necessariamente sair deste nó.

A demanda do depósito é suposta ser nula ($r_0 = 0$) enquanto as restrições (2.106) e (2.107) se referem à capacidade dos veículos.

LADANY e MEHREZ (1984) analisaram um problema prático de roteamento de veículos no qual um caminhão deve carregar um produto de tamanhos semelhantes de n diferentes origens localizados numa região e despachar em n locais situados noutra região. A frota é composta de um único veículo e para exemplos de dimensões reduzidas, os autores sugerem um novo procedimento de enumeração.

STEWART e GOLDEN (1984) apresentaram um novo algoritmo heurístico para o (PRV) utilizando relaxações lagrangeanas para converter o (PRV) num (PMCV) modificado.

No (PRV) foram considerados as seguintes restrições:

i) cada demanda é atendida por exatamente um único veículo.

ii) a demanda total de uma rota é sempre menor ou igual à capacidade do veículo a ele alocado.

iii) cada rota inicia e termina no único depósito da rede.

FORMULAÇÃO MATEMÁTICA DO (PRV) CONSIDERADO:

$$\text{Minimizar } \sum_k \sum_i \sum_j c_{ij} x_{ijk} \quad (2.109)$$

$$\text{S.A. } \sum_i \sum_j \mu_i x_{ijk} \leq Q, \quad k = 1, \dots, m \quad (2.110)$$

$$X = (x_{ijk}) \in S_m \quad (2.111)$$

onde μ_i = quantidade demandada no local i

Q = capacidade de cada um dos m veículos

c_{ij} = custo de percorrer (i, j)

n = número de demandas ($n = 0$ é o depósito)

m = número de veículos disponíveis

S_m = conjunto de soluções viáveis para o (PMCV)

As restrições (2.109) e (2.111) descrevem o (PMCV) que se acrescidas de (2.110) se transforme no (PRV).

Uma outra restrição, esta redundante, foi acrescida ao problema (2.109) - (2.111); a saber:

$$\sum_i \sum_j \mu_i x_{ijk} - \sum_i \sum_j \mu_e x_{ije} \geq 0, \text{ para } e = k + 1 \quad (2.112)$$

$$e = k = 1, 2, \dots, m - 1$$

O objetivo de incluir uma restrição redundante ao (PRV) foi a de garantir que a maior demanda fosse alocada para a rota $k = 1$, e a segunda maior demanda para $k = 2$ e assim sucessivamente, e que numa relaxação lagrangeana os multiplicadores de Lagrange sejam proporcionalmente distribuídos com estas demandas como mostrado a seguir.

Relaxando as restrições (2.110) do (PRV) e incorporando-as na função objetivo (2.109) com multiplicadores não negativos $\lambda_k \geq 0$, $k = 1, \dots, m$, teremos o seguinte problema (RL λ).

$$\text{Minimizar } \sum_k \sum_i \sum_j c_{ij} x_{ijk} + \sum_k \lambda_k (\sum_i \sum_j \mu_i x_{ijk} - Q) \quad (2.113)$$

$$\text{S.A. } \sum_i \sum_j \mu_i x_{ijk} - \sum_i \sum_j \mu_e x_{ije} \geq 0, \text{ com } e = k + 1 \text{ e } \quad (2.114)$$

$$k = 1, \dots, m - 1$$

$$X = (x_{ijk}) \in S_m \quad (2.115)$$

Assim, se $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$ e pelo menos uma destas desigualdades for estrita, então a rota de maior demanda (menor k) irá ser mais fortemente penalizada no caso da violação da capacidade do veículo, o que é bem razoável.

Com isso se evitaria de ocorrer, por exemplo, o caso oposto, isto é, penalizar mais a rota contendo a menor demanda.

Isto justifica a introdução da restrição adicional (2.112) ao (PRV).

A solução do problema relaxado (RL λ) para λ fixado, foi obtido através de um algoritmo de *CROES (1958)* modificado.

TAN e BEASLEY (1984) estudaram o problema de roteamento de veículos periódico (PRVP) que consiste em alocar veículos para clientes para vários dias de um período de T-dias; onde nem todos os clientes requerem atendimentos diários.

A solução do (PRVP) foi obtida através de técnicas heurísticas baseadas no algoritmo de alocação generalizado de *FISHER e JAİKUMAR (1981)*.

WALTERS (1984a) baseado na idéia de que os algoritmos do (PRV) devem ser tanto flexíveis quanto sofisticados para produzirem resultados reais; propôs um novo procedimento interativo para o (PRV) que solicita intervenções do analista nos processos decisórios existentes nas diversas etapas do método.

Em outro trabalho *WALTERS (1984b)* analisou o efeito de

arredondar distâncias entre clientes num (PRV) como fazem diversos autores em seus respectivos algoritmos.

O autor salienta que em alguns casos a solução encontrada pode ser muito melhorada se o arredondamento for efetuado somente nas distâncias do estágio final do algoritmo.

Para esta constatação, foram implementados algoritmos de CLARKE e WRIGHT (1964) nas versões sequenciais e paralela e o algoritmo de GASKELL (1967).

WONG e BEASLEY (1984) descreveram o problema de roteamento de veículos usando áreas fixas de entregas para cada veículo.

Este procedimento é conhecido como *agrupar primeiro e rotear posteriormente* e é feito dividindo a área original que contém todos os clientes em um conjunto de sub-áreas menores, onde cada uma destas sub-áreas com suas demandas serão incorporadas a uma nova rota.

Após a formação das rotas iniciais são tentados remanejamentos de clientes entre sub-áreas de modo a tentar obter soluções mais eficientes.

Resultados computacionais demonstraram a eficiência do método em problemas de pequeno e médio porte.

AGARWAL (1985) propôs resolver um (PRV) em duas etapas. Inicialmente gera-se o conjunto das rotas viáveis do problema através de um procedimento heurístico e numa segunda etapa, resolve-se um problema de particionamento (PP) do tipo:

$$\text{Minimizar } \sum_j c_j y_j \quad (2.116)$$

$$\text{S.A. } \sum_j a_{ij} y_j = e, \forall i \quad (2.117)$$

$$y_j \in \{0, 1\}, \forall j \quad (2.118)$$

Devido ao elevado número de colunas do (PP) usou-se um método de geração de colunas da matriz de particionamento $A = (a_{ij})$.

A solução do (PP) é feito por fases, inicialmente é resolvido o problema (PP) com relaxação linear, substituindo (2.118) por (2.118') $y_j \in [0, 1], \forall j$.

O problema linear (2.116), (2.117) e (2.118') é então resolvido pelo método simplex.

A cada iteração do simplex, resolve-se um sub-problema de geração de colunas assim descrito.

Seja w o vetor dual a (2.117). Assim o objetivo do

sub-problema de geração de colunas (SF) é encontrar j tal que $c_j - w_{aj} = \text{mínimo} (c_k - w_{ak})$ onde a_j é a j -ésima coluna da matriz $A = (a_{ij})$.

Retornando à etapa inicial do algoritmo global, na contribuição das rotas do (PRV) pode ser usado uma técnica do tipo *agrupar-rotear* cujos passos são:

- i) Escolher o conjunto de nós para cada rota a_j
- ii) Definir a sequência de atendimento de cada nó na rota a_j

Se cada nó for atendido por um único veículo numa única visita, ii) será o (PCV).

Assim neste caso para a_j , c_j representa o custo da solução do (PCV) sobre o conjunto de nós de a_j .

A matriz $A = (a_{ij})$ é definida como:

$$a_{ij} = \begin{cases} 1, & \text{se o nó } i \text{ pertence a rota } j \\ 0, & \text{c/c.} \end{cases}$$

Isto é, o vetor $a_j = (a_{ij})$ é um vetor zero-um.

Fazendo $a_j = x$ ou $(a_{ij} = x_i)$ e se $f(x) = f(a_j)$ representa o valor da função objetivo da solução do (PCV)

sobre os nós da coluna $a_j = x$, então o sub-problema (SP) é descrito na forma:

$$\text{Minimizar } f(x) = \sum_i w_i x_i \quad (2.119)$$

$$\text{S.A.} \quad \sum_i u_i x_i \leq U \text{ (restrição capacidade)} \quad (2.120)$$

$$x_i \in (0, 1) \quad \forall i \quad (2.121)$$

Note que o (SP) descrito pelas restrições (2.119) - (2.121) não é tão difícil de se resolver se $f(x)$ for linear, por exemplo $f(x) = \sum_i p_i x_i$, neste caso (SP) é exatamente o problema de Mochila (PMO).

Linearizando $f(x)$ como $\sum_i p_i x_i$ (se necessário), obtemos um limite inferior de (SP) e a solução final pode ser facilmente obtida por métodos do tipo Branch-and-Bound.

BELARDO, DUCHESSI e SEAGLE (1985) desenvolveram um sistema interativo com apoio gráfico para auxílio nas tomadas de decisões em problemas de planejamento de rotas de veículos.

O sistema implantado para microcomputadores apresenta uma visualização gráfica das rotas que estão sendo geradas que juntamente com os resultados do desenvolvimento do algoritmo de geração das rotas, possibilitam uma melhor

análise sobre a viabilidade das rotas.

BUXEY (1985) apresentou um novo algoritmo do tipo *savings* para o (PRV), onde as sintetizações são feitas aleatoriamente.

O procedimento se mostrou eficiente, apresentando uma substancial redução no tamanho da frota e na distância necessária para cobrir as demandas comparada com outros algoritmos desta classe.

DAGANZO (1985) apresentou uma nova estratégia e algumas fórmulas de distância para (PRV).

No problema é assumido que cada veículo pode atender no máximo uma quantidade de clientes pré-estabelecida.

As fórmulas de distância são geradas com a finalidade de analisar o problema da distribuição física dos clientes agrupados para cada rota, dentro de uma técnica do tipo *agrupar-rotear*.

DESROSIERS, LAPORTE e SOUMIS (1985) analisaram o problema de roteamento de veículos cheios (capacidade de carga saturada) e com restrições de tamanho máximo para as rotas.

Apresentou-se uma formulação alternativa para este tipo

de problema, baseado no trabalho de SKITT e LEVARY (1985).

Os autores mostraram ainda que quando as limitações de tamanho máximo não são muito restritivas, esta nova formulação pode ser vista como um (FCV) assimétrico com restrições adicionais.

KULKARNI e BHAVE (1985) apresentaram uma nova formulação matemática do (PRV) descrito como um problema de Programação linear inteira.

O (PRV) aqui estudado é composto de uma frota homogênea de veículos, e restrições de custo máximo para as rotas, e foi descrito da seguinte forma:

Considere as seguintes notações:

n = nó origem

P = capacidade dos veículos

T_k = custo máximo para a rota k

L = número máximo de nós a serem visitados por um veículo

Q_i = demanda do nó i ($Q_n = 0$)

y_i , μ_i , v_i são números reais arbitrários

$$\text{Minimizar} \quad \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij} \quad (2.122)$$

$$\text{S.A.} \quad \sum_{i=1}^N x_{ij} = 1, \quad j = 1, 2, \dots, n-1 \quad (2.123)$$

$$\sum_{j=1}^N x_{ij} = 1, \quad i = 1, 2, \dots, n-1 \quad (2.124)$$

$$\sum_{i=1}^{N-1} x_{iN} = V \quad (2.125)$$

$$\sum_{i=1}^{N-1} x_{Nj} = V \quad (2.126)$$

$$x_{ij} = 0 \text{ ou } 1, \quad i = 1, \dots, N, \quad j = 1, \dots, N \quad (2.127)$$

$$y_i - y_j + L x_{ij} \leq L - 1, \quad \text{para } i \leq i \neq j \leq n-1 \quad (2.128)$$

$$\mu_i - \mu_j + P x_{ij} \leq P - Q_i, \quad \text{para } i \leq i \neq j \leq n-1 \quad (2.129)$$

$$v_i - v_j + T x_{ij} \leq T - c_{ij}, \quad \text{para } i \leq i \neq j \leq n-1 \quad (2.130)$$

As restrições (2.122) - (2.126) asseguram que cada nó destino é servido por exatamente um único veículo e que os V veículos são usados.

A condição (2.128) foi mostrada por MILLER, TUCKER e ZEMLIN (1960) e asseguram que toda solução começa e termina

no nó origem N e que cada rota possui no máximo L nós.

Este resultado foi generalizado neste trabalho com a introdução das restrições (2.129) e (2.130) que asseguram respectivamente que o carregamento e o custo sobre cada rota é menor ou igual à capacidade P do veículo (2.129) e ao custo máximo permitido T (2.130).

LAPORTE, NGUYEN e NOBERT (1985) desenvolveram um novo algoritmo exato para o (PRV) com restrições de distância máxima.

No problema, supõe-se que cada nó destino deve ser atendido numa única visita e por um único navio. Os autores observaram que se vale a desigualdade triangular dos custos, então esta última condição não é restritiva como mostramos no exemplo dado a seguir:

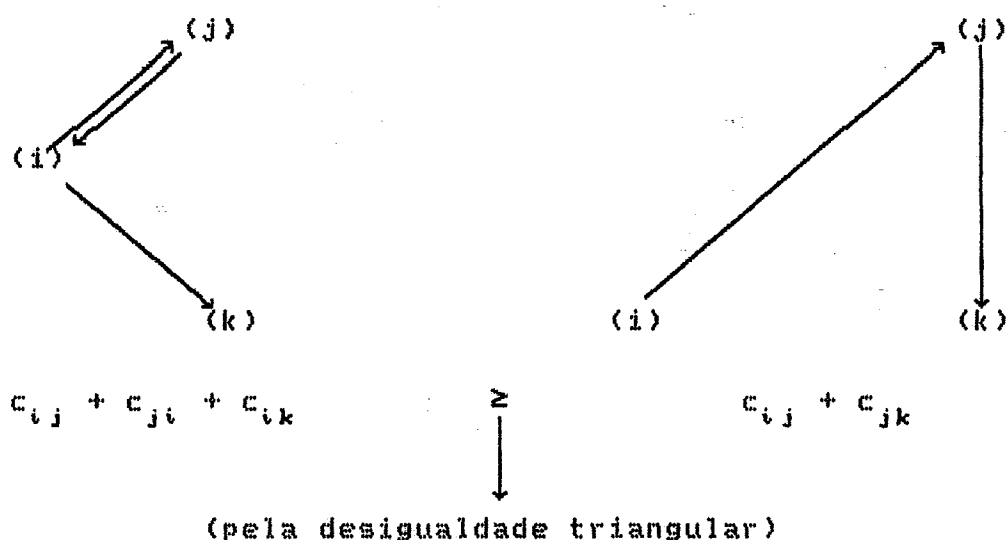


FIGURA 2.4

Por outro lado, se não vale a desigualdade triangular podemos ter casos onde é vantajoso passar por uma cidade mais de uma vez. Mas isso não implica necessariamente em violar a restrição de que cada nó destino deve ser atendido numa única visita e por um único veículo, podendo no entanto ser visitado mais de uma vez se isso reduzir a distância total (por exemplo se a matriz dos custos (c_{ij}) denotar a custo mínimo de i para j)

Inicialmente o problema é transformado num problema equivalente onde cada cidade (incluindo a origem) é visitado uma única vez através de um procedimento elaborado por *LENSTRA e RINNOOY KAN (1976)*.

O problema transformado é um (PMCV) (problema dos M-caixeiros viajantes) com restrições adicionais de tamanho máximo para os circuitos hamiltonianos.

Para a sua solução, foi proposto um algoritmo semelhante ao desenvolvido por *LAPORTE, MERCURE e NOBERT (1985)* onde são relaxados as restrições de eliminar sub-rotas, resultando num problema de alocação linear modificado que pode ser resolvido eficientemente usando o algoritmo proposto por *CARPANETO e TOTH (1980)*.

MEHREZ e STERN (1985) estudaram o problema de reabastecer uma frota de caminhões de dois tipos, um capaz e

o outro tipo incapaz de se auto reabastecer.

O problema é formulado como um modelo não linear com variáveis mistas, sendo posteriormente linealizado.

NELSON, NYGARD, GRIFFIN e SHREVE (1985) elaboraram um estado de arte sobre o procedimento heurístico do tipo savings de CLARKE e WRIGHT (1964) mostrando várias estruturas de dados e variações para este algoritmo.

DROR e LEVY (1986) analisaram vários métodos para o problema de gerar as melhores rotas num problema de transporte marítimo. O artigo sem trazer muitas informações sobre a aplicação sugere trabalhar com horizontes reduzidos para este tipo de problemas.

DROR e TRUDEAU (1986) apresentaram um problema de roteamento de veículos com demandas estocáveis e estabelecem neste artigo, o conceito de rotas incompletas para este tipo de problema.

Uma rota incompleta indica uma situação onde um veículo deixa de atender todos os clientes da sua rota motivado pelo término da sua carga após visitar somente parte de seus clientes (esta situação pode ocorrer devido às demandas serem não determinísticas).

Neste trabalho, os autores ilustram o efeito da

presença de uma *rota incompleta* sobre os custos esperados de uma *rota completa*, bem como o impacto provocado pela alocação de uma determinada rota no tocante aos custos esperados.

Mostrou-se também que os algoritmos heurísticos do tipo *savings* podem ser facilmente adaptados para (PRV) com demandas estocásticas.

Golden e Assad (1986) apresentaram o *estado da arte* do (PRV), discutindo os seus principais avanços obtidos pelos pesquisadores.

São abordados avanços nas metodologias, e o uso desta nova tecnologia em ambientes práticos, discutindo também as implementações de maior sucesso, os avanços no *realismo* dos modelos e o estágio de desenvolvimento dos novos algoritmos.

Finalmente são especuladas as possíveis direções de pesquisa nesta área.

Hochbaum e Shmoys (1986) estudaram vários procedimentos heurísticos para resolver o problema do engarrafamento de trânsito para pedestres e veículos, descrito como um (PRV) assimétrico com restrições de distância (tempo) máximo.

As heurísticas apresentadas garantem soluções com custos não maiores que duas vezes o valor ótimo e foi

mostrado que este resultado as coloca no nível dos melhores algoritmos heurísticos para o (PRV), no sentido de que a existência de outros procedimentos heurísticos com um melhor raio (pior caso) acarreta na igualdade $P = NP$, que é falso.

JORNSTEN e NASBERG (1986) apresentaram uma nova relaxação lagrangeana para o problema de alocação generalizada (PAO).

A relaxação se fundamenta na criação de cópias das variáveis do (PAO) de modo que o problema relaxado possa ser decomposto em problemas menores, independentes e mais simples.

A análise deste artigo nesta seção se deve ao fato de que estas relaxações podem ser facilmente adaptadas nos (PRV), como mostraremos nos capítulos posteriores.

O (PAO) é descrito da seguinte forma:

$$\text{minimizar } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.131)$$

$$\text{S.A. } \sum_{j=1}^n a_{ij} x_{ij} \leq b, \quad i = 1, \dots, m \quad (2.132)$$

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, n \quad (2.133)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n \quad (2.134)$$

Introduzindo uma cópia y_{ij} para cada variável x_{ij} podemos generalizar o (PAG) na forma $\overline{\text{PAG}}$:

$$\text{Minimizar } \alpha \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \beta \sum_{i=1}^m \sum_{j=1}^n c_{ij} y_{ij} \quad (2.135)$$

$$\text{S.A. } \sum_{j=1}^n a_{ij} x_{ij} \leq b_i, \quad i = 1, \dots, m \quad (2.136)$$

$$\sum_{i=1}^m y_{ij} = 1, \quad j = 1, \dots, n \quad (2.137)$$

$$x_{ij} = y_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n \quad (2.138)$$

$$x_{ij} \in (0, 1), \quad y_{ij} \in (0, 1), \quad i = 1, \dots, m \quad (2.139) \\ j = 1, \dots, n$$

onde α e β são parâmetros não negativos.

Observe que os problemas (PAG) e $\overline{\text{PAG}}$ descritos por (2.135) - (2.139) são equivalentes se $\alpha + \beta = 1$. No caso geral temos que

$$V(\text{PAG}) = (\alpha + \beta) V(\overline{\text{PAG}})$$

Relaxando as restrições de acoplamento (2.138) e incorporando-a na função objetivo de $\overline{\text{PAG}}$ com multiplicadores irrestritos λ_{ij} , vamos obter o seguinte problema relaxado $\overline{\text{PAG}}(\lambda)$.

$$\text{Min } \sum_{i=1}^m \sum_{j=1}^n (\alpha c_{ij} - \lambda_{ij}) x_{ij} + \sum_{i=1}^m \sum_{j=1}^n (\beta c_{ij} + \lambda_{ij}) y_{ij} \quad (2.140)$$

$$\text{S.A. } \sum_{j=1}^n a_{ij} x_{ij} \leq b_i, \quad i = \overline{1, m} \quad (2.141)$$

$$\sum_{i=1}^m y_{ij} = 1, \quad j = \overline{1, n} \quad (2.142)$$

$$x_{ij} \in (0, 1), \quad i = \overline{1, m} \\ j = \overline{1, n} \quad (2.143)$$

$$y_{ij} \in (0, 1), \quad i = \overline{1, m} \\ j = \overline{1, n} \quad (2.144)$$

Percebe-se facilmente que o $(\text{PAG}(\lambda))$ pode ser decomposto em dois subproblemas independentes, um com as variáveis x_{ij} e o outro com as variáveis y_{ij} como mostramos a seguir.

$$V(\text{PAGX}(\lambda)) = \text{Mínimo } \sum_{i=1}^m \sum_{j=1}^n (\alpha c_{ij} - \lambda_{ij}) x_{ij} \quad (2.145)$$

$$\text{S.A. } \sum_{j=1}^n a_{ij} x_{ij} \leq b_i, \quad i = 1, \dots, m \quad (2.146)$$

$$x_{ij} \in (0, 1), \quad i = 1, \dots, m \\ j = 1, \dots, n \quad (2.147)$$

$$V(\text{PAGY}(\lambda)) = \text{Mínimo } \sum_{i=1}^m \sum_{j=1}^n (\beta c_{ij} + \lambda_{ij}) y_{ij} \quad (2.148)$$

$$\text{S.A. } \sum_{i=1}^m y_{ij} = 1, \quad j = 1, \dots, n \quad (2.149)$$

$$y_{ij} \in \{0, 1\}, \quad i = 1, \dots, m \text{ e } j = 1, \dots, n \quad (2.150)$$

Temos que $V(\overline{\text{PAG}}(\lambda)) = V(\overline{\text{PAGX}}(\lambda)) + V(\overline{\text{PAGY}}(\lambda))$, $\forall A \in \mathbb{R}$

Além disso, $(\overline{\text{PAGX}}(\lambda))$ pode ser decomposto em m problemas da mochila.

O melhor limite de $V(\overline{\text{PAG}})$ é obtido resolvendo o seguinte problema dual (D).

$$V(D) = \underset{\lambda}{\text{máximo}} \{V(\overline{\text{PAG}}(\lambda))\} \quad (2.151)$$

Usando, por exemplo, os métodos do tipo subgradiente.

Finalmente, se $V(\text{RL1})$ e $V(\text{RL2})$ são os valores obtidos usando relaxação lagrangeana nas restrições (2.141) e (2.142) respectivamente, então valem as seguintes desigualdades:

$$V(\overline{\text{PAGL}}) \leq V(\text{RL1}) \leq V(\text{RL2}) \leq V(D) \leq V(\overline{\text{PAG}}) \quad (2.152)$$

LAPORTE, MERCURE e NOBERT (1986) descreveram um novo algoritmo exato para um (PRV) assimétrico descrito como um (PMCV) com restrições de capacidade.

O algoritmo desenvolvido é uma modificação do apresentado por *CARPANETO e TOIH (1980)*, onde inicialmente é resolvido um problema relaxado formado pelas restrições de alocação linear e posteriormente verificando outras restrições através do método Branch-and-Bound.

DEJAX e CRAINIC (1987) apresentaram outro resumo sobre o (FRV) enfatizando o papel das viagens ociosas no planejamento de viagens de uma frota de veículos.

Os autores analisaram a importância de um bom planejamento no setor de transportes sugerindo diversos procedimentos para minimizar este tipo de movimento de veículos.

DURAN (1987) apresentou uma nova solução para um problema de produção e distribuição de grande porte, utilizando-se de técnicas de decomposição.

O problema aplicativo refere-se a um conglomerado composto por fábricas de cerveja da Colômbia e a sua formulação matemática envolveu cerca de 85.000 variáveis sendo 2.600 variáveis inteiras zero-um e 14.000 restrições.

A solução obtida via MPSX/370-MIP da IBM proporcionou uma economia de 3,7% nos custos e foi obtido num tempo computacional considerado razoável.

GOLDEN e WASIL (1987) apresentaram sistemas para resolver um problema de roteamento de veículos de indústrias de bebidas, enquanto *YANO, CHAN e outros (1987)* desenvolveram novos softwares para o (PRV), onde o problema é formulado como um modelo de recobrimento.

LAPORTE, CHAPLEAU, LANDRY e MERCURE (1987) analisaram o problema de otimizar o serviço de coleta postal das várias caixas de correio existentes numa determinada região.

Os autores descrevem um novo procedimento do tipo *agrupar-rotear*, onde o roteamento é visto como um (PCV) generalizado.

No problema de agrupamento determinou-se quais as caixas de correio irão pertencer a cada rota e no problema de roteamento determinou-se a sequência de visitar a cada caixa contida em um agrupamento.

POIVIN (1987) descreveu um resumo sobre o (PRV) onde sempre sugere um novo sistema interativo para gerar rotas de um (PRV).

SALHI e RANDI (1987) desenvolveram uma nova técnica para o (PRV) baseado na combinação de métodos existentes na literatura.

Inicialmente o algoritmo gera uma solução inicial

usando heurísticas do tipo *CLARKE e WRIGHT (1964)* modificado, onde as economias são calculadas pela expressão:

$$s_{i,j} = d(\emptyset, i) + d(\emptyset, j) - a * d(i, j) \text{ com } 0 \leq a \leq 2$$

Outros procedimentos do algoritmo são:

i) método do tipo 3-ótimo de *LIN (1965)*

ii) uma rotina *COMBINE*, cuja função é combinar rotas testando a satisfação de restrições de capacidade e distância máxima, onde as combinações feitas são ordenadas através do cálculo das economias.

Além disso o algoritmo compõe-se de outros procedimentos complementares como as de proibir cruzamentos entre rotas, remover rotas ociosas, realocação de cliente para rotas etc.

Os resultados obtidos mostram uma pequena melhora na solução usando como solução inicial a melhor solução de várias heurísticas encontradas na literatura.

Finalmente, do ano de 1988, podemos citar os seguintes trabalhos sobre o (PRV).

CASCO, GOLDEN e WASIL (1988) fizeram uma análise sobre os problemas de roteamento de veículos com *BACKHAULS* que

traduzimos como pontos que antecedem os pontos de distribuição. Por exemplo, um supermercado é um ponto de distribuição e existem fornecedores deste ponto como vendedores de frutas, vegetais, criadores de aves que seriam os *BACKHAULS* do supermercado.

A literatura classifica este problema como uma extensão do problema de carregamento e despacho de veículos convencional, desde que aqui, o veículo pode carregar num ponto e descarregar noutro que não é necessariamente um ponto de distribuição.

DROR, LAPORTE e TRUDEAU (1988) escreveram outro resumo sobre a literatura existente do (PRV) com demandas estocásticas. Após a análise de várias metodologias existentes, os autores propõem uma nova formulação para este tipo de problema.

P5. O PROBLEMA DE ROTEAMENTO-LOCALIZAÇÃO (PRL)

O problema de gerar simultaneamente: rotas para o despacho de uma frota de veículos, e um conjunto de localizações de facilidades, é conhecido na literatura como o problema de roteamento e localização (PRL).

Estes problemas são estudados já há muitos anos mas na maioria das vezes *separadamente*, isto é, o (PRL) é particionado em dois problemas, sendo um específico sobre localização de facilidades (PL) e o outro tratando exclusivamente sobre o roteamento de veículos das facilidades para um conjunto de clientes (consumidores) (PRV).

As consequências do estudo em separado e as novas técnicas heurísticas e exatas do (PRL) foram apresentadas num trabalho recente de GILBERT LAPORTE (1989) " *A Survey of algorithms for locations-routing Problems*", do qual extraímos a síntese aqui exposto.

Inicialmente LAPORTE (1989) enfocou a necessidade de se construir algoritmos que tratassem simultaneamente do problema de roteamento e localização, citando para isso várias inconveniências de se aproximar o (PRL) por exemplo

como um (PL) ou como um (PRV).

De fato, a transformação de um (PRL) num (PL) significa aproximar o custo marginal do roteamento de veículos de facilidades para os clientes por uma função linear da distância radial entre o depósito e o cliente.

Vários autores, dentre eles *LAPORTE e NOBERT (1981)* mostraram que este tipo de aproximação pode levar muitas vezes a uma solução apenas sub-ótima, devido à não coincidência do ponto que minimiza a soma das viagens completas com o ponto que fornece a soma dos mínimos locais.

LAPORTE (1989) considera primordialmente os casos onde o conjunto de clientes e o conjunto de facilidades sejam ambos discretos e na maioria das vezes disjuntos.

Os sistemas de distribuição no (PRL) são representados por vários níveis (nível- λ), $\lambda \geq 2$, e no caso particular onde $\lambda = 3$, podemos identificar os três níveis como sendo:

nível 1 : facilidades primárias (fábricas)

nível 2 : facilidades secundárias (depósitos ou entrepostos)

nível 3 : usuários (clientes)

Os níveis 1 e 3 normalmente são compostos de pontos já

conhecidos e fixados. Por outro lado, os pontos do nível 2 juntamente com as rotas de distribuição, na maioria dos casos, constituem as variáveis de decisão do (PRL).

Quanto ao seu objetivo, o (PRL) normalmente requer a minimização de uma combinação linear dos custos de roteamento (fixos e variáveis) e dos custos de localização das facilidades.

Existem centenas de aplicações possíveis para o (PRL) e LAPORTE (1989) classifica-os em vários níveis e dentro destes níveis, em várias classes.

No (nível-2), o problema normalmente requer a localização de um ou mais depósitos e a designação de um conjunto de rotas ótimas dos depósitos para os clientes.

Dentre as aplicações deste nível citados por LAPORTE (1989) podemos destacar os seguintes:

OR e PIERSKALLA (1979) descreveram um problema de localização de *Bancos de Sangue* com o objetivo de determinar a localização de um número fixo de bancos de sangue dentre vários pontos candidatos e além disso, a alocação ótima de hospitais para estes bancos e finalmente a geração de rotas ótimas de despacho do sangue de bancos para os hospitais através de uma frota de veículos com restrições de capacidade e de número máximo de viagens diárias.

Uma outra aplicação do (nível-2) foi desenvolvido por *McLAIN, DURCHOLZ e WILBORN (1984)* que escreveram sobre o problema de localização de áreas para operações militares aéreas, de forma que o tempo de voo dentre as rotas que ligam estas áreas com um número de bases seja minimizado. O problema consiste inicialmente de 5 áreas de operações, 84 bases militares e entre 14 a 18 rotas de ligações.

Os problemas do (nível-3) são classificados em várias classes. Uma dessas classes é denominada $M(R,T)$ e é basicamente constituída de fábricas, depósitos e usuários, com caminhos de ligação entre as fábricas e os depósitos e de ciclos (caminhos de ida e volta) entre os depósitos e os usuários.

Dentre as aplicações dessa classe, podemos citar o trabalho de *WALTON-GANDY e DOHRN (1973)* que tratou do (PRL) de indústrias de alimentos e de bebidas, dimensionadas por 5 fábricas e 11 depósitos.

Os autores sugerem, após estudos, fechar 3 dos atuais depósitos e a abertura de 8 novas depósitos, o que resultaria num lucro adicional de $\$ 60.000$ por ano ao conglomerado.

Outra classe do (nível-3) descrito por *LAPORTE (1989)* é o $M(T,R)$ que difere das outras classes no tocante aos custos

de percorrer trechos entre os níveis 1 e 2 e os níveis 2 e 3.

Uma aplicação do $M(T,R)$ foi desenvolvida por *LABBÉ* e *LAPORTE (1986)* que consiste na localização ótima de caixas de correio numa área rural ou urbana.

A área global é dividida em várias zonas de dimensões menores com o prévio conhecimento do número de caixas existentes e do potencial de cada zona.

O problema consiste em determinar as localizações ótimas das caixas de correio que resultem na minimização de uma combinação linear de várias inconveniências pré-estabelecidas.

Uma diferença básica entre as classes $M(R,T)$ e $M(T,R)$ é que no último as decisões de localização são em relação às facilidades do 2.^o nível, enquanto no $M(R,T)$ são no primeiro nível.

Uma terceira classe de (PRL) de (nível-3) foi descrito por *LAPORTE (1989)* como $M(T,T)$.

Nesta classe ambos os tipos de veículos, os primários (que atendem do primeiro ao segundo nível) e os secundários (que atendem do segundo ao terceiro nível) geram trips completos (rotas de ida e volta).

Uma aplicação de $M(T,T)$ é o trabalho desenvolvido por *JACOBSEN e MADSEN (1978 e 1980)* que descreveram o problema da distribuição diária de aproximadamente 150.000 jornais que são despachados por veículos primários da oficina gráfica aos entrepostos e pelos veículos secundários que os despacham dos entrepostos às 4.500 bancas de jornais existentes.

Finalmente os problemas de roteamento-localização de quatro níveis (níveis-4) podem ser do tipo $M(R,R,T)$ que por sua vez, podem ser compostos por: fábricas, armazéns atacadistas, depósitos dos armazéns e consumidores.

Os algoritmos heurísticos existentes para o (PRL) normalmente exploram a idéia de decompor o problema global em problemas de localização, problema de alocação de facilidades para usuários e problema de roteamento de veículos. Portanto um algoritmo típico que se encaixa neste tipo de procedimento será descrito como uma reunião de rotinas de localização-alocação, do problema das p -medianas, de agrupamento e de roteamento de veículos.

Quanto ao uso das técnicas exatas para o (PRL) existem dois tipos de formulação matemática para este problema, sendo que um deles envolve variáveis com dois índices e o outro, variáveis com três índices.

Um exemplo de formulação matemática para o (PRL) com variáveis de dois índices pode ser da forma:

- Problema simétrico

- Um único depósito a localizar

- $N = \{1, \dots, n\}$ conjunto dos usuários

- $I \subseteq N$ candidatos à localização do depósito

- m = número de veículos disponíveis

- K é o conjunto das frotas de veículos existentes

$$y_i = \begin{cases} 1, & \text{se e só se, a facilidade for alocada em } i \\ 0, & \text{c/c.} \end{cases}$$

x_{ij} = número de vezes que o arco (i, j) ou (j, i) for utilizado por um veículo na solução final.

$$x_{ij} = \begin{cases} 0, 1, 2, & \text{se } i \text{ ou } j \in I \\ 0, 1, & \text{se } i \text{ e } j \in N-I \end{cases}$$

$$\text{Minimizar } \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$$

(2.152)

$$\text{S.A.} \quad \sum_{i \in N} y_i = 1 \quad (2.153)$$

$$\sum_{i \ll k} x_{ik} + \sum_{j \gg k} x_{kj} = 2 + 2(m-1)y_k, \quad k \in I \quad (2.154)$$

$$\sum_{i \ll k} x_{ik} + \sum_{j \gg k} x_{kj} = 2, \quad k \in N-I \quad (2.155)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq (1/2)\Delta(S) - 1, \quad (S \subset N, 2 \leq |S| \leq n-2) \quad (2.156)$$

$$y_i \in (0, 1), \quad i \in I \quad (2.157)$$

$$x_{ij} = \begin{cases} 0, 1 \text{ ou } 2, & \text{se } i \text{ ou } j \in I \\ 0, 1, & \text{se } i \text{ e } j \in N-I \end{cases} \quad (2.158)$$

onde $\Delta(S) = \sum_{i \in S} \left(\sum_{k \gg i} x_{ik} + \sum_{k \ll i} x_{ki} \right)$ é o grau de S .

Finalmente, as soluções do (PRL) via técnicas exatas se baseiam principalmente em relaxações lagrangeanas e programação dinâmica com relaxações.

I.2) O PROBLEMA DO SCHEDULING DE VEÍCULOS (PSV)

Segundo *BODIN, GOLDEN, ASSAD e BALL (1983)*, um problema do *SCHEDULING* de veículos é um problema de roteamento de veículos (PRV) com restrições adicionais de tempo.

Assim, um (PSV) tem como objetivo, gerar um conjunto de *schedules* viáveis a um menor custo e/ou tempo possível.

Um *schedule* é definido como sendo uma sequência de visitas ordenadas no espaço e no tempo a serem efetuadas por um veículo.

As restrições de tempo mais comumente encontradas num (PSV) são: tempo para iniciar ou terminar uma tarefa (trip), tempo limite para concluir uma rota, restrições de precedências, etc.

A presença de intervalos de tempo *time-windows* num (PRV) ou (PSV) definem uma outra classe de problemas de roteamento que analisaremos no item 1.3) deste trabalho.

Existem na literatura centenas de aplicações do (PSV), e dentre elas podemos citar como as mais populares o problema de *scheduling* de aviões e tripulações aéreas e o problema de *scheduling* de ônibus escolar.

Listamos a seguir, numa *ordem cronológica*, parte da literatura existente sobre o (PSV).

Na pesquisa que realizamos sobre a literatura existente sobre o (PSV), indubitavelmente o problema da programação de vôos e/ou de tripulações aéreas foi a aplicação mais

encontrada.

Um dos primeiros *survey* sobre o problema de programação de tripulações aéreas foi escrito por *ARABEYRE, FEARNLEY, STEIGER e TEATHER (1969)*.

Neste trabalho, os autores concluíram que devido às suas dimensões elevadas, este tipo de problema é normalmente resolvido através de procedimentos heurísticos, sendo muito restrito o uso de técnicas exatas conhecidas na época.

O problema de scheduling de tripulações aéreas, segundo os normalmente envolve os seguintes itens:

i) SEGURANÇA.

ii) compatibilidade dos itens que compõem um voo.

iii) A política da companhia aérea.

iv) Uma busca econômica sólida.

Numa programação matemática, os itens i), ii) e iii) são normalmente colocados como restrições e iv) como o objetivo a ser otimizado.

Uma solução do problema consiste normalmente de *rotações* ou uma sequência de *trips* ou ainda *schedules*

fechados (supõe-se que cada tripulação tenha uma base de onde elas partem para efetuar atendimentos e para retornar após o seu cumprimento).

Neste survey, os autores afirmaram que usualmente o problema é descrito como um Problema de Programação Linear Inteira com variáveis zero-um onde as matrizes de restrições possuem uma estrutura particular.

A partir desta configuração, desenvolveu-se o perfil deste trabalho que cobre basicamente os seguintes itens referentes à sua matriz de restrições:

1. FORMULAÇÃO
2. GERAÇÃO
3. CUSTOS DA TRIPULAÇÃO
4. OTIMIZAÇÃO
5. ACOPLAMENTO DAS ETAPAS DE PLANEJAMENTOS ADJACENTES.

Dentre estes itens, destacamos aqui o processo de formulação e de otimização da matriz de restrições.

As formulações matemáticas normalmente seguem um modelo de recobrimento (*set covering* ou *set partitioning model*)

onde a matriz de restrições $A = (a_{ij})$ é formada por componentes zero-um e definidas como:

$$a_{ij} = \begin{cases} 1, & \text{se a tripulação } i \text{ está no schedule } j \\ 0, & \text{caso contrário} \end{cases}$$

Dentre as aplicações, os autores citaram vários exemplos ligados a companhias aéreas descrevendo suas dimensões iniciais, dimensões obtidas após o uso de técnicas de reduções, a função objetivo e o modelo utilizado (PP) = Problema de particionamento ou (PR) = Problema de recobrimento, conforma nos mostra a tabela 2.2.1 a seguir:

COMPANHIA	DIMENSÃO INICIAL DA MATRIZ A		DIMENSÃO MATRIZ REDUZIDA		FUNÇÃO OBJETIVO	MODELO USADO (PP) OU (PR)
	LIN. (m)	COL. (n)	m	n		
SWISSAIR	200	até 5000	200	2000	otimizar o n.º de tripulações, n.º paradas noturnas e os tempos de serviço	(PP)
AIR FRANCE	—	—	150	1500	custos e penalidades	(PP)
AER-LINGUS	—	—	36	300	número de tripulações	(PP)
AMERICAN AIRLINES	100	500	100	300	minimizar salários e custos em hotéis	(PP) programação diária (PR) programação semanal
BEA	—	—	250	3000	n.º de tripulações	(PR)
BOAC	600	600	—	—	*	(PR) e (PP)
LUFTHANSA	*	*	450	800	n.º de tripulações	(PP)
UNITED AIRLINES	611	10000	600	8000	n.º de tripulações, tempo de voo e custos refeição	(PP)
AIR CANADA	75-100	30000	75-100	3000	custos e número de tripulações	(PP)
KLM	—	—	60	100	n.º de hs creditadas	(PP)
SAS	200	2000	200	2000	n.º de tripulações e outros	(PP) e (PR)

(*) = dados não divulgados

(TABELA 2.2.1)

Quanto ao ítem *OTIMIZAÇÃO*, várias técnicas são utilizadas como nos mostra a tabela 2.2.2. a seguir:

COMPANHIA	TÉCNICA DE OTIMIZAÇÃO UTILIZADA
SWISSAIR	Enumeração combinatorial e busca exaustiva
AIR FRANCE	Para prob. reduzidos-algoritmos combinatoriais Para prob. com $n < 100$ -> busca exaustiva Para prob. com $n=350-500$ -> Programação Linear e enumeração Para prob. com $n > 500$ -> Enumeração parcial e busca direta
AER-LINGUS	Branch-and-Bound
AMERICAN AIRLINES	Branch-and-Bound e Enumeração Implícita
BEA	*
BDAL	Branch-and-Bound
LUFTHANSA	Branch-and-Bound
UNITED AIRLINES	Cortes de Gomory
AIR CANADA	Branch-and-Bound
KLM	Branch-and-Bound
SAS	Planos de Cortes

(*) = dados não fornecidos

(TABELA 2.2.2.)

Outra aplicação muito popular no (PSV) se refere ao problema de scheduling de Ônibus numa zona urbana.

Um dos trabalhos pioneiros foi desenvolvido por SAHA

(1970) que dividiu uma cidade em várias zonas e alocou uma frota de Ônibus para cada zona.

Cada frota possuía aproximadamente 300 Ônibus e o objetivo do problema foi a de atender a todos os trips do problema com um menor número possível de Ônibus.

SALZBORN (1970) desenvolveu um método matemático para minimizar o número de trens necessários para cobrir os horários de pico de um sistema ferroviário e também o tempo total das viagens.

Foi mostrado inicialmente que o número total de trens requerido é igual ao número máximo de trens que podem operar simultaneamente no sistema. A partir deste resultado, formulou-se um modelo de Programação Linear Inteiro zero-um para obter o tempo total mínimo para efetuar o conjunto de viagens.

APPELGREN (1971) desenvolveu duas técnicas de Programação Inteira para resolver o problema de roteamento e scheduling de navios.

O problema consiste em gerar sequências ótimas de carregamentos para uma frota de navios num determinado horizonte de planejamento.

Cada carregamento é caracterizado por sua receita,

tipo, tamanho, portos de carregamento e descarregamentos e tempos de carregamento e descarregamento. A data do carregamento pode ser um intervalo de tempo *time-window* e neste caso, a data de descarregamento será uma variável dependente da data do carregamento.

A frota consiste de navios que são caracterizados pelo seu tamanho, capacidade de transportar tipos de cargas, velocidade, posição inicial e um *valor do tempo* que é dado por uma receita diária relativa esperada de um carregamento efetuado.

Os algoritmos descritos neste trabalho, utilizam-se de dois métodos de Programação Linear Inteira. O primeiro é um algoritmo de cortes que entretanto não conseguiu resolver os problemas testes com uma estrutura mais complexa. O segundo método de solução é um algoritmo de Branch-and-Bound que nos testes realizados se mostrou muito eficiente para problemas com 10 a 15 navios e 15 a 25 carregamentos.

HAUER (1971) estudou o problema de escolher o melhor tipo de ônibus (convencional, micro-ônibus, articulado, elétrico ou de duplo pavimento) no serviço de transporte público onde é aceito qualquer tipo de ônibus. Neste trabalho foi apresentado um algoritmo para orientar a seleção do tipo e do tamanho da frota, baseado numa redução sistemática do conjunto de opções de escolha, usando conceitos de *ótimo de Pareto*.

LEVIN (1971) descreveu um problema de roteamento e scheduling de vôos de uma companhia aérea.

O problema foi formulado como um modelo de Programação Inteira zero-um e resolvido por um algoritmo do tipo *Land-and-Doig* (1960).

ANGEL, CAUDLE, NOONAN e WHINSTON (1972) trataram do problema de transporte de estudantes das suas casas até a escola e vice-versa.

O problema teve como objetivo, minimizar o número de schedules respeitando as restrições de espaço e de tempo.

Os autores lembraram que existe na literatura, vários métodos para resolver este tipo de problema como os procedimentos de enumeração ou os algoritmos para resolver o problema do caixeiro viajante (PCV) modificados, entretanto eles recordam que na maioria das vezes, estes métodos se mostram ineficientes para tratarem de problemas com um elevado número de paradas.

Partindo deste princípio, os autores desenvolveram um novo algoritmo aproximado que satisfaz todas as restrições e ao mesmo tempo reduz as dimensões do problema.

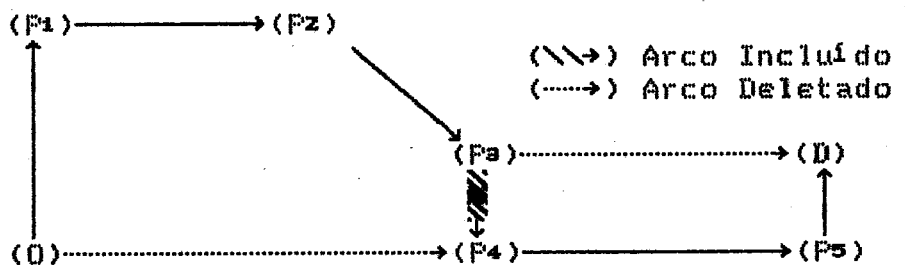
As soluções obtidas foram comparadas com métodos

manuais utilizados na prática e se mostraram sempre iguais ou superiores a estes.

BENNETT e GAZIS (1972) adaptaram o algoritmo das economias de *CLARKE e WRIGHT (1964)* para o problema de scheduling de uma frota de Ônibus escolar.

Inicialmente foi alocado para cada parada um número de Ônibus suficiente para o seu pleno atendimento.

Numa segunda etapa, efetua-se síntetizações entre dois schedules desde que o schedule resultante não viole a restrição de capacidade do Ônibus. O procedimento de síntetização é uma versão atualizada do algoritmo original onde são deletados o último arco de um dos schedules e o primeiro arco do outro schedule, adicionando entre eles, um novo arco conectando os dois schedules como mostra o exemplo dado a seguir:



(R1) : Schedules Originais: $0 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow D$ e $0 \rightarrow P4 \rightarrow P5 \rightarrow D$

(R2) : Schedules Resultantes: $0 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P5 \rightarrow D$

FIGURA 2.5

As economias ao sintetizar as rotas R1 e R2 nos pontos P3 e P4 são dados por:

$$s_{P_3P_4} = T_{P_3D} + T_{OP_4} - T_{P_3P_4}, \text{ onde } T_{ij} = \text{tempo de percorrer o arco } (i, j).$$

Finalmente para satisfazer a regra de arcos a deletar (o último de um e o primeiro do outro), os autores sugeriram em alguns casos, um ressequenciamento dos schedules já existentes, se isso for conveniente.

TIBREWALA, PHILIPPE e BROWNE (1972) desenvolveram um algoritmo muito simples que produz solução ótima para o problema de scheduling de homens ou equipamentos em vários turnos de serviço, com a condição de que eles devem folgar dois períodos consecutivos no ciclo analisado.

BAKER, CRABILL e MAGAZINE (1973) formularam um problema de alocação no tempo de funcionários de uma empresa, sujeito a uma variedade de restrições, como um Problema de Programação Linear Inteira. A formulação possui uma matriz de restrições com uma estrutura particular que permite a sua solução de forma rápida e eficiente. Foram analisados relações entre a solução ótima e a flexibilidade nos dados de entrada.

GAVISH e SCHWEITZER (1974) apresentaram uma nova

formulação para o problema de roteamento e scheduling de veículos. O problema é inicialmente formulado como um problema de transporte com restrições adicionais de compatibilidade (PTA) para posteriormente ser reduzido a um (PT) clássico, permitindo com isso, a solução de problemas com até 1.000×1.000 em menos de um minuto de tempo computacional.

O (PTA) teve a seguinte formulação matemática:

x_{ij} = número de veículos que após completar a tarefa de i , viaja diretamente de i para j .

m_i^c = número de veículos liberados após completar a tarefa i .

m_j^o = número de veículos requeridos para a origem da tarefa j .

t_j^c = tempo para os veículos efetuarem a tarefa j .

t_j^o = tempo gasto pelos veículos na origem de j .

t_{ij} = tempo para percorrer diretamente o arco (i, j) .

h_{ij} = margem de segurança do tempo t_{ij}

w_j = tempo máximo de espera de um veículo em j .

c_i^c = custo de viagem de um veículo do final de i para o depósito mais próximo.

a_j^o = custo de viagem de um veículo da tarefa inicial j para o depósito mais próximo.

b_{ij} = custo de viagem de um veículo do final de i para a origem de j .

$s_{ij} = a_i^c + a_j^o - b_{ij}$ = economia de custo por veículo se as tarefas i e j forem combinadas.

FORMULAÇÃO MATEMÁTICA

$$\text{Maximizar } \sum_{i=1}^{n_c} \sum_{j=1}^{n_o} s_{ij} x_{ij} \quad (2.159)$$

$$\text{S.A. } \sum_{i=1}^{n_c} x_{ij} \leq m_j^o, \quad j = 1, \dots, n_o \quad (2.160)$$

$$\sum_{j=1}^{n_o} x_{ij} \leq m_i^c, \quad i = 1, \dots, n_c \quad (2.161)$$

$$x_{ij} = 0 \text{ se } t_i^c + t_{ij} + h_{ij} \geq t_j^o, \quad i = 1, \dots, n_c \quad (2.162)$$

$$j = 1, \dots, n_o$$

$$x_{ij} = 0 \text{ se } t_i^c + t_{ij} \leq t_j^o - w_j, \quad i = 1, \dots, n_c \quad (2.163)$$

$$j = 1, \dots, n_o$$

$$x_{ij} = 0 \text{ se } Q_{ij} = 0 \quad i = 1, \dots, n_c \quad (2.164)$$

$$j = 1, \dots, n_o$$

$$x_{ij} = 0 \text{ se } M_{ij} = 0 \quad i = 1, \dots, n_c \quad j = 1, \dots, n_o \quad (2.165)$$

Onde $Q_{ij} = \begin{cases} 1, & \text{se as cargas do nó } j \text{ e } i \text{ são compatíveis} \\ & \text{para serem transportadas num mesmo veículo} \\ 0, & \text{caso contrário} \end{cases}$

$M_{ij} = \begin{cases} 1, & \text{se as características das tarefas } i \text{ e } j \text{ são} \\ & \text{compatíveis} \\ 0, & \text{caso contrário} \end{cases}$

Neste trabalho, os autores propuseram transformar o (FTA) num (FT) clássico, gerando inicialmente o conjunto de soluções viáveis para as restrições (2.162) - (2.165).

CHISMAN (1975) estudou uma nova variação do (PCV) clássico onde o caixeiro deve não só visitar um conjunto de N cidades exatamente uma única vez e retornar ao seu ponto de origem, mas onde um subconjunto (agrupamento) de cidades devem ser visitadas contiguamente.

Por exemplo, se as cidades i, j, k formam um desses agrupamentos, elas devem ser visitadas em uma dessas ordens:

$(i-j-k), (i-k-j), (j-i-k), (j-k-i), (k-i-j)$ ou $(k-j-i)$

A ordem de visitas em cada agrupamento não é especificada e a formulação desta nova versão consiste das restrições do (PCV) clássico adicionado de restrições do tipo:

$$\sum_{i \in C_k} \sum_{j \in C_k} x_{ij} = |C_k| - 1, \text{ onde } C_k \text{ é um agrupamento com } k$$

cidades e $|C_k|$ é a cardinalidade de C_k .

RICHARDSON (1976) desenvolveu um novo procedimento ótimo para o problema do scheduling de aviões. O problema consiste em determinar a sequência de paradas de cada avião de uma base para um conjunto de cidades destinos,

retornando ao seu final, a uma das duas bases existentes.

O problema foi modelado como um problema de Programação Inteira e a solução obtida através do método de Decomposição de BENDERS.

ORLOFF (1976b) analisou uma classe de problemas de scheduling de veículos e mostrou que a versão com restrições de tempo fixadas pode ser facilmente resolvido usando uma nova formulação para o problema.

Quanto à versão do (PSV) com intervalos de tempo o autor mostrou evidências da sua dificuldade de obter resultados satisfatórios com o uso de métodos exatos.

ARISAWA e ELMAGHRABY (1977 a,b) escreveram dois trabalhos sobre o (PSV). No primeiro artigo os autores analisaram o caso míope, ou seja, o problema que considera somente dois horizontes de planejamento diários, o atual (hoje) e o posterior (amanhã). O problema teve como objetivo minimizar penalidades operacionais esperadas consistindo basicamente dos custos de movimentos ociosos e os custos de atraso nas entregas.

Em dois casos particulares, a solução se resumiu a uma análise de custos marginais e no caso geral, foi proposto um modelo de fluxos. A aplicação nos dois trabalhos consistiu de um problema de despacho de caminhões de um depósito

central para vários terminais de descarga.

No caso míope, foi proibido o tráfego entre terminais, restrição esta relaxada no caso dinâmico.

MARQUEZ e MEDINA-MORA (1977) estudaram um problema do Banco Central do México que requer uma frota de veículos para transportar dinheiro e outros valores entre várias cidades do país.

Para isto existem vários tipos de veículos disponíveis para efetuarem um conjunto de trips completos (circuitos).

A partir de um conjunto de informações fornecidas pelo Banco, os autores propuzeram uma formulação semelhante ao problema de transporte, onde o fluxo em alguns arcos está condicionado à existência de fluxos em outros arcos da rede.

POLLACK (1977) desenvolveu um trabalho minucioso sobre os trabalhos teóricos desenvolvidos para o problema de planejamento de vôos de uma companhia aérea.

Neste artigo, o autor enfatizou que apesar de muito estudado, este tipo de problema não é bem aceito na prática devido à excessiva simplificação adotada nos modelos desenvolvidos, desconsiderando itens essenciais no processo de planejamento de vôos como: condições de continuidade de etapas de planejamentos, variáveis inteiras, isolamento de

cada avião, custos não lineares e função rendimento, fluxo de passageiros, nível mínimo de serviços, múltiplos critérios, dimensões do problema dentre outros.

ASGHARZADEH e NEWELL (1978) desenvolveram uma nova estratégia para resolver o problema de despacho de veículos de transporte público a partir de uma garagem central. A frota de N veículos deve sair da garagem para visitar um conjunto de pontos de parada e retornar à sua origem após atender um determinado número de passageiros dentro de um período de tempo. Os passageiros devem ser transportados dos pontos de parada para os seus respectivos destinos e o objetivo foi o de minimizar a média de tempo de espera dos passageiros nas paradas.

BODIN e KURSH (1978) descreveram um método computacional para determinar percursos quase ótimos para o problema da limpeza de ruas feita por vassouras mecânicas. Os percursos gerados por procedimentos heurísticos, envolvem restrições espaciais e temporais. O método utilizado se baseia em heurísticas das classes "agrupar-rotear" e "giant-route".

GAVISH, SCHWEITZER e SHLIFER (1978) descreveram um problema de scheduling de uma frota de ônibus numa região metropolitana composto de centenas de trips relativamente curtos.

Os autores desenvolveram um algoritmo exato que minimiza a soma dos custos operacionais, sendo testada para um problema real com cerca de 4500 Ônibus, produzindo soluções com até 10 % de economia.

Neste trabalho, o problema de scheduling de Ônibus para trips foi dividido em duas etapas.

Na etapa inicial o objetivo foi a de combinar os trips sequencialmente para formar schedules e na segunda etapa, designou-se os Ônibus para as garagens existentes.

NOTAÇÃO DO PROBLEMA

a_i = tempo correspondente ao final do trip i no horizonte de planejamento.

b_j = tempo correspondente ao início do trip j no horizonte de planejamento.

onde $a_1 \leq a_2 \leq \dots \leq a_n$ e $b_1 \leq b_2 \leq \dots \leq b_n$ e $b_j \geq a_i$ para $j > i$

$$x_{ij} = \begin{cases} 1, & \text{se o final do trip } i \text{ é conectado com o início trip } j \\ 0, & \text{caso contrário.} \end{cases}$$

$$x_{n+1,j} = \begin{cases} 1, & \text{se a garagem fornece um Ônibus para início} \\ & \text{trip } j \\ 0, & \text{c/c.} \end{cases}$$

$$x_{i,n+1} = \begin{cases} 1, & \text{se após o final do trip } i, \text{ o Ônibus retorna à} \\ & \text{garagem} \\ 0, & \text{c/c} \end{cases}$$

$$Q_{ij} = \text{custo} = \begin{cases} D, & \text{se } i = n + 1, j = 1, 2, \dots, n \\ 0, & i = 1, 2, \dots, n + 1, j = n + 1 \\ L_{ij} = E_{ij} + K_{ij} + R_{ij} + W_{ij}, & i, j = 1, \dots, n. \end{cases}$$

onde D = economia do atual horizonte de planejamento por redução de uma unidade de Ônibus.

E_{ij} = custo de viagem ociosa entre i e j incluindo custos operacionais e diretos.

K_{ij} = penalidade por não usar o arco (i,j)

$$R_{ij} = \begin{cases} R \geq 0, & \text{se os trips } i \text{ e } j \text{ pertencem ao mesmo} \\ & \text{centro de controle} \\ 0, & \text{c/c.} \end{cases}$$

FORMULAÇÃO MATEMÁTICA DA PRIMEIRA ETAPA

$$\text{Minimizar} \quad \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} Q_{ij} x_{ij} \quad (2.166)$$

$$\text{S.A.} \quad \sum_{i=1}^{n+1} x_{ij} = 1, \quad j = 1, \dots, n \quad (2.167)$$

$$\sum_{j=1}^{n+1} x_{ij} = 1, \quad i = 1, \dots, n \quad (2.168)$$

$$\sum_{j=1}^{n+1} x_{n+1,j} = n \quad (2.169)$$

$$\sum_{i=1}^{n+1} x_{i,n+1} = n \quad (2.170)$$

$$x_{ij} (b_j - a_i - t_{ij} - t_1) \geq 0, \quad i, j = 1, 2, \dots, n \quad (2.171)$$

$$x_{ij} (b_j - a_i - t_{ij} - t_2) \leq 0, \quad i, j = 1, 2, \dots, n \quad (2.172)$$

$$x_{ij} \geq 0 \text{ e inteiros } i, j = 1, \dots, n+1 \quad (2.173)$$

As restrições (2.171) e (2.172) afirmam que a conexão do final do trip i com o início do trip j só é possível se $t_1 \leq b_j - a_i - t_{ij} \leq t_2$, $i, j = 1, \dots, n$.

FORMULAÇÃO MATEMÁTICA DA SEGUNDA ETAPA

$$\text{minimizar } \sum_{i=1}^k \sum_{j=1}^l c_{ij} y_{ij} \quad (2.174)$$

$$\text{S.A. } \sum_{j=1}^l y_{ij} = 1, \quad i = 1, \dots, k \quad (2.175)$$

$$\sum_{i=1}^k y_{ij} \leq d_j, \quad j = 1, \dots, l \quad (2.176)$$

$$y_{ij} \in (0, 1), \quad \forall_i = 1, \dots, k, \quad \forall_j = 1, \dots, l \quad (2.177)$$

onde k = número de schedules existentes

l = número de garagens existentes

c_{ij} = custo de alocar o schedule i para um Ônibus da garagem j .

$$y_{ij} = \begin{cases} 1, & \text{se o schedule } i \text{ for alocado para um} \\ & \text{veículo da garagem } j \\ 0, & \text{c/c.} \end{cases}$$

KAO (1978) analisou um (PCV) com tempo de percurso estocástico e cujo objetivo é o de obter um schedule com maior probabilidade de concluí-lo dentro de um período de tempo especificado. A solução proposta foi um procedimento de Programação Dinâmica envolvendo técnicas de

Branch-and-Bound.

COOK e RUSSEL (1978) analisaram uma generalização do problema de despacho de veículos com restrições adicionais de tempo e demandas estocásticas. A solução foi obtida através de um procedimento heurístico usando simulação GPSS.

PICARD e QUEYRANNE (1978) descreveram o problema do caixeiro viajante com restrições adicionais de tempo (PCVRT) onde o custo de cada transição depende não somente dos dois destinos envolvidos mas também das suas respectivas posições na sequência que define o schedule.

O (PCVRT) foi formulado como um Problema de Programação Inteira em quatro versões diferentes.

STEIN (1978b) apresentou um estudo analítico de aspectos básicos que envolvem o problema *Dial-a-Ride*. Trabalhando com modelos mais simples, o autor descreveu novos algoritmos onde as suas performances podem ser probabilisticamente determinados no sentido assintótico.

BAKER, BODIN, FINNEGAN e PONDER (1979) desenvolveram um novo procedimento heurístico para o problema do planejamento de tripulações aéreas (PPTA) da Federal Express Corporation de Memphis, Tennessee. O (PPTA) normalmente assume a existência de um conjunto fixo de vôos para os quais devem ser alocados tripulações.

Neste trabalho, foram utilizados algoritmos de roteamento e scheduling de veículos para gerar soluções aproximadas para o (PPTA). Desta forma cada voo foi interpretado como um cliente de demanda igual ao número de pessoas que compõe a sua tripulação. Cada componente desta tripulação é visto como um "veículo" a ser despachado, de capacidade finita, desembarcando uma unidade para cada cliente visitado. O local onde cada tripulação está baseado representa a origem dos schedules dos veículos.

Portanto supondo que as tripulações retornem à sua base após efetuarem um schedule, o objetivo do problema passa a ser encontrar um conjunto de schedules de custo mínimo satisfazendo as suas restrições de espaço e de tempo.

O trabalho atende basicamente três metas do (PPTA).

- Construir Schedules.
- Melhorar Schedules.
- Efetuar composições entre Schedules.

Para o problema de construção de Schedules, foi utilizado uma adaptação da heurística proposta por *BODIN, KYDES e ROSENFELD (1975)*.

Os procedimentos de geração de bons schedules, se mostraram dispendiosos na prática o que provocou a necessidade de elaborar esquemas de melhorar schedules, baseados nos métodos 2-ótimos e 3-ótimos de LIN (1965) e os k-ótimo de LIN e KERNIGHAN (1973).

BODIN e BERMAN (1979) desenvolveram um novo procedimento para resolver o problema de roteamento e scheduling de uma frota de Ônibus escolar, com o objetivo de:

-minimizar o custo total de transporte

-minimizar o tempo médio de transporte

Os schedules devem ser gerados diariamente e a solução encontrada se mostrou 20% mais econômica em relação à solução que estava sendo utilizada na prática.

GIANNESI e NICOLETTI (1979) descreveram um novo procedimento para solucionar o problema de scheduling de tripulações aéreas que segundo os autores, ocupa cerca de 10% dos custos totais de uma companhia aérea, se posicionando como o maior custo unitário após os custos dos combustíveis. Desta forma, o trabalho enfoca a importância de uma solução que fornece um percentual de economia, lembrando que em determinados casos práticos, 1% representou até \$600.000 de economia anual. O problema estudado neste

artigo foi formulado como um (PMCV) com restrições adicionais de tempo.

SCHULTZ (1979) descreveu um novo algoritmo heurístico para scheduling de veículos baseado no método de *CLARKE e WRITE (1964)*. O algoritmo é muito simples e permite um roteamento manual sem a necessidade de usar um computador, para resolver pequenos problemas.

Nos vários testes realizados, os resultados segundo o seu autor foram encorajadores, levando-se em conta a sua simplicidade.

WOLTERS (1979) apresentou um problema de transporte aéreo envolvendo oito bases. Cada base possui uma quantidade conhecida de bens que devem ser transportados para as outras sete bases. São conhecidos também o tempo de viagem entre duas bases distintas e a capacidade de cada avião.

O transporte pode ser feito diretamente ou passando por bases intermediárias, dentro do período de tempo pré-estabelecido (22 horas, 5 horas).

A solução deste problema foi obtida por etapas; inicialmente gerou-se um limite inferior para o número de aviões necessários para transportar todos os bens, a seguir resolveu-se um problema de programação linear misto que minimiza o tempo total de viagem para transportar todos os

bens e finalmente na última etapa alocou-se aviões para schedules.

DESROSIERS, FERLAND, ROUSSEAU, LAPALME e CHAPLEAU (1980) desenvolveram um sistema modular automático (*TRANSCOL*) para planejar e organizar o transporte de estudantes de uma área contendo aproximadamente 50 escolas e 20.000 estudantes.

O sistema *TRANSCOL* inclui quatro módulos que são executados sequencialmente e cujas descrições são resumidas a seguir:

i) Existe um módulo inicial *LOCATION* que especifica os dados iniciais para os outros módulos.

ii) Módulo *BUS ROUTE GENERATOR* que constrói dois tipos de schedules.

- Nas áreas urbanas, os estudantes são embarcados e desembarcados nas paradas existentes ou geradas, neste último caso, é resolvido uma variante do (PCV).

- Nas áreas rurais, os estudantes são embarcados e desembarcados em frente às suas casas e o problema neste caso é descrito como uma variação do (PCC).

iii) *SCHOOL SCHEDULING* especifica schedules para

escolas com o objetivo de maximizar o número de rotas que pode ser executado sequencialmente pelo mesmo Ônibus.

iv) *ROUTE SCHEDULING* determina segmentos das rotas a serem executadas sequencialmente por um Ônibus.

DULAC, FERLAND e FORGUES (1980) também analisaram o problema de roteamento e scheduling de Ônibus escolar para um conjunto de bairros urbanos.

O procedimento foi descrito em quatro passos que resumimos a seguir:

Passo 1: Cada estudante é alocado para uma parada mais próxima à sua casa. Normalmente esta parada é dos dois nós incidentes do arco onde está localizada a sua residência.

Passo 2: Um conjunto de paradas é determinado para cada escola ou grupos de escolas.

Passo 3: Os estudantes são alocados para as paradas.

Passo 4: São determinados conjuntos de rotas para cada escola (grupo de escolas) independentemente.

Devido à complexidade do problema, os autores utilizaram heurísticas tanto para designar estudantes para paradas quanto para gerar schedules de Ônibus.

Existem na literatura, várias generalizações para o (PCV) dentre elas as que incluem capacidade do caixeiro (veículo), relações de precedências ou restrições adicionais de tempo.

Duas novas generalizações do (PCV) foram apresentadas por GOLDEN, LEVY e DAHL (1981) que propuseram também procedimentos heurísticos para a sua solução.

A primeira generalização do (PCV) foi denominada O *Problema do Viajante de Compras (PVC)* descrito da seguinte forma: Existe uma origem s , um conjunto $I = \{1, 2, \dots, m\}$ de mercados e um conjunto $K = \{1, 2, \dots, n\}$ de produtos que devem ser adquiridos pelo viajante nestes mercados.

O (PVC) tem como objetivo gerar um ciclo passando pela origem s e por um subconjunto $J \subset I$ de modo que cada um dos n produtos sejam adquiridos em um dos mercados $J \subseteq I$, minimizando os custos da distância total percorrida pelo viajante e os custos de compra dos n produtos.

Foi assumido que:

1) Cada um dos n produtos está disponível em pelo menos um mercado.

2) O viajante pode passar por um mercado mais de uma

vez sem adquirir produtos.

3) Os produtos não estão disponíveis na origem s . Além disso, foram definidas duas matrizes:

$D = (d_{ik})$, onde d_{ik} = custo do produto k no mercado i .

$C = (c_{ij})$, onde c_{ij} = custo de viajar de i para j diretamente.

Se o produto k não está disponível no mercado i então $d_{ik} = \text{máximo}(d_{ik}, c_{ij})$.

Finalmente, no caso particular onde $m = n$, cada mercado dispõe de uma única unidade de um único produto diferente das demais e restringindo a visita a cada mercado em uma única vez, então o (PVC) se torna o (PCV).

Neste trabalho, os autores propuseram uma heurística do tipo *savings* para o (PVC) que resumimos a seguir:

HEURÍSTICA PARA O (PVC)

Passo 1: Determinar a matriz das distâncias mínimas entre dois mercados

$$\bar{C} = (\bar{c}_{ij}) \quad i = 1, \dots, m, \quad j = 1, \dots, m$$

Passo 2: Gerar o schedule inicial.

Encontrar o mercado i^* que vende o maior número de produtos da lista de compras a um menor preço global. Isto é, encontre o mercado i^* que minimiza $\sum_{k=1}^m d(i, k)$

Formar a rota inicial: $R : s \rightarrow i^* \rightarrow s$.

Passo 3: Determinar o menor preço de cada produto de k dentre os mercados de R .

$$f(R, p) = \underset{R}{\text{mínimo}} \{d(i, p), \forall p \in K\}$$

Comparar $f(R, p)$ com o menor preço de p nos mercados $K-R$

$$g(R, q, p) = \underset{K-R}{\text{máximo}} \{f(R, p) - d(q, p); 0\}, \quad \forall p \in K \\ \forall q \in K-R$$

Passo 4: Busca da maior economia

Encontrar o mercado $q^* \in K-R$ e os mercados

adjacentes i^* e $j^* \in R$ tais que maximizem o valor:

$$S(i, j, q) = \bar{c}_{ij} - \bar{c}_{iq} - \bar{c}_{qj} + \sum_{p=1}^n g(R, q, p)$$

Faça $S(i^*, j^*, q^*) = \text{máximo} (S(i, j, q))$

Passo 5: Se $S(i^*, j^*, q^*) \leq 0$ (FIM)

Se $S(i^*, j^*, q^*) > 0$, inserir q^* entre i^* e j^*

Atualizar $R, f(R, p), \forall p \in K, g(R, q, p) \forall p \in K$
 $\forall q \in K-R$

retornar ao passo 4.

Na segunda generalização do (PCV) proposta neste trabalho, foi suposto que cada par ordenado (i, j) está associado com um lucro e um tempo de percurso. O objetivo do problema foi a de encontrar uma sub-rota que inicia e termina na origem, maximizando os lucros e terminando num tempo não superior a um limite pré-estabelecido.

OCHI (1989), descreveu o (FVC) como um modelo de Problema de Programação Linear Inteira propondo novas restrições que evitam a formação de sub-rotas inviáveis, na solução do (FVC).

PSARAFTIS (1980) examinou o problema *Dial-a-Ride* focalizando as versões estática e dinâmica. No caso

estático, a função objetivo minimiza uma combinação de pesos formada pelo tempo de servir todos os passageiros e o tempo total de espera e de viagem de cada passageiro. Nesta versão a restrição de capacidade do veículo é considerada prioritária. No caso dinâmico, são consideradas possibilidades de incorporar novos passageiros no schedule abrindo-se assim possibilidades de novas modificações. Em ambos os casos, a solução foi encontrada através de métodos de Programação Dinâmica.

SOUKIS, FERLAND e ROUSSEAU (1980) descreveram um novo modelo para o problema de roteamento e scheduling de vôos de aviões procurando maximizar o lucro da companhia e a satisfação dos passageiros.

O procedimento avança na parte tocante às interações entre passageiros e o roteamento de aviões propriamente dito.

E apesar do elevado nível de complexidade da formulação desenvolvida, os resultados computacionais foram bem satisfatórios, graças em parte à eficiência dos métodos utilizados na sua solução.

STERN e HERSH (1980) desenvolveram uma nova técnica para o problema de programação de equipes de limpeza nos interiores dos aviões. A performance deste trabalho, determinado também pelo tempo total gasto pela equipe, é

limitado a um intervalo de tempo em que o avião está pousado em terra e disponível para este serviço.

Existem restrições de horário de cada membro da equipe de limpeza referente aos intervalos de lanche e almoço.

O problema tem como objetivo, encontrar a dimensão ótima de cada equipe para cada turno de modo que o custo global deste pessoal seja minimizado.

Este problema foi formulado como um Modelo de Programação Inteira e Linear e a solução obtida usando técnicas de Decomposição Linear.

BAKER e FISHER (1981) desenvolveram um novo esquema heurístico, a partir de um modelo de recobrimento (MR) para resolver o problema de scheduling de tripulações aéreas (PSTA), da Federal Express Corporation.

Foram apresentados resultados computacionais de problemas com até 3.000 linhas e 15.000 colunas da matriz do (MR) resolvidos num IBM 3033 em menos de uma hora.

O (MR) foi formulado e interpretado da seguinte forma:

$$\text{Minimizar } \sum_{j=1}^n c_j x_j \quad (2.178)$$

$$\text{S.A. } \sum_{j=1}^n a_{ij} x_j \geq 1, \dots, m, i \in I \quad (2.179)$$

$$x_j \in \{0, 1\}, j = 1, \dots, n \quad (2.180)$$

onde c_j = custo do schedule j

$$a_{ij} = \begin{cases} 1, & \text{se o vôo } i \text{ está contido no schedule } j \\ 0, & \text{c/c.} \end{cases}$$

n = número total de schedules viáveis.

$$x_j = \begin{cases} 1, & \text{se o schedule } j \text{ for selecionado.} \\ 0, & \text{c/c.} \end{cases}$$

O modelo de recobrimento (MR) é classificado na literatura como um problema de difícil resolução (NP-hard) o que limita o uso de algoritmos ótimos para a sua solução nas aplicações de grande porte.

Assim, neste artigo foi proposto uma nova heurística para resolver o (MR) descrito resumidamente da seguinte forma:

HEURÍSTICA PARA (MR)

Passo 1: INICIALIZAÇÃO

O conjunto solução é vazio, nenhum voo está coberto.

Passo 2: CRITÉRIO PARA ANEXAR NOVOS VOOS NO SCHEDULE:

Identificar o schedule que efetua um voo não coberto a um menor custo, efetuar a alocação (voo, schedule) e atualizar: o conjunto de voos descobertos e o conjunto de schedules.

Passo 3: TESTAR A SOLUÇÃO ATUAL:

Se todos os voos estão cobertos (FIM);

caso contrário, retornar ao Passo 2).

BEASLEY (1981) adaptou o algoritmo de *savings* de CLARKE e WRITE (1964) para um problema de scheduling de veículos onde o tempo gasto para percorrer o trecho entre dois pontos distintos diretamente é uma função da hora do dia em que a viagem ocorrer.

Para isso, o dia foi dividido em k períodos consecutivos de tempo, onde $\{t_{ij}^k\}$ é a matriz de tempo de

percurso direto entre i e j no período k .

Para adaptar o algoritmo de *savings* a esta versão, foram estabelecidos:

i) uma nova regra para definir as economias s_{ij} em termos dos $\{t_{ij}^k\}$.

ii) uma regra para calcular o tempo de percurso quando a viagem entre dois nós engloba dois ou mais períodos consecutivos de tempo.

iii) um novo critério para testar a viabilidade no algoritmo que pode ter num schedule, tempos de percursos diferentes, se o sentido do schedule for alterado, por exemplo de $(o-i-j-o)$ para $(o-j-i-o)$.

BROWN e GRAVES (1981) descreveram um novo sistema de despacho de caminhões tanques transportando derivados de petróleo de uma refinaria para cerca de 80 terminais de venda.

A frota consiste de aproximadamente 300 caminhões que devem efetuar diariamente um total de 2.600 viagens.

Os autores analisaram um modelo de Programação Linear Inteira gerado para resolver o problema onde:

i : índice da ordem de atendimento

j : índice do caminho

$J(i)$: conjunto de caminhos viáveis com a ordem i

c_{ij} : custo de transporte do cliente da ordem i com o caminho j

t_{ij} : tempo de transporte do cliente da ordem i com o caminho j

\bar{s}_j e s_j : deslocamentos máximo e mínimo do caminho j

\bar{z}_j e z_j : taxa de penalidade por violar os limites de deslocamentos do caminho j .

$$y_{ij} = \begin{cases} 1, & \text{se o cliente da ordem } i \text{ é alocado para o} \\ & \text{caminho } j \\ 0, & \text{c/c.} \end{cases}$$

$$x_j = \begin{cases} \bar{z}_j \\ 0 \\ z_j \end{cases} = \text{variável de penalidade}$$

FORMULAÇÃO MATEMÁTICA PROPOSTA

$$\text{Min. } \sum_i \sum_{j \in J(i)} c_{ij} y_{ij} + \sum_j x_j [s_j - \sum_i t_{ij} y_{ij}] \quad (2.181)$$

$$\text{S.A. } \sum_{j \in J(i)} y_{ij} = 1 \quad \forall_i \quad (2.182)$$

$$s_j = \bar{s}_j \text{ e } x_j = \bar{z}_j \text{ quando } \sum_i t_{ij} y_{ij} > \bar{s}_j \quad (2.183)$$

$$s_j = \underline{s}_j \text{ e } x_j = \underline{z}_j \text{ quando } \sum_i t_{ij} y_{ij} < \underline{s}_j \quad (2.184)$$

$$s_j = \bar{s}_j \vee \underline{s}_j \text{ e } x_j = 0 \text{ quando} \\ \underline{s}_j \leq \sum_i t_{ij} y_{ij} \leq \bar{s}_j \quad (2.185)$$

$$y_{ij} \in (0, 1) \quad \forall_i, \forall_j \quad (2.186)$$

Note que $\underline{s}_j < \bar{s}_j \Rightarrow \bar{z}_j \leq 0 \leq \underline{z}_j$

As restrições (2.182) asseguram o atendimento das demandas numa única visita enquanto (2.183), (2.184) e (2.186) executam as composições desejadas no modelo.

LEVARY (1981) analisou um problema ligado a uma loja de prestação de serviços, que atende uma região previamente delimitada.

Neste problema, os clientes requisitam diferentes tipos de serviços com restrições no horário em que eles devem ser efetuados num determinado dia.

Para estes atendimentos, a loja dispõe de um número limitado de funcionários que devem receber diariamente uma programação dos atendimentos a efetuar.

Foi suposto que cada funcionário está apto a resolver todo tipo de serviço prestado pela loja.

O artigo propõe um método heurístico para este problema, tendo como objetivos básicos:

i) minimizar o tempo total de percursos

ii) gerar um conjunto de schedules diariamente

iii) os schedules devem ser gerados de modo que os tempos de percurso e de serviço estejam ao seu final mais próximos dos tempos disponíveis dos funcionários.

MARSTEN e SHEPARDSON (1981) destacaram a popularidade dos modelos de particionamento (PF) ou de recobrimento (MR) para representar os problemas de roteamento e scheduling de tripulações aéreas.

Neste artigo, os autores utilizaram um modelo de

particionamento resolvendo-o via técnicas de relaxação lagrangeana e decomposições:

O (PP) é da forma:

$$\text{Minimizar } \sum_{j=1}^n c_j x_j \quad (2.187)$$

$$\text{S.A. } \sum_{j=1}^n a_{ij} x_j = 1, \quad i = 1, \dots, m \quad (2.188)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (2.189)$$

$$\text{onde } a_{ij} = \begin{cases} 1, & \text{se o vôo } i \text{ está incluído no schedule } j \\ 0, & \text{c/c.} \end{cases}$$

c_j = custo global associado ao trip j

$$x_j = \begin{cases} 1, & \text{se o schedule } j \text{ for selecionado} \\ 0, & \text{c/c.} \end{cases}$$

O (PP), neste caso, tem como objetivo gerar um conjunto de schedules (um para cada tripulação) que atenda o conjunto de vôos programados a um menor custo global possível.

Os autores destacaram a dificuldade de se trabalhar com um modelo (MP) ou (MR) nas aplicações práticas devido ao elevado número de colunas do problema (30.000 no caso da Air

Canada e Air France).

SCHRAGE (1981) apresentou um novo *survey* sobre as formulações e estruturas dos problemas de roteamento e scheduling de veículos (PRSV).

Neste trabalho foram descritas e analisadas as restrições que aparecem mais frequentemente nos (PRSV) tais como: tempo máximo de viagem, capacidade dos veículos, tipos de veículos, divisão e junção de cargas, intervalos de tempos, etc.

BALL, BODIN e DIAL (1983) descreveram um novo procedimento computacional para o (PSV) e de tripulações num sistema de trânsito urbano.

A metodologia utilizada difere da maioria dos trabalhos encontrados na literatura, já que, aqui, os veículos e as tripulações são programadas simultaneamente no tempo ao invés de programar inicialmente os schedules dos veículos para depois alocar as tripulações.

BALL, GOLDEN, ASSAD e BODIN (1983) resolveram um problema de distribuição de uma grande empresa de produtos químicos envolvendo o traslado de demandas entre vários pontos origens e destinos localizados nos EUA e Canadá. O objetivo do trabalho foi de determinar o tamanho ótimo da frota de veículos da empresa e um conjunto de schedules

quase-ótimos. A solução foi obtida através do uso de três algoritmos heurísticos dos tipos "route first-cluster second" e "greedy insertion".

SCOTT (1984) apresentou na sua tese de doutorado um trabalho sobre a programação de veículos num sistema de Trânsito Urbano.

No trabalho são propostas novas técnicas de programação de veículos formuladas como Problemas de Programação Linear.

Na aviação comercial, ocorre com alguma frequência o caso onde um ou mais aviões de uma companhia aérea serem retirados de operação por razões técnicas obrigando a companhia a operar com um reduzido número de aviões, insuficiente para cobrir toda a demanda prevista de passageiros, pensando nisto:

TEODOROVIC e GUBERINIC (1984) desenvolveram uma nova metodologia de scheduling das linhas aéreas que leve em conta esta situação.

EVANS e NORBACK (1985) descreveram um sistema automatizado para o problema de distribuição de refeições através de uma frota de veículos.

A solução foi obtida utilizando procedimentos anteriormente propostos por *EVANS (1982)* e uma heurística do

(PCV) proposto por *NORBACK e LOVE (1979)* e nos testes realizados o sistema produziu economias de até 10% em relação a solução adotada anteriormente.

LAVOIE, MINOUX e ODIER (1985) propuseram um novo procedimento para o planejamento das tripulações aéreas.

Inicialmente o problema é descrito como um modelo de recobrimento (MR) com um elevado número de colunas, resolvido através do método simplex com um novo procedimento de geração de colunas.

O sub-problema de geração de colunas é um problema de caminho mínimo com restrições de scheduling.

O algoritmo foi testado para um exemplo real com cerca de 132 vôos obtendo bons resultados computacionais, onde a solução obtida se mostrou de 4 a 5% mais econômica em relação ao programa anteriormente utilizado pela empresa.

SKITT e LEVARY (1985) desenvolveram um método aproximado de geração de colunas para um problema de distribuição de produtos em quatro níveis. O problema foi formulado como um modelo de Programação Linear onde cada coluna representa uma rota viável. Devido ao elevado número de rotas viáveis, foi utilizado um método de geração de colunas para atualizar as bases do Simplex.

Nos problemas de planejamento de vôos aéreos, existem situações onde os programas de vôos são truncados devido à ocorrência de situações imprevistas como condições meteorológicas desfavoráveis.

Uma consequência imediata é o aumento do número de cancelamentos de vôos previamente programados, quebrando-se assim a estrutura dos schedules previamente gerados.

Pensando nisso, *TEODOROVIC (1985)* propôs um método de programação de vôos que levasse em conta fatores climáticos de cada região e suas eventuais consequências.

O método deveria incluir portanto procedimentos de reotimização diante de alterações no número de aviões ou aeroportos disponíveis de modo que um menor número de passageiros fiquem sem atendimento.

CEGIELLA, SZYMANOWSKI e PROKURATORSKI (1986) desenvolveram um sistema de coleta de leite (COLDS) cujo objetivo é o de gerar um conjunto de schedules para uma frota de veículos que deve recolher o leite de vários pontos e transportá-lo até uma das fábricas de laticínios.

CRAINIC e ROUSSEAU (1986) apresentaram um novo procedimento para resolver o problema de programação de tripulações aéreas baseado na técnica de geração de colunas.

O problema é formulado como um modelo de recobrimento (MR) que permite que mais de uma tripulação embarque num mesmo vôo. Neste caso uma tripulação é aquela escalada de fato para este vôo e os outros viajam para as suas novas origens (viagens ociosas).

RONEN (1986) analisou o problema de roteamento e scheduling de navios heterogêneos que são alocados para o transporte de um conjunto de cargas para os seus respectivos destinos.

Neste trabalho foram propostos três algoritmos, uma heurística, um gerador de schedules viáveis e um algoritmo de otimização baseado na técnica de enumeração implícita aplicado a uma formulação inteira mista e não linear descrito da seguinte forma:

Considere:

J = conjunto dos portos

K = conjunto de cargas

N_k = conjunto de schedules viáveis para o navio k

a_{jk} = custo unitário de embarque para o porto j pelo navio k

c_k = capacidade do navio k

f_{km} = custo do navio k efetuar o schedule m

F_k = custo diário do navio k parado em um porto

Q_j = tamanho da carga do porto j

U_j = vazão de desembarque do porto j

T_{kn} = variável de decisão = $\begin{cases} 1, & \text{se o navio k efetua} \\ & \text{schedule n} \\ 0, & \text{c/c.} \end{cases}$

x_{jkn} = variável = $\begin{cases} 1, & \text{se o navio k entra no porto j, no} \\ & \text{schedule n} \\ 0, & \text{c/c.} \end{cases}$

$0 \leq y_{jk} \leq 1$.. parcela da carga para o porto j que é carregado no navio k

FORMULAÇÃO MATEMÁTICA

$$\text{Min. } \sum_{k \in K} \sum_{j \in J} \sum_{n \in N} a_{jk} Q_j y_{jk} X_{jkn} + \sum_{k \in K} \sum_{j \in J} \sum_{n \in N_k} (Q_j P_k / U_j) *$$

$$* y_{jk} X_{jkn} + \sum_{k \in K} \sum_{n \in N_k} f_{kn} t_{kn} \quad (2.190)$$

$$\text{S.A. } \sum_{j \in J} \sum_{n \in N_k} Q_j X_{jkn} Y_{jk} \leq C_k, \quad k \in K \quad (2.191)$$

$$\sum_{n \in N_k} t_{kn} = 1, \quad k \in K \quad (2.192)$$

$$\sum_{k \in K} \sum_{n \in N_k} x_{jkn} y_{jk} = 1, \quad j \in J \quad (2.193)$$

$$\sum_{k \in K} \sum_{n \in N_k} C_k x_{jkn} \geq Q_j, \quad j \in J \quad (2.194)$$

$$\left. \begin{array}{l} x_{jkn} \in (0,1), \quad t_{kn} \in (0,1) \\ \emptyset \leq y_{jk} \leq 1 \end{array} \right\} \begin{array}{l} \forall j \in J \\ \forall k \in K \\ \forall n \in N \end{array} \quad (2.195)$$

As restrições não lineares (2.191) se referem à satisfação da capacidade de cada navio k , enquanto (2.192) afirma que cada schedule deve ser alocado a exatamente um navio.

A condição (2.193) (não-linear) nos diz que as demandas de cada posto devem ser completamente satisfeitos.

Finalmente (2.194) (redundante) diz que a soma das capacidades alocadas a um ponto j deve ser maior ou igual à demanda de j .

Os três algoritmos desenvolvidos neste trabalho foram testados para um conjunto de problemas com até 13 cargas, 13 portos e visitando no máximo 3 portos por viagem. O método de enumeração implícita se mostrou eficiente apenas para problemas com dimensões reduzidas (máximo de 5 cargas e 7 portos).

SCOTT (1986) estudou o problema de alocar uma frota de veículos para atender um conjunto fixado de horários de trips.

Este trabalho discute a controvérsia existente na literatura sobre a viabilidade de se otimizar simultaneamente a dimensão da frota e os custos operacionais.

Discordando da opinião de alguns autores, este artigo defende a idéia de otimização simultânea mostrando que normalmente não existe nenhuma incompatibilidade entre os dois objetivos fornecendo inclusive condições suficientes que mostram que a solução que minimiza os custos

operacionais também fornece a frota mínima de veículos.

BROWN, GRAVES e RONEN (1987) apresentaram um problema de planejamento de uma frota de navios petroleiros em longo curso.

O problema é formulado como um modelo de particionamento com restrições adicionais (MPA) e a solução fornece as velocidades ótimas, os melhores schedules de cada navio e minimiza o custo total das viagens em lastro.

CHRISTOFIDES, VALDES e TAMARIT (1987) desenvolveram um novo algoritmo da classe Branch-and-Bound para o problema de planejamento de novos investimentos com recursos limitados.

O algoritmo utiliza arcos disjuntivos para resolver conflitos gerados quando o conjunto de atividades a serem programados exige recursos maiores que os disponíveis no período.

São analisados quatro limites inferiores, sendo um deles um algoritmo do caminho mais longo, o segundo usando relaxação linear e restrições adicionais de cortes, o terceiro baseado em relaxações lagrangeanas do modelo anterior e o quarto baseado numa generalização do caminho mais longo com uso de arcos disjuntivos.

SCULLI, MOK e CHEUNG (1987) descreveram um problema de

scheduling de veículos para a coleta do lixo urbano.

Vários algoritmos heurísticos foram testados para um exemplo real, resultando em soluções com até 8 % de economia em relação ao método manual até então utilizado.

SINCLAIR e VANM DYK (1987) descreveram um problema do tipo "routing and scheduling" de veículos para o transporte de cargas em containers.

O problema também conhecido como "tractor-trailer" possui normalmente um conjunto de características que listamos a seguir:

- são transportadas somente cargas completas
- time-windows de tempos de chegada
- Existência de uma ordem de visitas (restrições de precedências) entre alguns clientes
- conjunto não homogêneo de operações

Devido à complexidade do problema, este artigo sugere um procedimento heurístico do tipo "route first - cluster second" para a sua solução.

CARPANETO, DELLA'MICO, FISCHETTI e TOTH (1988) desenvolveram novos algoritmos para gerar limites inferiores para o (PSV) com múltiplos depósitos.

O trabalho descreve um novo critério de dominância baseado no trabalho de FISHETTI e TOTH (1987) e apresenta um novo método do tipo Branch-and-Bound para o (PSV).

FISHER (1988) descreveu um novo survey sobre as implementações do problema de roteamento e scheduling de veículos (PRSV). O trabalho ressalta o uso de técnicas de relaxações lagrangeanas que tem viabilizado o uso dos métodos exatos de otimização na solução de um (PRSV).

Foram apresentados quatro modelos de (PRSV) e propostas de novas relaxações.

NYGARD, GREENBERG, BOLKAN e SWENSON (1988) apresentaram um novo algoritmo heurístico para o problema de roteamento e scheduling de veículos com o objetivo de minimizar o comprimento total dos schedules sujeito às restrições de capacidade dos veículos e restrições de tempo máximo de chegada a um nó.

O método proposto descreve o problema como um modelo de alocação generalizada (PAG) e usa procedimentos de otimização local e técnicas iterativas de ajuste de custos na sua solução.

A heurística desenvolvida foi aplicada a um problema real de entrega de jornais diários da gráfica para cerca de 300 entrepostos, com a exigência de que estes cheguem aos

entrepósitos no máximo até 6:00 horas da manhã, para viabilizar os despachos subsequentes.

PSARAFITIS (1988) apresentou um novo resumo sobre o estágio atual de desenvolvimento do Problema de roteamento de veículos Dinâmico (PRVD), também conhecido na literatura como o problema de roteamento em tempo-real.

O trabalho inicia descrevendo o relacionamento existente entre os problemas de roteamento estático (PRVE) e o problema de roteamento dinâmico (PRVD), identificando fatores que às vezes tornam estes dois problemas muito diferentes um do outro.

Algumas características próprias do (PRVD) ~~são listadas~~ a seguir:

i) *As dimensões do tempo são essenciais:* note que isto nem sempre ocorre num (PRVE). Quando este item for importante no (PRVE), o problema é denominado Problema de roteamento e scheduling de veículos, em outros casos o (PRVE) simplesmente não inclui as restrições de tempo, por exemplo, no Problema do Caixeiro Viajante (PCV), o tempo de percurso normalmente é suposto ser proporcional à distância percorrida.

ii) *Os schedules podem ter duração não-limitadas:* Ao contrário do (PRVE) onde a duração de um schedule é

normalmente limitada ou pela capacidade do veículo ou pelo horizonte de planejamento limitado, no caso dinâmico, podemos ter schedules cuja duração é ilimitada.

iii) As informações futuras podem ser imprecisas ou desconhecidas.

iv) Os eventos mais próximos são mais importantes: Ao contrário do (PRVE) onde se dá os mesmos "pesos" aos diferentes eventos próximos ou afastados.

v) Mecanismos que atualizam informações são essenciais: Isto porque no (PRVD) todas as entradas estão sujeitas a revisões a qualquer estágio do seu desenvolvimento.

vi) Decisões de resequenciamento e realocações podem ser autorizadas: Pode ocorrer, por exemplo, no caso onde indexamos um novo dado de entrada ao problema.

vii) Os tempos Computacionais devem ser "rápidos". No (PRVD) existe a necessidade de várias reotimizações de schedules o que obriga uma rapidez maior nos tempos computacionais para viabilizar o procedimento.

viii) Mecanismos de adiamento indefinido são essenciais: O atendimento de uma demanda pode ser postergado indefinidamente devido às características geográficas desfavoráveis deste em relação às outras demandas.

ix) A função objetivo pode ser "diferente". Ao contrário do (PRVE) onde existe a necessidade de minimizar custos ou distâncias totais, no (PRVD) isso pode perder o sentido, por exemplo se os schedules, permanecem "em aberto", neste caso, os custos também ficarão "em aberto".

Ainda neste trabalho, foi apresentado um novo algoritmo dinâmico aplicado ao problema do despacho de cargas por uma frota de navios.

Discute-se também as perspectivas futuras desta área.

II.3) O PROBLEMA DE ROTEAMENTO E SCHEDULING DE VEÍCULOS COM RESTRIÇÕES TIME-WINDOWS (PRSTW).

Nos últimos anos, os pesquisadores da área de roteamento e scheduling de veículos têm se preocupado em desenvolver modelos matemáticos e algoritmos que se aproximem mais dos problemas da vida real.

Para isso, foram desenvolvidos algoritmos específicos para tratarmos de problemas de grande porte englobando restrições que representem mais fielmente um modelo real.

Uma das restrições que normalmente encontramos nos problemas práticos são intervalos de tempo *time-windows* dentro do qual uma determinada tarefa deve ser iniciada ou efetuada.

Um problema de roteamento e scheduling com restrições do tipo Time-Windows é constituído basicamente de uma frota de veículos localizados em um depósito e alocados para atender um conjunto de clientes distribuídos numa região geográfica limitada com demandas conhecidas e com Time-Windows, dentro do qual o cliente deve ser atendido.

O objetivo, normalmente é o de gerar um conjunto de schedules viáveis, um para cada veículo, de modo a minimizar

os custos fixos e variáveis do problema.

A partir deste modelo básico, desenvolveram-se centenas de variações, desde modelos mais simples (mas nem por isso mais fáceis de resolver) como o problema do caminho mínimo com *time-windows* ou o problema do caixeiro viajante (PCV) com *time-windows* até modelos mais complexos como a aplicação (FP) (veja sua descrição no capítulo 2 deste trabalho).

Os problemas de roteamento e scheduling de veículos com restrições do tipo *time-windows* (PRSTW) definem uma nova classe de problemas de roteamento de veículos devido às suas características próprias que a tornam muito mais complexas.

Por exemplo, na ausência de *time-windows*, podemos especificar a priori o conjunto de tarefas que sucede uma tarefa *i*, o que é impossibilitada na sua presença devido ao desconhecimento prévio do ponto de *time-window* a ser utilizado pela tarefa *i*, em outras palavras, as tarefas que são candidatas a suceder a tarefa *i* variam de acordo com o tempo do intervalo *time-window* que a tarefa *i* é iniciada (ou concluída).

Na maioria dos trabalhos desenvolvidos sobre o (PRSTW) as restrições *time-windows* são apenas satisfeitas sem que exista uma preocupação maior de otimizá-la.

Nestes casos, as variáveis de tempo não aparecem

explicitamente na função objetivo normalmente associados somente a custos (lucros).

Isto pode ser facilmente observado pelos objetivos que normalmente desejam-se otimizar, geralmente envolvendo custos globais relacionados somente com variáveis espaciais de fluxo e/ou de decisão espacial.

Na literatura consultada, encontramos poucos exemplos onde os tempos recebem explicitamente importância igual ou maior às decisões espaciais. Um caso onde isto pode ou deve ocorrer são os problemas de roteamento e scheduling de veículos com várias velocidades, onde fica evidente a importância no critério de escolha de tempos para iniciar ou efetuar uma tarefa.

Esta importância passa a ser clara nestes tipos de problemas na medida que um tempo reduzido implica em velocidades maiores e conseqüentemente em custos maiores de combustíveis enquanto tempos maiores implicam normalmente em custos menores de combustíveis.

Estes dois objetivos são conflitantes e acabam exigindo a presença dos dois itens (tempo e custos espaciais) na função objetivo do problema, ou como função mono-objetivo ou, se for conveniente, na forma de múltiplos objetivos.

Esta preocupação será mencionada nos capítulos

posteriores deste trabalho, como por exemplo, na definição do critério de economias gerado no capítulo 4 deste trabalho.

Apresentamos a seguir, uma breve sinopse (numa ordem cronológica de publicações) de parte dos trabalhos encontrados na literatura sobre os (PRSTW).

Os problemas de roteamento e scheduling de veículos com restrições time-windows ganharam impulso, graças em parte ao surgimento de novas técnicas de otimização como, por exemplo, as relaxações lagrangeanas que viabilizaram o uso de técnicas exatas na solução de problemas lineares contínuos e combinatoriais de grande porte.

Mas há quase vinte anos atrás, *MARTIN-LOF (1970)* já descrevia um problema de determinar schedules com restrições de janelas de tempo para um sistema de trânsito de ônibus ou aviões, de modo a minimizar o número de veículos necessários para efetuar um conjunto de trips.

Os vários trips existentes foram caracterizados pela presença de janelas de tempo com origem, destino, tempo e custo de percurso conhecidos.

O problema foi formulado como um modelo de Programação linear misto e resolvido por uma técnica do tipo Branch-and-Bound.

Mais recentemente encontramos uma nova generalização para o Problema do Caixeiro Viajante (PCV), incluindo restrições adicionais de scheduling.

O problema definido como o Problema do Caixeiro Viajante com restrições Time-Windows (PCVTW) foi desenvolvido por *BAKER (1983)* que propôs uma nova formulação matemática para a versão simétrica com a propriedade da desigualdade triangular.

Obs.: Os tempos considerados são quantidades medidas a partir do momento inicial (tempo zero)

t_i = o tempo que o caixeiro visita a cidade i

t_{n+1} = tempo para completar um ciclo hamiltoniano

$|X|$ = valor absoluto de X

t_{ij} = tempo mínimo para percorrer (i, j) onde

$T = (t_{ij})$ é a matriz completa, não-negativa e simétrica

$[a_i, b_i]$ = time-window da cidade i , onde $a_i \geq 0$

$$\text{Minimizar } t_{n+1} - t_1 \quad (2.196)$$

$$\text{S.A. } t_i - t_1 \geq t_{1i}, \quad i = 2, 3, \dots, n \quad (2.197)$$

$$|t_i - t_j| \geq t_{ij}, \quad i = 3, 4, \dots, n, \quad 2 \leq j < i \quad (2.198)$$

$$t_{n+1} - t_i \geq t_{i, n+1}, \quad i = 2, 3, \dots, n \quad (2.199)$$

$$t_i \geq 0, \quad i = 1, 2, \dots, n+1 \quad (2.200)$$

$$a_i \leq t_i \leq b_i, \quad i = 2, 3, \dots, n \quad (2.201)$$

Foi demonstrado neste trabalho que o (FRSTW) descrito por (2.196) - (2.201), gera uma solução ótima para o (PCV), ou seja, demonstrou-se que o (FRSTW) é uma extensão do (PCV).

As restrições (2.198) permitem para cada par (i, j) , duas alternativas, e a escolha de um dos dois casos para cada par (i, j) gera um novo modelo linear.

A proposta de *BAKER (1983)* foi a de relaxar as restrições não-lineares (2.198) e as time-windows (2.201) do (FRSTW), gerando o seguinte problema linear relaxado (PR1) :

$$\text{Minimizar } t_{n+1} - t_1$$

$$\text{S.A. } t_i - t_1 \geq t_{1i}, \quad i = 2, 3, \dots, n$$

$$t_{n+1} - t_i \geq t_{i-1}, \quad i = 2, 3, \dots, n$$

$$t_i \geq 0, \quad i = 1, 2, \dots, n+1$$

O valor ótimo do problema relaxado PR1 foi obtido resolvendo o dual de PR1 que é um problema do caminho mais longo num grafo direcionado com $n+1$ nós, denotado (DR1) e descrito por:

$$\text{Maximizar } \sum_{j=2}^n (t_{j-1} x_{j-1} + t_{j-1} x_{j+n-2})$$

$$\text{S.A. } \sum_{j=1}^{n-1} (-X_j) \leq -1, \quad j = 1, 2, \dots, n-1$$

$$X_j - X_{j+n-1} \leq 0$$

$$\sum_{j=1}^{n-1} X_{j+n-1} \leq 1$$

$$X_j, X_{j+n-1} \geq 0, \quad j = 1, 2, \dots, n-1$$

O valor ótimo $V(\text{DR1})$ do (DR1) ou $V(\text{PR1})$ de (PR1) ($V(\text{PR1}) = V(\text{DR1})$) foi usado como um limite inferior para um procedimento de Branch-and-Bound aplicado ao (PRSTW) com valores discretos ($t_i \geq 0$ e inteiros).

DESROSIERS, DUMAS e SOUMIS (1984) descreveram um problema *Dial-a-ride* composto de um único veículo,

restrições time-windows, paradas de embarque e desembarque, relações de precedências, e restrições de capacidade do veículo.

O problema foi formulado como um modelo de Programação misto, descrito a seguir:

Considere:

t_i = variável de tempo de chegada ao nó i

$$X_{i,j} = \text{var. de fluxo} = \begin{cases} 1, & \text{se a rota usa o arco } (i,j) \\ 0, & \text{caso contrário.} \end{cases}$$

Y_i = variável que representa o número de passageiros no veículo na saída do nó i .

$[A_i, B_i]$ = time-window do nó i .

$D_{i,j}$ = distância do arco (i, j)

$T_{i,j}$ = tempo de percurso do arco (i, j)

M_i = tempo requerido para embarcar os passageiros do nó i .

C_i = número de passageiros a serem transportados no nó i .

$$\text{Minimizar } z = \sum_{i=1}^{2n} \sum_{j=1}^{2n+1} D_{ij} X_{ij} \quad (2.202)$$

$$\text{S.A. } \sum_{j=1}^n X_{0j} = 1 \quad (2.203)$$

$$\sum_{i=n+1}^{2n} X_{i,2n+1} = 1 \quad (2.204)$$

$$\sum_{i=0}^{2n} X_{ik} = \sum_{j=1}^{2n+1} X_{kj} = 1, \quad k = 1, 2, \dots, 2n \quad (2.205)$$

$$X_{ij} \in \{0, 1\}, \quad i = 0, 1, \dots, 2n, \quad j = 1, 2, \dots, 2n+1 \quad (2.206)$$

$$A_i \leq t_i \leq b_i, \quad i = 0, 1, \dots, 2n+1 \quad (2.207)$$

$$t_i < t_{n+i}, \quad i = 1, 2, \dots, n \quad (2.208)$$

$$\text{Se } X_{ij} = 1 \rightarrow t_i + M_i + T_{ij} \leq t_j, \quad i=0, 1, \dots, 2n \quad (2.209) \\ j = 1, 2, \dots, 2n+1$$

$$Y_0 = C_0, \quad 0 \leq y_i \leq C \quad i = 1, 2, \dots, 2n+1 \quad (2.210)$$

$$\text{Se } X_{ij} = 1 \rightarrow y_i + C_i = y_j, \quad i = 0, 1, \dots, 2n \quad (2.211) \\ j = 1, 2, \dots, 2n+1$$

As restrições (2.203) - (2.206) descrevem um problema de fluxo (roteamento puro), onde cada nó é visitado uma única vez (2.205), com um único veículo saindo da origem 0 (2.203), e terminando no nó $2n+1$ (2.204).

As condições (2.207) - (2.209) são as restrições de scheduling do Problema, onde (2.207) são os time-windows de cada nó, (2.208) as relações de precedências, (2.209) as restrições de acoplamento entre os fluxos e as variáveis de tempo, (2.210) é a restrição de capacidade do veículo e finalmente (2.211) o acoplamento entre as variáveis de fluxo com as variáveis y_i .

O problema (2.202) - (2.211) foi resolvido através de um procedimento de Programação Dinâmica Forward e testado para os problemas com até $n = 40$ paradas em no máximo 5,27 segundos num CYBER 173.

Um dos trabalhos mais citados atualmente na literatura do problema de roteamento de veículos com time-window foi desenvolvido por *DESROSIERS, SOUMIS e DESROCHERS (1983)*.

O artigo trata da solução de um problema de roteamento com time-window (FRTW) descrito como um modelo de particionamento (*set partitioning model*) ou (MP), que por sua vez foi resolvido via Simplex e Branch-and-Bound.

Devido ao elevado número de colunas viáveis do Simplex (schedules viáveis), foi utilizado um procedimento de geração de colunas, descrito como um problema de caminho mínimo com time-windows (PCMTW).

O (PRTW) foi formulado como um Problema de Programação Linear Inteira misto e descrito da seguinte forma:

$$\text{Minimizar } \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.212)$$

$$\text{S.A. } \sum_{j \in N} x_{ij} = \sum_{j \in N} x_{ji} = 1, i \in P \quad (2.213)$$

$$x_{ij} \geq 0, (i, j) \in A \quad (2.214)$$

$$\text{Se } x_{ij} > 0 \rightarrow t_i + t_{ij} \leq t_j \quad (i, j) \in I \quad (2.215)$$

$$a_i \leq t_i \leq b_i, i \in P \quad (2.216)$$

$$x_{ij} \in \{0, 1\}, (i, j) \in A \quad (2.217)$$

onde P = conjunto dos trips

I = conjunto dos arcos intertrips (entre dois trips)

N = conjunto de nós da rede original

A = conjunto dos arcos da rede original

As restrições (2.212), (2.213) e (2.214) descrevem um problema de roteamento puro sem restrições de scheduling, conhecido na literatura como o problema de fluxo de custo mínimo que possui a propriedade de integralidade.

As restrições de acoplamento (2.215) afirmam que se houver fluxo positivo entre (i, j) , então as restrições de tempo relacionados com (i, j) devem ser atendidas.

Finalmente, (2.216) são as restrições de Time-Windows do problema.

O (PRTW) descrito pelas restrições (2.212) - (2.217) não é linear devido às restrições (2.215) que podem no entanto ser linearizadas na forma:

$$t_i + t_{ij} - t_j \leq (1 - X_{ij}) M_{ij}, \quad (i, j) \in I \quad (2.218)$$

$$\text{onde } M_{ij} \geq b_i + t_{ij} - a_j$$

As restrições (2.218) são equivalentes a (2.215) somente quando $X_{ij} = 0$ ou $X_{ij} = 1$.

Neste trabalho, o (PRTW) foi transformado num modelo de particionamento (MP), gerando a priori, o conjunto dos schedules viáveis do (PRTW).

$$\text{Minimizar } \sum_{j \in S} c_j y_j \quad (2.219)$$

$$\text{S.A. } \sum_{j \in S} a_{ij} y_j = 1, \quad i \in P \quad (2.220)$$

$$y_j \in \{0, 1\}, \quad j \in S \quad (2.221)$$

onde c_j = custo global do schedule j

$$a_{ij} = \begin{cases} 1, & \text{se o trip } i \text{ está contido no schedule } j \\ 0, & \text{c/c.} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{se o schedule } j \text{ for selecionado} \\ 0, & \text{c/c.} \end{cases}$$

Devido ao elevado número de schedules viáveis existentes foi utilizado um sub-problema gerador de colunas para a atualização das bases do Simplex aplicado ao (MF).

O sub-problema foi descrito como um Problema do caminho mínimo com restrições time-windows (PCMTW).

$$V(\text{PCMTW}) = \text{Mínimo} \sum_{(i,j) \in A} (c_{ij} - \sigma_j) x_{ij} \quad (2.222)$$

$$\text{S.A.} \quad \sum_{j \in N} x_{ij} = \sum_{j \in N} x_{ji}, \quad i \in P \quad (2.223)$$

$$\sum_{j \in P} x_{sj} = \sum_{j \in P} x_{jt} = 1 \quad (2.224)$$

$$\text{Se } x_{ij} > 0 \Rightarrow t_i + t_{ij} \leq t_j; \quad (i, j) \in I \quad (2.225)$$

$$a_i \leq t_i \leq b_i, \quad i \in P \quad (2.226)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A \quad (2.227)$$

onde $c_{ij} - \sigma_j$ = custo relativo da coluna não básica j .

A solução do (PCMTW) foi obtida através de uma generalização do Algoritmo de *FORD* e *BELLMAN* para o problema de caminho mínimo.

O problema mestre (MP) foi resolvido via algoritmo Simplex e Branch-and-Bound para obter soluções inteiras.

SWERSEY e *BALLARD* (1984) analisaram um problema de programação de ônibus escolar para o transporte de estudantes das suas residências para as escolas.

O modelo aloca um ônibus para cada schedule gerado, conhecendo os pontos de parada para embarque e as escolas onde os estudantes devem ser desembarcados num intervalo de tempo pré-estabelecido, com o objetivo de minimizar o número de schedules (ônibus) necessários para o atendimento de todos os estudantes envolvidos.

Definiu-se a variável:

$$x_{ij} = \begin{cases} 1, & \text{se o trip } i \text{ antecede o trip } j \text{ no schedule} \\ 0, & \text{c/c.} \end{cases}$$

O problema de conectar trips para minimizar o número de schedules foi inicialmente descrito da forma:

$$\text{Minimizar } \sum_{j=1}^r x_{0j} \quad (2.228)$$

$$\text{S.A. } \sum_{i=1}^r x_{ij} = 1, \quad j = 1, \dots, r \quad (2.229)$$

$$\sum_{j=1}^r x_{ij} \leq 1, \quad i = 1, \dots, r \quad (2.230)$$

$$E_{s(j)} \leq t_j \leq L_{s(j)}, \quad j = 1, \dots, r \quad (2.231)$$

$$x_{ij} t_j \geq x_{ij} t_i + x_{ij} T_{ij}, \quad i = 1, \dots, r \quad (2.232)$$

$$j = 1, \dots, r$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, r \quad (2.233)$$

$$j = 1, \dots, r$$

O modelo P1 não é linear devido à presença das restrições (2.232) que impõe condições para que o início do trip j seja conectado com o final do trip i .

As restrições (2.229) e (2.230) garantem a continuidade do schedule, enquanto (2.231) são as janelas de tempo de cada trip i .

Posteriormente os autores sugeriram modificações em P1 transformando-o num modelo linear e totalmente inteiro, este

Último através da discretização dos intervalos contínuos das janelas de tempo. O problema linear inteiro foi resolvido usando o código LINDO num DEC 20.

DESROCHERS e SOUMIS (1985b) descreveram um problema de roteamento de veículos com restrições time-windows, como um problema de caminho mínimo com time-windows (PCMTW). O algoritmo desenvolvido neste artigo é pseudo-polinomial no tempo e foram testados para exemplos com até 2.500 nós.

O (PCMTW) consiste em encontrar a rota de custo mínimo cobrindo todos os nós da rede onde cada nó i deve ser visitado no intervalo de tempo $[a_i, b_i]$, este objetivo é alcançado resolvendo o seguinte problema de programação linear misto.

$$\text{Minimizar } \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.234)$$

$$\text{S.A. } \sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = \begin{cases} 1, & \text{se } i = p \\ 0, & \text{se } i \in N, i \neq p, i \neq q \\ -1, & \text{se } i = q \end{cases} \quad (2.235)$$

$$x_{ij} \geq 0, (i, j) \in A \quad (2.236)$$

$$\text{Se } x_{ij} > 0 \rightarrow t_i + t_{ij} \leq t_j, (i, j) \in A \quad (2.237)$$

$$a_i \leq t_i \leq b_i, i \in N \quad (2.238)$$

onde N = conjunto de nós

A = conjunto de arcos

p = nó fonte

q = nó destino

c_{ij} = custo de transporte entre (i, j)

t_{ij} = tempo para percorrer o arco (i, j)

t_i = variável de tempo início da tarefa i

$[a_i, b_i]$ = time-window do nó i

x_{ij} = fluxo do arco (i, j)

A solução do (PCM) com restrições adicionais, normalmente tem sido resolvido através de Programação Dinâmica ou através de métodos de relaxação lagrangeana.

Neste trabalho, os autores propuseram uma generalização do Algoritmo de *FORD e BELLMAN* e uma nova generalização do algoritmo de *DIJKSTRA*, ambos com restrições adicionais de time-windows.

Um outro trabalho semelhante sobre o (PCMTW) foi

desenvolvido por *DESROCHERS, SAUVE e SOUMIS (1985)*.

DESROSIERS, SOUMIS, DESROCHERS e SAUVE (1985)

trabalharam com a formulação do (PRSTW) descrito em (2.212) - (2.217) desenvolvendo dois novos algoritmos do tipo Branch-and-Bound sendo um deles, uma adaptação do método de *CARPANETO e TOOTH (1980)* onde relaxaram inicialmente as restrições de scheduling para obter um limite inferior para o método de Branch-and-Bound. No segundo, relaxaram as restrições de scheduling, usando Branch-and-Bound na discretização dos time-windows.

FERLAND (1985) escreveu um resumo sobre os problemas de scheduling de veículos com time-windows.

Para cada tarefa i , foi relacionado um conjunto de restrições do tipo:

-Um local de partida LP_i

-Um local de destino LD_i

-Um horário de partida HP_i

-Um tempo de duração TD_i

-Um tempo final $TF_i = HP_i + TD_i$

-Um time-window TW_i

O artigo descreveu vários procedimentos heurísticos e exatos para este tipo de problemas, sem no entanto efetuar comparações entre eles.

DESROSIERS, SAUVE e SOUMIS (1986) descreveram o problema de minimizar a frota heterogênea de veículos necessários para visitar um conjunto de nós, sujeito às restrições de espaço e de time-windows.

Foi suposto a existência de uma única origem de onde os veículos devem sair e retornar ao final da jornada.

Neste artigo foram propostos três novos algoritmos sendo um método de relaxação aumentada e duas relaxações lagrangeanas.

O problema foi formulado como:

$$\text{Minimizar } \sum_{j=1}^n Vx_{0j} + \sum_{i=0}^n \sum_{j=0}^n c_{ij}x_{ij} \quad (2.239)$$

$$\text{S.A. } \sum_{j=0}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (2.240)$$

$$\sum_{i=0}^n x_{ik} = \sum_{j=0}^n x_{kj}, \quad k = 1, 2, \dots, n \quad (2.241)$$

$$\sum_{j=0}^n x_{0j} = n \quad (2.242)$$

$$\sum_{i=0}^n x_{i0} = n \quad (2.243)$$

$$A_i \leq t_i \leq B_i, \quad i = 1, 2, \dots, n \quad (2.244)$$

$$t_i + t_{ij} - t_j \leq M_{ij} (1 - x_{ij}), \quad i, j \in \{1, 2, \dots, n / M_{ij} > 0\} \quad (2.245)$$

$$x_{ij} \geq 0 \text{ e inteiros } i, j = 0, 1, \dots, n \quad (2.246)$$

onde $M_{ij} = B_i + t_{ij} - A_j$

$$e x_{\infty} = \sum_{i=1}^n \sum_{j=1}^n x_{ij}$$

As restrições (2.240) - (2.243) e (2.246) formam um problema espacial, sem restrições de scheduling, sendo que (2.240) afirma que cada nó é visitado uma única vez, (2.241) são as restrições de conservação de fluxos que podem ser substituídas pelas restrições equivalentes $\sum_{i=0}^n x_{ij} = 1, j = 1, \dots, n$.

As restrições (2.242) e (2.243) são redundantes já que podem ser obtidas de (2.240) e (2.241) usando a definição de x_{∞} .

As restrições (2.244) são os intervalos Time-Windows dos nós e (2.245) descrevem a compatibilidade entre as variáveis x_{ij} e as variáveis de tempo t_i .

JAW, ODoni, PSARAFTIS e WILSON (1986) desenvolveram um novo procedimento heurístico para o problema *Dial-a-Ride* com restrições time-windows.

A heurística é do tipo *inserção-sequencial* e aloca passageiros para veículos determinando para cada um, os seus horários de embarque e desembarque.

Testes computacionais foram efetuados para exemplos numéricos com até 2.600 passageiros e com até 20 atividades simultâneas para os veículos.

GRAHAN e NUTTHE (1986) efetuaram comparações entre várias heurísticas do problema de scheduling de ônibus escolar, na presença de restrições time-windows.

As heurísticas analisadas foram adaptações dos métodos de ORLOFF (1976), SWERSEY e BALLARD (1984) e LEVIN (1971).

Uma análise detalhada sobre o estudo do *pior caso* de procedimentos heurísticos de problemas de roteamento e scheduling de veículos com restrições time-windows (PRSTW) foi apresentado por SOLOMON (1986).

Foram considerados os (PRSTW) simétricos, onde cada veículo efetua uma única viagem, frota heterogênea de veículos, uma única origem além de restrições de

Time-Windows e supondo válidas as desigualdades triangulares.

O autor analisou heurísticas das classes *savings*, *vizinho mais próximo*, *inserções*, e outros, concluindo que estas heurísticas para o (PRSTW) possuem uma ordem de complexidade que *não podem ser limitadas por uma constante*, o que comprova a tese de que os (PRSTW) são mais difíceis de se resolver do que os (PRV) ou (PSV) sem restrições time-windows.

FERLAND e FORTIN (1987) introduziram um novo procedimento heurístico para o (PRSTW) com time-windows móveis (não fixos).

FERLAND e MICHELON (1987) descreveram formulações e algoritmos heurísticos para o (PRSTW) no caso particular onde a frota de veículos é homogênea, com um único depósito de onde os veículos saem e retornam ao final da sua jornada, supondo ainda que cada tarefa deve ser efetuada por um único veículo e numa única visita.

Os autores fizeram algumas previsões sobre o comportamento das heurísticas para o (PRSTW) a partir dos modelos de Problemas de roteamento sem Time-Windows encontrados na literatura, sem no entanto efetuarem testes computacionais.

PALETTA (1987) apresentou uma nova formulação matemática para um problema de despacho de veículos com restrições time-windows.

O problema envolve um Sistema de distribuições de um produto gerado por uma unidade produtora para um conjunto de n usuários $I = \{1, 2, \dots, n\}$ onde cada usuário $i \in I$ está associado a um vetor (q_i, T_i, K_i) , onde q_i indica a quantidade do produto requerido por i , T_i os períodos onde a quantidade fixa q_i é requerido por i , e K_i indica o número de períodos entre duas entregas para i ($1 \leq K_i \leq T_i$), também conhecido com restrições de janela de tempo.

Neste trabalho, foi proposto uma nova heurística para este problema ilustrado por experimentos computacionais efetuados.

SOLOMON (1987) apresentou um novo resumo sobre algoritmos heurísticos para o problema de roteamento e scheduling de veículos com restrições time-windows (PRSTW), onde a frota é homogênea, e cada destino é atendido por um único veículo numa única visita, saindo todos de uma única origem.

O artigo escrito preliminarmente em (1984) tratou das seguintes classes de heurísticas para o (PRSTW).

-savings (economias)

-*Nearest-Neighbor* (vizinho mais próximo)

-*Insertion* (inserção)

-*Sweep* (varredura)

Nos experimentos computacionais, o autor revela a superioridade das heurísticas da classe *insertion* sobre as demais, explicado em parte pelo domínio das restrições *time-windows* nas alocações de tarefas para os veículos.

Um dos *surveys* mais recentes sobre o (PRSTW) foi desenvolvido por *DESROCHERS, SAVELSBERGH e SOUMIS (1988)*.

Neste trabalho, foram apresentados diversas formulações e algoritmos exatos e aproximados para um conjunto de (PRSTW) - como o problema do despacho de veículos, o problema dos *M* caixeiros viajantes, o problema *Dial-a-Ride*, todos com restrições adicionais de *Time-Windows*.

Outro artigo recente sobre o (PRSTW) foi desenvolvido por *SOLOMON, BAKER e SCHAFFER (1988)* que analisaram a eficiência de várias implementações de heurísticas para o (PRSTW).

CAPÍTULO III

O PROBLEMA APLICATIVO (FP)

INTRODUÇÃO

Embora os problemas de roteamento de veículos seja um assunto muito estudado na literatura com inúmeras aplicações (veja *BODIN, GOLDEN, ASSAD e BALL (1983) e ASSAD (1988)*), não existe de nosso conhecimento, muitos trabalhos publicados sobre roteamento e scheduling de navios.

Dentre as publicações encontradas podemos destacar os trabalhos de *APPELGREN (1969 e 1971)*; *LADERMAN, GLEIBERMAN e EGAN (1966)*, *RONEN (1982 e 1986)*, *BECKER (1987) e BROW, GRAVES e RONEN (1987)*.

Neste capítulo, daremos um breve resumo sobre o trabalho de *BECKER (1987)* que serviu de modelo inicial para o desenvolvimento das formulações e das técnicas de solução apresentadas nos capítulos posteriores deste trabalho.

O PROBLEMA (FP)

Uma das aplicações do problema de roteamento de veículos desenvolvidos no Brasil, está relacionada com a programação de uma frota de navios petroleiros de médio e grande porte, em longo curso, da *Frota Nacional de Petroleiros (FRONAPE)*, órgão ligado ao departamento de transportes da *PETROBRÁS*.

O problema foi estudado inicialmente por *BECKER (1987)* que descreveu detalhadamente, propondo uma solução através de uma técnica de otimização linear.

Posteriormente, foram desenvolvidos novos resultados sobre a sua teoria e técnicas de solução por *BECKER, MACULAN e OCHI (1988)*, *OCHI (1988)*, e *OCHI e MACULAN (1989)*.

Descrevemos a seguir, uma síntese deste problema baseado no trabalho de *BECKER (1987)*, que passamos a denotar por (FP) "*FRONAPE-PETROBRÁS*".

O serviço de transportes de longo curso da *FRONAPE* é baseado em contratos de transportes de petróleo e seus derivados para a *PETROBRÁS* e de minério de ferro e de alguns produtos químicos para terceiros.

Para o atendimento destes contratos, a *FRONAPE* dispõe atualmente (*) de cerca de 35 navios entre próprios e afretados, constituídos de petroleiros e minéro-petroleiros.

Num horizonte de planejamento de curto prazo, aproximadamente 60 dias, existem cerca de 50 contratos (trips) a serem analisados.

A complexidade do problema (FP) se deve principalmente ao elevado número de restrições impostas para aproximar o modelo matemático ao problema real.

Algumas destas restrições se relacionam com os navios, portos de origem e destino dos trips, tipos de carga de cada trip, velocidade do navio, clima, avarias nos navios, tempos de espera para atracar e/ou carregar e/ou descarregar não previstos, condições do casco do navio, estado e tipo de pintura do navio, período de docagem, corrente marítima, tempo limite para o navio concluir um trip, intervalos de tempo (*time-window*) para o navio aportar na origem de um trip pronto para operar, restrições de compatibilidade entre navio-porto, navio-carga, navio-rota, navio-bandeira, custos diferenciados entre navios próprios e afretados, capacidade do navio em relação a carga do trip, a posição do navio no início de um planejamento, etc.

(*) dados encontrados em BECKER (1987)

A programação dos trips, cuja sequência gera um *schedule* está sujeito a frequentes revisões, podendo ocorrer modificações no espaço de poucos dias. Desta forma, num planejamento pode-se alocar trips para um navio ainda não incorporado à frota disponível de navios.

Esta situação pode ocorrer por exemplo, quando todos os navios da frota se mostrarem inviáveis para efetuar estes trips, obrigando a empresa a afretar emergencialmente um ou mais navios no mercado.

Neste tipo de afretamento, o contrato é normalmente feito para apenas uma viagem (um trip).

Quanto ao objetivo, o problema consiste em programar navios no horizonte de planejamento, gerando um conjunto de *schedules*, um para cada navio, de modo a atender todos os trips analisados, minimizando os custos fixos e variáveis do problema.

Uma palavra sobre a interpretação dos tempos de janela (*time-windows*) do problema (FP).

O intervalo $[a_i, b_i]$ de cada trip i , no problema (FP) é o intervalo de tempo dentro do qual os navios alocados a este trip, devem aportar no porto de sua origem prontos para operarem.

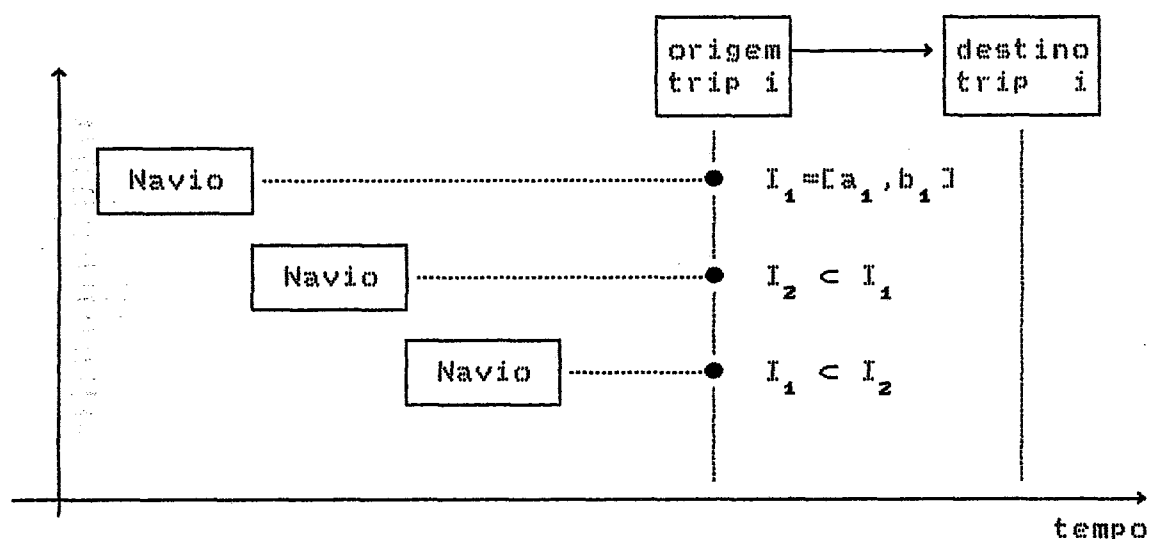
Na prática, os intervalos *time-windows* não permanecem fixos até o seu atendimento.

Na programação real, utilizada atualmente pela *FRONAPE* (*), não existe nenhuma otimização nos *time-windows*, assim, para efeito prático, o tempo de chegada no trip i pode ser colocado a priori igual a b_i (o pior caso numa otimização de tempos).

No início de cada planejamento, é fornecido o *time-window* inicial $I_1 = [a_i, b_i]$ que será o intervalo de tempo utilizado nas formulações matemáticas deste trabalho.

À medida que aproximarmos da origem do trip, o (s) operador (es) do (s) navio (s) alocado (s) para este trip, devem fornecer a administração do porto origem do trip i , intervalos mais precisos sobre o seu dia de chegada, com isso, o intervalo I_1 será reduzido para um intervalo I_2 contido em I_1 . Tal procedimento será efetuado até que o último intervalo fornecido pelos operadores dos navios, feito a poucos dias, do dia de chegada, se restringe a um único ponto (único dia), como mostra a figura 3.1 a seguir.

(*) veja BECKER (1987)



INTERPRETAÇÃO DO TIME-WINDOW NO (PF)

FIGURA 3.1

Um dos fatores que torna o problema (FP) mais complexo que outros problemas de roteamento e scheduling de veículos, refere-se à necessidade de se trabalhar com variáveis contínuas de tempo t_{ni}^o = tempo necessário para o navio n aportar na origem do trip i .

Além disso, enquanto na maioria dos problemas de roteamento com ou sem *time-windows*, os veículos são do mesmo tipo (frota homogênea) e estão todos localizados numa única origem no início de cada planejamento, isto não ocorre no problema (FP), onde a frota de navios é heterogênea e cada

navio pode estar em qualquer nó ou arco da rede no início de cada planejamento.

Assim, o tempo de chegada de um navio n a um determinado trip (nó da rede transformada), no início de planejamento, depende da localização do navio no instante zero do horizonte e das velocidades usadas pelo navio n para chegar à origem deste trip (que supomos serem constantes).

No problema (FP), t_{ni}^0 possui um valor mínimo inicial igual a e_{nk}^0 que é igual a zero se o navio n já estiver na origem do trip k , no início do seu planejamento.

Assim, existem dois casos a analisar:

i) $t_{ni}^0 = e_{ni}^0$, se o trip i for o trip inicial do *schedule* do navio n .

ii) $t_{ni}^0 \geq e_{nk}^0 + t_{nvk} + t_{nv'ki}$, se $i \neq k$, onde i é o trip posterior ao trip k , no *schedule* do navio n , e t_{nvk} e $t_{nv'ki}$ são respectivamente os tempos para o navio n efetuar o trip k , na velocidade v e o tempo para n efetuar na velocidade v' , o trecho entre o final do trip k com o início do trip i (*arco intertrip*).

Somente após t_{ni}^0 ou t_{nk}^0 (tempo de chegada de n ao trip i na combinação de velocidades utilizadas verificarem a condição i) ou ii) é que será analisado o atendimento ou não

das restrições *time-windows* do trip i .

BECKER (1987) resolveu o problema (FP) em duas etapas.

Inicialmente foi desenvolvido um gerador de *schedules*, através de um procedimento de enumeração explícita que fornece um conjunto de *schedules* viáveis para cada navio, descartando os dominados por critérios previamente estabelecidos.

Na etapa seguinte, o problema (FP) foi descrito como um modelo de recobrimento com restrições adicionais (MRA) descrito de forma semelhante ao modelo F1 desenvolvido no capítulo III deste trabalho.

O (MRA) foi resolvido pelo programa MPSX-MIP da IBM que usa SIMPLEX e Branch-and-Bound para obter soluções lineares inteiras.

Resultados computacionais mostraram que em vários exemplos, o simplex já fornece solução ótima inteira.

CAPÍTULO IV

FORMULAÇÕES MATEMÁTICAS PARA O PROBLEMA (FP)

INTRODUÇÃO

Apresentamos aqui um conjunto de formulações matemáticas para o problema (FP).

Propomos vários tipos de modelos de Programação Linear Inteira, desde o modelo clássico de recobrimento (formulação 1) até formulações mais complexas que analisam as restrições do problema numa única etapa (formulações 2,3,4,5,6 e 7).

Em termos práticos as formulações 1 e 2 se aproximam mais do problema real, embora as demais formulações mantenham as principais restrições do problema.

IV.1) FORMULAÇÃO 1 (F1)

Nesta formulação o problema (FF) é particionado em duas etapas.

A primeira etapa consiste em gerar todos os schedules viáveis do problema, a menos dos redundantes e em alguns casos, quando for possível uma comparação, a exclusão de schedules dominados segundo critérios de custo e/ou de tempo.

A vantagem de se trabalhar com todos os schedules viáveis vem a longo prazo, pois uma decisão de conservar um schedule *caro* no atual horizonte de planejamento pode ser compensado nos planejamentos futuros com a encampação de um maior número de trips no prolongamento deste schedule.

De fato, um schedule pode ser considerado *caro* se o custo deste schedule for elevado, mas por outro lado, a elevação de custos pode estar relacionada com uma redução do tempo para efetuar-lo, possibilitando aí um número maior de trips candidatos nos planejamentos posteriores.

O problema da geração dos schedules na primeira etapa, consiste em verificar o atendimento de algumas restrições espaciais e das restrições de tempo do problema.

Na etapa seguinte, busca-se dentre os schedules não descartados na fase inicial, um conjunto de schedules que cubram todos os trips a um menor custo global possível.

Este último problema pode ser descrito como um problema *set-partitioning* ou *set-covering* com restrições adicionais.

Trabalhamos aqui com uma formulação do tipo *set-covering* com restrições adicionais, visto que ela é mais abrangente que a versão *set-partitioning* com restrições adicionais.

Veremos neste capítulo que, embora raras, existem algumas situações reais que impossibilitam o uso de um modelo *set-partitioning*.

Nesta formulação, vamos supor conhecido o conjunto de *schedules* viáveis do problema, dedicando-nos somente com a segunda etapa do problema.

Para a etapa inicial podemos usar desde algoritmos de enumeração explícita até heurísticas que priorizam um dos objetivos conflitantes custo ou tempo, descartando com isso parte dos schedules viáveis dominados segundo o critério utilizado (custo ou tempo).

NOTAÇÃO DA FORMULAÇÃO 1

$N = \{N\}$ = Frota heterogênea de navios disponíveis.

$K = \{k\}$ = Conjunto de trips a serem analisados no atual horizonte de planejamento.

R = conjunto de restrições locais do problema (compatibilidades entre navio/carga, navio/porto, navio/demanda, navio/percurso, etc).

$(N_k \subset N)$ = conjunto de navios de N compatíveis com o trip $k \in K$, segundo R .

$(K_n \subset K)$ = conjunto de trips de K compatíveis com o navio $n \in N$, segundo R

$S(n)$ = conjunto de schedules viáveis do navio $n \in N$

$S = \bigcup_{n \in N} S(n)$ = conjunto de schedules viáveis do problema.

C_s = custo global do schedule $s \in S$.

$E = (e_{ks})$ = matriz dos coeficientes das restrições de recobrimento, onde

$$e_{ks} = \begin{cases} 1, & \text{se o trip } k \text{ pertence ao schedule } s \in S \\ 0, & \text{c/c.} \end{cases}$$

$$x_s = \text{var. binária} = \begin{cases} 1, & \text{se o schedule } s \text{ está na solução} \\ 0, & \text{c/c.} \end{cases}$$

FORMULAÇÃO MATEMÁTICA (F1)

A formulação (F1) é descrita como um problema de recobrimento com restrições adicionais (SCA).

$$V(F1) = \text{Mínimo} \sum_{s \in S} C_s x_s \quad (4.1)$$

$$\text{S.A.} \quad \sum_{s \in S} e_{ks} x_s \geq 1, \quad \forall k \in K \quad (4.2)$$

$$\sum_{s \in S(n)} x_s \leq 1, \quad \forall n \in N \quad (4.3)$$

$$x_s \in \{0, 1\}, \quad \forall s \in S \quad (4.4)$$

O conjunto de restrições (4.2), nos diz que cada trip $k \in K$ deve estar contido em pelo menos um schedule $s \in S$.

As restrições adicionais (4.3) requerem que cada navio $n \in N$ deve ter no máximo um schedule na solução final e finalmente (4.4) são as variáveis zero-um do problema.

A formulação (SCA) permite por (4.2) que um mesmo trip

$k \in K$ esteja incluído em mais de um schedule $s \in S$ na solução final do problema, o que na prática significa efetuar, sem necessidade, o trip k mais de uma vez.

Uma alternativa para evitar esta redundância seria substituir as restrições (4.2) de (SCA) por restrições do tipo (4.2') $\sum_{s \in S} e_{ks} x_s = 1, \forall k \in K$, transformando o problema (SCA) num problema *set-partitioning* com restrições adicionais (4.2').

Mas este procedimento pode tornar o problema inviável em alguns casos como mostramos no exemplo dado a seguir.

Exemplo: Suponha que um trip $k \in K$ esteja contido em todos os schedules $s \in S$, esta situação pode ocorrer com exclusão prévia de schedules dominados seguindo algum critério estabelecido. De fato, se todos os schedules viáveis fossem armazenados poderíamos ter o seguinte caso.

TRIPS	SCHEDULES					
	s_1	s_2	s_3	s_4	s_5	s_6
k	1	1	0	1	1	0
i	1	0	1	0	0	0
j	0	0	0	1	0	1

Tabela 4.1

Existem inicialmente três schedules viáveis (s_1, s_2, s_3) para o navio $n_1 \in N$ e três schedules viáveis (s_4, s_5, s_6) para o navio $n_2 \in N$. Neste caso, não seria possível um trip k estar contido em todos os seis schedules do problema.

Mas os schedules s_3 e s_6 poderiam ser descartados na seguinte situação, dado a seguir:

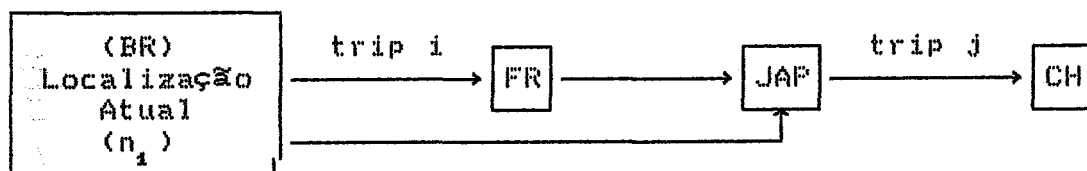


FIGURA 1.1

Podemos estabelecer os seguintes critérios:

i) Vale a pena $n_1 \in N$ efetuar somente o trip i pois $n_1 \in N$ já está localizado na origem do trip $i \in K$.

ii) Vale a pena $n_1 \in N$ efetuar os trips i e j já que o trecho em lastro que n_1 percorre neste percurso não é "muito elevado" (FR \rightarrow JAP).

iii) Não vale a pena n_1 efetuar somente j a partir da sua localização inicial (BR), já que o trecho em lastro para

n_1 seria muito longo (BR---)JAP) (veja Fig. 4.1)

Desse modo poderíamos pensar em descartar o schedule s_9 do conjunto $S(n_1)$ (veja Tabela 4.1).

Com um raciocínio semelhante, poderíamos encontrar situações reais para descartar o schedule s_6 de $S(n_2)$.

Após os descartes, as restrições (4.2') aplicadas à tabela 4.1 geram o seguinte sistema de equações lineares inconsistente.

$$\left. \begin{array}{rcl} X_1 + X_2 + & X_4 + X_5 & = 1 \\ X_1 & & = 1 \\ & X_4 & = 1 \end{array} \right] \quad \text{(sistema inconsistente)}$$

Por outro lado, usando (4.2) ao invés de (4.2') na tabela (4.1), juntamente com (4.4) teríamos um sistema de equações lineares consistente da forma:

$$\left. \begin{array}{rcl} X_1 + X_2 & + X_4 + X_5 & \geq 1 \\ X_1 & & = 1 \\ & X_4 & = 1 \end{array} \right] \quad \text{(consistente)}$$

IV-2) FORMULAÇÃO (F2)

A proposta desta formulação é a de representar o problema (FP) de uma forma mais próxima possível do problema real.

A única simplificação visível que fazemos é quando as variáveis contínuas de tempo t_n^0 que são determinados sem especificar a combinação de velocidades utilizadas pelo navio $n \in N$ para percorrer o trecho entre o destino do trip i efetuado mais recentemente (ou a sua atual localização) até origem do trip $j \in K$.

Na prática isso não representa maiores problemas, pelo contrário tal procedimento pode ser visto como uma liberdade dada ao operador do navio n para decidir a que velocidade deve ser percorrido o trecho em lastro (i,j) desde que atenda às restrições de tempo dos trips j e seus sucessores no schedule.

Assim os valores t_{ni}^0 , t_{nj}^0 , serão gerados sem especificar as velocidades utilizadas, mas com a certeza de que eles satisfazem todas as restrições de "scheduling" do problema.

Por exemplo, se i e j são dois trips do schedule do navio $n \in N$, onde i é o trip efetuado por n mais recentemente

e j o trip a ser efetuado após i , com

$$[a_j, b_j] = [30, 35] = \text{time-window de } j$$

$$t_{ni}^0 = 15, y_{nvi} = 1, t_{nvi} = 5, t_{nvi j} = 8$$

$t_{nvi j} = 10, t_{nvi j} = 15$, neste caso teremos:

$$t_{ni}^0 + t_{nvi} + t_{nvi j} = 28 \in [30, 35]$$

$$t_{ni}^0 + t_{nvi} + t_{nvi j} = 30 \in [30, 35]$$

$$t_{ni}^0 + t_{nvi} + t_{nvi j} = 35 \in [30, 35]$$

Portanto o operador do navio $n \in N$ teria as opções de percorrer o trecho em lastro (i,j) ou na velocidade constante v' ou v'' .

Nesta formulação, todas as restrições do problema são tratadas numa única etapa, tendo como objetivo gerar um conjunto de schedules de custo global mínimo.

Vamos supor aqui que é possível ordenar no tempo o conjunto de trips $K = (k_1, k_2, \dots, k_n)$, onde $k_i < k_j$ significa que se k_i e k_j estão num mesmo schedule, então o trip j deve ser atendido após o trip i .

NOTAÇÃO DA FORMULAÇÃO (F2)

$N = \{n\}$ = frota heterogênea de navios disponíveis

$K = \{k\}$ = conjunto de trips, ordenado no tempo, do atual horizonte de planejamento.

$V = \{v\}$ = conjunto das velocidades dos navios

R = conjunto das restrições locais do problema (ex: compatibilidade entre navio-carga, navio-porto, navio-percurso, navio-bandeira, navio-demanda, etc).

$N_k \subset N$ = conjunto de navios que atendem às restrições R do trip $k \in K$.

$K_n \subset K$ = conjunto de trips que atendem às restrições locais R do navio $n \in N$.

$PK(k)$ = conjunto de trips de K , posteriores (no tempo) ao trip $k \in K$.

$AK(k)$ = conjunto de trips de K , anteriores (no tempo) ao trip $k \in K$.

q_{ni} = capacidade do navio $n \in N$, referente à carga do trip $i \in K$.

D_i = Demanda do trip $i \in K$.

c_{nvi} = custo de transporte do navio n percorrer em lastro o arco (i, j) na velocidade $v \in V$.

c_{nvi} = custo para o navio $n \in N$ efetuar o trip i , na velocidade $v \in V$.

OBS 1: Os tempos envolvidos nesta formulação são quantidades em dias contados à partir do início do atual

horizonte de planejamento.

e_{nk}^0 = tempo inicial estimado para o navio $n \in N$, aportar pronto para operar, na origem do trip i , no início do schedule do navio $n \in N$.

t_{nj}^0 = variável contínua de tempo estimado para o navio $n \in N$ aportar pronto para operar, na origem do trip $j \in K$. Em particular, se j é o primeiro trip do schedule, então $t_{nj}^0 = e_{nj}^0$.

t_{nvi} = tempo para o navio n efetuar o trip i na velocidade $v \in V$.

$t_{nvi j}$ = tempo para o navio n percorrer em lastro o trecho entre o final do trip i e a origem do trip j , na velocidade $v \in V$.

U_i = tempo limite para concluir o trip $i \in K$.

$[a_i, b_i]$ = time-window do trip i , ou o intervalo de tempo dentro do qual os navios alocados para este trip devem aportar na origem de i , prontos para operarem.

$x_{nvi j}$ = variável binária =
$$\begin{cases} 1, & \text{se o navio } n \text{ percorre o arco} \\ & (i, j), \text{ na velocidade } v \in V. \\ 0, & \text{c/c.} \end{cases}$$

y_{nvi} = variável binária =
$$\begin{cases} 1, & \text{se o navio } n \text{ efetua o trip } i \\ & \text{na velocidade } v \in V. \\ 0, & \text{c/c.} \end{cases}$$

OBS 2 : O problema dos tempos iniciais t_{ni}^0

A princípio nas formulações que se seguem, os valores t_{ni} devem satisfazer:

$$(*) \quad t_{ni}^0 \geq e_{ni}^0,$$

$$(**) \quad a_i \leq t_{ni}^0 \leq b_i, \quad \forall i \in K, \quad \forall n \in N$$

As restrições (*) podem ser suprimidas se efetuarmos previamente as seguintes modificações:

Para cada navio $n \in N$ faça:

$$a_i = \text{máximo} (a_i, e_{ni}^0), \quad \forall i \in K$$

$$\text{se } a_i \leq b_i \quad \text{OK!}$$

$$\text{se } a_i > b_i, \quad \text{faça } C_{nvi} = +\infty, \quad \forall v \in V$$

Tal procedimento é efetuado para cada navio $n \in N$ e implicitamente obriga que

$$\sum_{v \in V} y_{nvi} = 0 \quad \text{e} \quad x_{nvi} = 0 \quad \text{se } a_i > b_i$$

FORMULAÇÃO MATEMÁTICA (F2)

1 - Função objetivo : deseja-se minimizar o custo global de efetuar todos os trips.

$$V(F2) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in PK(i)} c_{nvi j} x_{nvi j} + \\ + \sum_{n \in N} \sum_{i \in V} \sum_{i \in K} c_{nvi} y_{nvi}$$

2 - Satisfação da demanda de cada trip $i \in K$

$$\sum_{n \in N_i} \sum_{v \in V} q_{ni} y_{nvi} \geq D_i, \forall i \in K$$

Este conjunto de restrições nos diz que a capacidade total referente à carga do trip i dos navios a ele alocados deve ser maior ou igual à demanda do trip i .

A necessidade de usar uma desigualdade do tipo maior ou igual (\geq) ao invés de uma igualdade em (2) é pelo fato de que, em alguns casos, a soma das capacidades q_{ni} não coincide com D_i .

3 - Ligação entre trechos em lastro (i, j) e o atendimento dos trips i e j .

$$\sum_{i \in K} \sum_{v \in V} x_{nvi j} = \sum_{v \in V} y_{nvi j} \leq 1, \forall n \in N, \forall j \in PK(i)$$

Estas restrições têm a seguinte interpretação:

$$3.A) \quad \sum_{i \in K} \sum_{v \in V} x_{nvi j} = \sum_{v \in V} y_{nv j}$$

(\Rightarrow) Neste sentido, se um navio $n \in N$ chega na origem do trip j , a uma dada velocidade v , então este navio deve necessariamente efetuar o trip $j \in K$, em alguma velocidade $v \in V$.

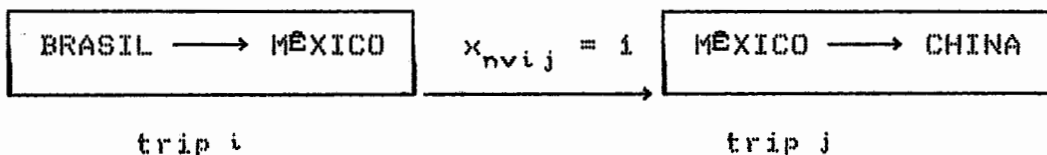
(\Leftarrow) Reciprocamente, se o navio n efetua o trip $j \in K$ então n deve necessariamente chegar em j de algum trip i , anterior a j ($i \leq j$).

No caso particular onde o navio $n \in N$ já está na origem do trip j e efetua j (portanto n não *chega* em j), existem dois casos a se analisar.

i) O destino do trip i é igual à origem do trip j .

ii) O trip j é o primeiro trip do schedule do navio n .

No caso i), teremos por exemplo, a seguinte configuração:



Nos dois casos i) e ii) geramos um trip artificial i_0 , anterior a j , criado previamente somente para atender esta restrição. Tal procedimento é feito se necessário para todo $n \in N$ e todo $j \in K$ com $c_{nvi_0j} = 0$

3.B) Interpretação da desigualdade: $\sum_{v \in V} y_{nvj} \leq 1, \forall n \in N$
e $\forall j \in K$.

Significa que todo trip j deve ser efetuado no máximo uma vez durante o atual horizonte de planejamento, por cada navio $n \in N$. A possibilidade de vários navios efetuarem o mesmo trip $j \in K$ é descartada pelo custo positivo correspondente da função objetivo.

3.C) Finalmente $\sum_{i \in K} \sum_{v \in V} X_{nvi_j} \leq 1, \forall n \in N, \forall j \in PK(i)$ nos diz que cada navio $n \in N$ deve visitar o trip j no máximo uma vez.

OBS.3: Devemos tomar um certo cuidado ao analisar o caso (3.C) onde os nós da rede agora são trips e cada trip contém no mínimo dois nós da rede física original, que são respectivamente a origem e o destino do trip.

Assim observe que a condição (3.C) não impede por exemplo que ocorra a seguinte situação:

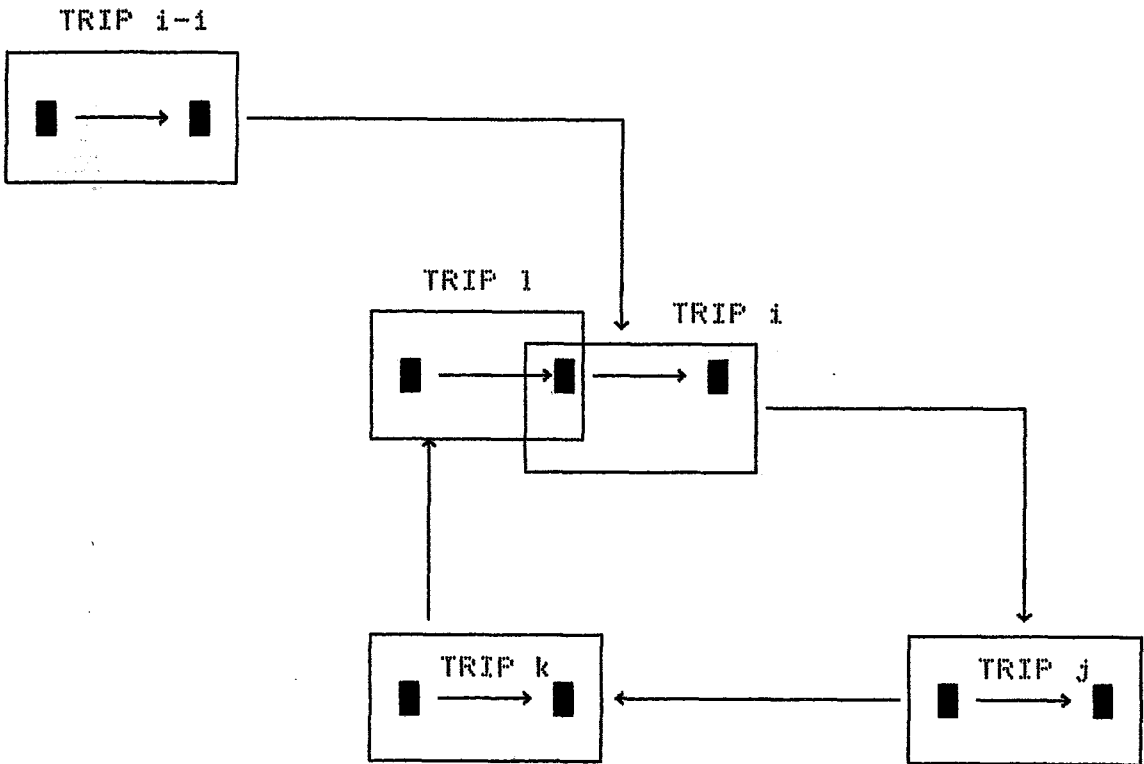


FIGURA 4.2

Na figura 4.2, temos que $(i-1) < i < j < k < l$ e um mesmo navio $n \in N$ chegou à origem do trip i duas vezes, mas efetuando dois trips distintos a saber, o trip i e o trip l .

Isto é, o navio n não chegou ao trip i duas vezes, já que na segunda visita à origem do trip i , na verdade, ele chegou ao destino do trip l , distinto de i .

4 - Variáveis binárias zero-um:

$$X_{nvi j} = \begin{cases} 1, & \text{se o navio } n, \text{ na velocidade } v, \text{ percorre} \\ & \text{ocioso o arco } (i,j) \\ 0, & \text{c/c.} \end{cases}$$

$$y_{nvj} = \begin{cases} 1, & \text{se o navio } n, \text{ na velocidade } v, \text{ efetua o} \\ & \text{trip } j \\ 0, & \text{c/c.} \end{cases}$$

$$\forall n \in N, \forall v \in V, \forall i \in K, \forall j \in P^K(i)$$

As variáveis $X_{nvi j}$ são definidas somente para $i < j$.

5 - Atendimento do tempo para um navio $n \in N$, chegar à origem do trip j , vindo de i , pronto para operar (ligação entre o problema de roteamento e *scheduling*).

$$t_{ni}^0 + t_{nvi} y_{nvi} + t_{nv'ij} - t_{nj}^0 \leq (2 - \sum_{v \in V} (y_{nvj} + x_{nvi j})) M_{nvv'ij}$$

$$\text{onde } M_{nvv'ij} \geq b_i + t_{nvi} + t_{nv'ij} - a_j, \forall n \in N, \forall v, v' \in V, \\ \forall i \in K, \forall j \in K.$$

O conjunto de restrições (5) nos diz que se o navio n percorre o arco (i,j) em lastro, onde $i < j$, então o seu tempo de chegada à origem do trip j é igual ou maior ao tempo de n chegar e efetuar o trip i somado com o tempo de n percorrer o arco (i,j) .

Existem quatro casos a serem analisados:

Note que por (3), $\sum_{v \in V} y_{nvj} \leq 1$ e $\sum_{v \in V} x_{nvi} \leq 1$ e por

$$(4) \sum_{v \in V} y_{nvj} \in \{0, 1\}, \sum_{v \in V} x_{nvi} \in \{0, 1\}, \forall n \in N, \forall j \in PK(i), \forall i \in K.$$

CASO 1: Quando $\sum_{v \in V} y_{nvj} = 0$ e $\sum_{v \in V} x_{nvi} = 0$, neste caso as inequações (5) são redundantes. Isto é, se o navio n não efetua o trip j e se n não percorre o arco (i, j) , então não há necessidade de satisfazer as restrições do tempo do trip j .

CASO 2 : Quando $\sum_{v \in V} y_{nvj} = 1$ e $\sum_{v \in V} x_{nvi} = 1$, neste caso $t_{ni}^0 + t_{nvi} + t_{nvi} \leq t_{nj}^0$, se n efetuou o trip i , $i < j$.

Ou seja, se n percorre (i, j) e efetua o trip j então n deve atender às restrições de tempo do trip j .

CASO 3: $\sum_{v \in V} y_{nvj} = 0$ e $\sum_{v \in V} x_{nvi} = 1$, esta situação não se verifica pois $\sum_{v \in V} x_{nvi} = 1 \implies \sum_{i \in K} \sum_{v \in V} x_{nvi} = 1 \implies \sum_{v \in V} y_{nvj} < \sum_{v \in K} \sum_{v \in V} x_{nvi}$, contrariando (3).

O caso 3 na prática significa proibir o caso de n chegar ao trip j e não efetuar j .

CASO 4: $\sum_{v \in V} y_{nvj} = 1$ e $\sum_{v \in V} x_{nvi} = 0$, neste caso (5) é redundante. Note que este caso, ao contrário do caso 3, não

contraria (3) pois $\sum_{v \in V} x_{nvi j} = 0$ não implica necessariamente que $\sum_{i \in K} \sum_{v \in V} x_{nvi j} = 0$.

6 - Satisfação do tempo limite U_j para concluir o trip $j \in K$.

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} y_{nvj}) M_{nvj}, \text{ onde}$$

$$M_{nvj} \geq b_j + t_{nvj}, \quad \forall n \in N, \quad \forall v \in V, \quad \forall j \in K$$

Pelas restrições (3.B) e (4), sabemos que $\sum_{v \in V} y_{nvj} \in (0, 1) \quad \forall n \in N, \quad \forall j \in K$. Logo existem dois casos a se analisar.

CASO 1: $\sum_{v \in V} y_{nvj} = 1 \implies U_j \geq t_{nj}^0 + t_{nvj}, \quad \forall j \in K, \quad \forall n \in N, \quad \forall v \in V$

CASO 2: $\sum_{v \in V} y_{nvj} = 0 \implies U_j \geq t_{nj}^0 + t_{nvj} - M_{nvj}, \quad \forall j \in K, \quad \forall n \in N, \quad \forall v \in V$ (redundante).

7 - Satisfação das restrições time-windows.

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall n \in N, \quad \forall j \in K.$$

Portanto os conjuntos das restrições 1 a 7 descrevem o seguinte problema de programação linear mista.

$$V(F2) = \text{Mínimo} \sum_{n \in N} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} c_{nvi j} x_{nvi j} + \sum_{n \in N} \sum_{v \in U} \sum_{j \in K} c_{nvj} y_{nvj} \quad (4.5)$$

$$\text{S.A.} \quad \sum_{n \in N_i} \sum_{v \in U} q_{ni} y_{nvi} \geq D_i, \quad \forall i \in K \quad (4.6)$$

$$\sum_{i \in K} \sum_{v \in U} x_{nvi j} = \sum_{v \in U} y_{nvj} \leq 1, \quad \forall n \in N, \quad \forall j \in PK(i) \quad (4.7)$$

$$x_{nvi j} \in (0, 1), \quad y_{nvj} \in (0, 1), \quad \forall n \in N, \quad \forall v \in U, \quad \forall i \in K, \quad \forall j \in PK(i) \quad (4.8)$$

$$t_{ni}^0 + t_{nvi} y_{nvi} + t_{nv'ij} - t_{nj}^0 \leq (2 - \sum_{v \in U} (y_{nvj} + x_{nvi j})) M_{nv'ij}, \quad \text{onde } M_{nv'ij} \geq b_i + t_{nvi} + t_{nv'ij} - a_j, \quad \forall n \in N, \quad \forall v, v' \in U, \quad \forall i \in K, \quad \forall j \in PK(i) \quad (4.9)$$

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in U} y_{nvj}) N_{nvj}, \quad \text{onde} \quad (4.10)$$

$$N_{nvj} \geq b_j + t_{nvj}, \quad \forall n \in N, \quad \forall v \in U, \quad \forall j \in K.$$

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall j \in K, \quad \forall n \in N \quad (4.11)$$

IV-3) FORMULAÇÃO 3 (F3)

Esta formulação é uma variação da formulação anterior (F2), onde são eliminadas as variáveis de fluxo x_{nvi} , passando a trabalhar somente em função das variáveis de decisão y_{nvi} (que passamos a chamar de x_{nvi}) e das variáveis contínuas de tempo t_{ni}^0 .

NOTAÇÃO DE (F3)

$N = \{n\}$ = frota heterogênea de navios disponíveis.

$K = \{k\}$ = conjunto ordenado de trips.

$V = \{v\}$ = conjunto de velocidade dos navios.

R = conjunto das restrições locais do problema (veja notação (F2))

$N_k \subset N$ = conjunto de navios de N que atendem às restrições R do trip $k \in K$.

$K_n \subset K$ = conjunto de trips de K que atendem às restrições locais R do navio $n \in N$.

$PK(i)$ = conjunto de trips de K , posteriores ao trip $i \in K$.

$AK(i)$ = conjunto de trips de K , anteriores ao trip $i \in K$.

q_{ni} = capacidade do navio $n \in N$, referentes a carga do trip i .

D_i = demanda do trip $i \in K$.

c_{nvi} = custo global para o navio $n \in N$ efetuar o trip i , na velocidade $v \in V$.

t_{nij} = tempo para o navio $n \in N$ percorrer em lastro o arco (i, j) (supomos aqui a velocidade constante).

t_{nj}^0 = variável contínua do tempo de chegada do navio n à origem do trip $j \in K$. Em particular se j é trip inicial do schedule então $t_{nj}^0 = e_{nj}^0$ onde

e_{nj}^0 = tempo constante estimado para que o navio n aporte na origem do trip j , no início do schedule do navio $n \in N$.

x_{nvi} = variável binária = $\begin{cases} 1, & \text{se o navio } n, \text{ atende o trip} \\ & i, \text{ na velocidade } v \in V \\ 0, & \text{c/c} \end{cases}$

FORMULAÇÃO MATEMÁTICA (F3)

1 - Função objetivo: deseja-se minimizar o custo global para efetuar todos os trips $k \in K$.

$$V(F3) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} x_{nvi}$$

2 - Satisfação da demanda de cada trip $i \in K$

$$\sum_{n \in N_i} \sum_{v \in V} q_{ni} x_{nvi} \geq D_i, \quad \forall i \in K$$

3 - Variáveis de decisão zero-um:

$$x_{nvi} \in \{0, 1\}, \forall n \in N, \forall v \in V, \forall i \in K$$

4 - Restrições no número de visitas a um trip por cada navio

$$\sum_{v \in V} x_{nvi} \leq 1, \forall n \in N, \forall i \in K$$

5 - Restrições de acoplamento entre os problemas de roteamento e de *scheduling*.

$$t_{ni}^0 + t_{nvi} + t_{nij} - t_{nj}^0 \leq (2 - \sum_{v \in V} x_{nvi} - \sum_{v \in V} x_{nvj}) M_{nvi j}$$

$$\text{onde } M_{nvi j} \geq b_i + t_{nvi} + t_{nij} - a_j, \forall n \in N, \forall v \in V, \forall i \in K, \\ \forall j \in PK(i)$$

Pelas restrições (4) sabemos que:

$$\sum_{v \in V} x_{nvi} \leq 1, \forall n \in N, \forall i \in K, \text{ portanto:}$$

$$i) \text{ Se o navio } n \text{ efetua os trips } i \text{ e } j \implies \sum_{v \in V} x_{nvi} =$$

$$\sum_{v \in V} x_{nvj} = 1$$

$$\text{Então } t_{ni}^0 + t_{nvi} + t_{nij} - t_{nj}^0 \leq 0$$

ii) Se o navio n efetua i mas não efetua j

$$\sum_{v \in V} x_{nvi} = 1 \text{ e } \sum_{v \in V} x_{nvj} = 0 \implies t_{ni}^0 + t_{nvi} + t_{nij} -$$

$$- t_{nj}^0 \leq M_{nvi j} \quad (\text{redundante})$$

iii) Se o navio n efetua j mas não efetua i

$$\sum_{v \in V} x_{nvi} = 0 \text{ e } \sum_{v \in V} x_{nvj} = 1 \implies t_{ni}^0 + t_{nvi} + t_{nvj} - t_{nj}^0 \leq M_{nvi j} \quad (\text{redundante})$$

iv) Se o navio não efetua nem i nem j

$$\sum_{v \in V} x_{nvi} = \sum_{v \in V} x_{nvj} = 0 \implies t_{ni}^0 + t_{nvi} + t_{nij} - t_{nj}^0 \leq 2 M_{nvi j} \quad (\text{redundante})$$

6 - Satisfação dos *time windows*

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall n \in N, \quad \forall j \in K$$

7 - Satisfação do tempo limite para efetuar o trip $k \in K$.

$$U_k \geq t_{nk}^0 + \sum_{v \in V} t_{nvk} x_{nvk}, \quad \forall k \in K, \quad \forall n \in N$$

Os conjuntos de restrições 1 a 7 acima, descrevem o seguinte problema de programação linear mista:

$$V(F3) = \text{Mínimo } \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum c_{nvi} x_{nvi} \quad (4.12)$$

$$\text{S.A.} \sum_{n \in N_i} \sum_{v \in V} q_{ni} x_{nvi} \geq D_i, \quad \forall i \in K \quad (4.13)$$

$$x_{nvi} \in \{0, 1\}, \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K \quad (4.14)$$

$$\sum_{v \in V} x_{nvi} \leq 1, \quad \forall n \in N, \quad \forall i \in K \quad (4.15)$$

$$t_{ni}^0 + t_{nvi} + t_{nij} - t_{nj}^0 \leq (2 - \sum_{v \in V} x_{nvi} - \sum_{v \in V} x_{nvj}) M_{nvi j} \quad (4.16)$$

$$\text{onde } M_{nvi j} \geq b_i + t_{nvi} + t_{nij} - a_j, \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K, \\ \forall j \in PK(i)$$

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall n \in N, \quad \forall j \in K \quad (4.17)$$

$$U_j \geq t_{nj}^0 + \sum_{v \in V} t_{nvj} x_{nvj}, \quad \forall n \in V, \quad \forall j \in K, \quad \forall v \in V \quad (4.18)$$

IV.4 - FORMULAÇÃO 4 (F4)

Apresentamos aqui, uma formulação matemática para o problema (FP), onde é retirada a hipótese dos trips serem pré-ordenados no tempo. Além disso, propomos um outro esquema para resolver o problema da geração dos tempos t_{ni}^0 iniciais, através da inclusão de alguns trips do horizonte de planejamento anterior.

NOTAÇÃO DE (F4)

$N = \{n\}$ = frota heterogênea de navios disponíveis.

\bar{K} = conjunto de trips do atual horizonte de planejamento.

$K(n)$ = último trip alocado ao navio n no horizonte de planejamento anterior

$K = \bigcup_{n \in N} K(n)$ = conjunto dos trips efetuados mais recentemente por cada navio, nos planejamentos anteriores.

$K = K \cup \bar{K}$ = conjunto de trips considerados nesta formulação.

$V = \{v\}$ = conjunto de velocidade dos navios

R = conjunto das restrições locais do problema (veja notação de (F2))

$N_k \subset N$ = subconjunto de navios de N , que atendem às restrições R do trip $k \in K$.

$K_n \subset K$ = subconjunto de trips de K , que atendem às restrições R do navio $n \in N$.

q_{ni} = capacidade do navio n , referente a carga do trip $i \in K$.

D_i = demanda do trip $i \in K$

C_{nvi} = custo de transporte em lastro do navio n , na velocidade v , percorrer o trecho (i, j)

c_{nvi} = custo para o navio n , efetuar na velocidade v , o trip $i \in K$.

$$t_{ni}^0 = \begin{cases} \text{valor constante } e_{ni}^0 & \text{se } i \in K \\ \text{ou} \\ \text{variável contínua de tempo} & \text{se } i \in \bar{K} \end{cases}$$

onde t_{ni}^0 = tempo estimado para o navio n aportar na origem do trip i , pronto para operar.

t_{nvi} = tempo para o navio n percorrer, em lastro, o trecho entre o destino de i à origem de j , na velocidade v .

U_i = tempo limite para concluir o trip $i \in K$.

$[a_i, b_i]$ = time window do trip $i \in K$.

x_{nvi} = var. bin. zero-um =
$$\begin{cases} 1, & \text{se o navio } n \text{ percorre} \\ & (i,j) \text{ na velocidade } v \in V \\ 0, & c/c. \end{cases}$$

y_{nvi} = var. bin. =
$$\begin{cases} 1, & \text{se o navio } n \text{ efetua o trip } i, \text{ na} \\ & \text{velocidade } v \in V \\ 0, & c/c \end{cases}$$

FORMULAÇÃO MATEMÁTICA DE (F4)

- Função objetivo:

$$V(F4) = \text{Mínimo} \quad \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in K} c_{nvi} x_{nvi} + \\ + \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} y_{nvi}$$

Desejamos minimizar o custo global de efetuar os trips do atual horizonte de planejamento.

2 - Ligação entre o trecho em lastro (i, j) e o

atendimento do trip $j \in \bar{K}$.

$$\sum_{i \in \bar{K}} \sum_{v \in V} x_{nvi j} = \sum_{v \in V} y_{nvj} \leq 1, \quad \forall j \in \bar{K}, \quad \forall n \in N$$

3 - Satisfação das demandas de cada trip $i \in \bar{K}$.

$$\sum_{n \in N_i} \sum_{v \in V} q_{ni} y_{nvi} \geq D_i, \quad \forall i \in \bar{K}$$

4 - Variáveis zero-um do problema:

$$x_{nvi j} \in \{0, 1\}, \quad y_{nvj} \in \{0, 1\}, \quad \forall n \in N, \forall v \in V, \forall i \in \bar{K}, \forall j \in PK(i)$$

5 - Ligação entre dois horizontes de planejamento adjacentes

$$\sum_{v \in V} \sum_{i \in \bar{K}} \sum_{j \in \bar{K}} x_{nvi j} = 1, \quad \forall n \in N$$

As restrições 5 nos dizem que cada navio $n \in N$ deve percorrer exatamente um trecho de ligação entre dois horizontes de planejamento adjacentes.

6 - Ligação entre os problemas do roteamento e scheduling

$$t_{ni}^0 + t_{nvi} + t_{nv'ij} - t_{nj}^0 \leq (1 - \sum_{v \in V} x_{nvi j}) M_{nvv'ij},$$

onde $M_{nvv'ij} \geq b_i + t_{nvi} + t_{nv'ij} - a_j, \quad \forall n \in N, \forall v, v' \in V, \quad \forall j \in \bar{K},$

$\forall j \in PK(i)$

As desigualdades (6) são restrições clássicas nos problemas de roteamento e scheduling de veículos com restrições de *time windows* e pode ser usada em (F4) sem análise prévia dos t_{ni}^0 , devido à redefinição do conjunto de trips $K = \bar{k} \cup k$.

Existem dois casos a analisar:

i) Se j é o primeiro trip do schedule do navio $n \in N$.

Neste caso, não existe para este schedule, nenhum trip $i \in \bar{k}$ tal que $\sum_{v \in V} x_{nvi} = 1$. Isto é, não existe viagem no sentido $i \rightarrow j$, com $i, j \in \bar{k}$.

Mas nesta formulação (F4), podemos afirmar neste caso, que o navio n está concluindo ou já concluiu um trip $i \in k(n)$, logo existe um $i \in K = k \cup \bar{k}$ tal que $\sum_{v \in V} x_{nvi} = 1$, para $i \in K = \bar{k} \cup k$.

Assim, $t_{nj}^0 \geq t_{ni}^0 + t_{nvi} + t_{nv'ij}$, onde $i \in k(n)$.

ou $t_{nj}^0 \geq e_{ni}^0 + t_{nvi} + t_{nv'ij}$, onde $i \in k(n)$.

ii) Por outro lado, se $j \in \bar{k}$ não for o trip inicial do schedule do navio $n \in N$, neste caso existe um trip $i \in \bar{k}$, efetuado anteriormente a j . Também neste caso (6) é consistente.

7 - Atendimento do tempo limite para concluir um trip $j \in \bar{K}$.

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} y_{nvj}) M_{nvj}, \text{ onde}$$

$$M_{nvj} \geq b_j + t_{nvj}, \forall n \in N, \forall v \in V, \forall j \in \bar{K}$$

8 - Restrições *time windows* de cada trip $i \in \bar{K}$.

$$a_i \leq t_{ni}^0 \leq b_i, \forall n \in N, \forall i \in \bar{K}$$

Portanto o conjunto de restrições 1 a 8 descrevem o seguinte problema de programação linear mista.

$$\begin{aligned} V(F4) = \text{Mínimo} & \sum_{n \in N} \sum_{v \in V} \sum_{i \in \bar{K}} \sum_{j \in \bar{K}} c_{nvij} x_{nvij} + \\ & + \sum_{n \in N} \sum_{v \in V} \sum_{i \in \bar{K}} c_{nvi} y_{nvi} \end{aligned} \quad (4.19)$$

$$\text{S.A.} \quad \sum_{n \in N_i} \sum_{v \in V} q_{ni} y_{nvi} \geq D_i, \forall i \in \bar{K} \quad (4.20)$$

$$\sum_{i \in \bar{K}} \sum_{v \in V} x_{nvij} = \sum_{v \in V} y_{nvj} \leq 1, \forall j \in \bar{K}, \forall n \in N \quad (4.21)$$

$$\sum_{v \in V} \sum_{i \in \bar{K}} \sum_{j \in \bar{K}} x_{nvij} = 1, \forall n \in N \quad (4.22)$$

$$x_{nvij} \in (0, 1), y_{nvi} \in (0, 1), \forall n \in N, \forall v \in V, \forall i \in \bar{K}, \forall j \in \bar{K} \quad (4.23)$$

$$t_{ni}^0 + t_{nvi} + t_{nv'ij} - t_{nj}^0 \leq (1 - \sum_{v \in V} x_{nvi_j}) M_{nvv'ij}$$

onde $M_{nvv'ij} \geq b_i + t_{nvi} + t_{nv'ij} - a_j$, $\forall n \in N$,

$$\forall v, v' \in V, \forall i \in K, \forall j \in \bar{K} \quad (4.24)$$

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} y_{nvj}) N_{nvj}, \text{ onde} \quad (4.25)$$

$$N_{nvj} \geq b_j + t_{nvj}, \forall n \in N, \forall v \in V, \forall j \in \bar{K}$$

$$a_i \leq t_{ni}^0 \leq b_i, \forall n \in N, \forall i \in \bar{K} \quad (4.26)$$

CONCLUSÕES DE (F4)

Ao incorporar trips do horizonte de planejamento anterior, conseguimos utilizar uma formulação direta para geração das variáveis contínuas de tempo t_{ni}^0 sem relacionarmos os valores de e_{ni}^0 , com a_i e b_i (como sugerido em (F2), (F3)).

Mas com isso aumentamos o número de trips do problema de $|K|$ para $|K| + |N|$.

IV-5) FORMULAÇÃO 5 (F5)

A formulação (F5) é baseada na formulação (F2) com as seguintes simplificações:

i) Supomos aqui velocidade única para os navios.

ii) Cada trip deve ser atendido numa única visita e por um único navio.

No caso (ii), se existe alguma demanda $D_i > q_{ni}$ para todo navio $n \in N$, neste caso dividimos D_i em um conjunto de demandas menores $D_i(j)$, tal que $D_i(j) \leq q_{ni}$ para algum navio $n \in N$, onde q_{ni} é a capacidade do navio n referente a carga do trip $i \in K$.

Desta forma, $D_i(j)$ representará a demanda de um novo trip $i(j)$, onde $D_i = \bigcup_j D_i(j)$ e onde os custos, e tempos entre dois trips $i(j_1)$ e $i(j_2)$ serão os custos e tempos de ir da origem ao destino do trip i e finalmente custos e tempos entre um $i(j)$ e um trip original $k+i$ será dado pelo custo e tempo entre os trips k e o trip i .

Finalmente nesta formulação, as restrições de acoplamento entre variáveis de fluxo x_{ni} com as variáveis de decisão y_{nj} e variáveis de tempo t_{nj}^o são alteradas de modo a facilitar o uso de técnicas de relaxação.

NOTAÇÃO DE (F5)

N = Conjunto de navios heterogêneos da frota.

K = Conjunto ordenado de trips que compõe o atual horizonte de planejamento.

R = Conjunto das restrições locais (veja notação de (F2)).

$(N_k \subset N)$ = Subconjunto de N , que atendem às restrições R do trip $k \in K$.

$(K_n \subset K)$ = Subconjunto de K , que atendem as restrições R do navio $n \in N$.

$PK(k)$ = Subconjunto de trips de k , posteriores (no tempo) ao trip $k \in K$.

$AK(k)$ = Subconjunto de trips de K , anteriores (no tempo) ao trip $k \in K$.

q_{ni} = capacidade do navio n , referente a carga do trip $i \in K$.

D_i = Demanda do trip $i \in K$.

C_{nij} = Custo do navio $n \in N$ se deslocar em lastro o trecho entre o destino de i e a origem de j .

$C_{nii} = \infty$.

C_{ni} = custo para o navio n efetuar o trip $k \in K$.

e_{ni}^0 = Tempo inicial constante para o navio n aportar na origem do trip i pronto para operar no início do schedule do navio n .

t_{ni}^0 = Variável de tempo inicial para aportar na origem do trip i pronto para operar. Em particular se i for o trip inicial do schedule de n , então $t_{ni}^0 = e_{ni}^0$.

t_{ni} = Tempo para o navio n efetuar o trip i .

t_{nij} = Tempo para o navio n percorrer em lastro o trecho entre o destino de i à origem de j .

U_j = Tempo limite para concluir o trip j .

$[a_i, b_i]$ = *time-window* do trip i , isto é o intervalo de tempo dentro do qual o navio a ele alocado deve aportar na sua origem, pronto para operar.

$$x_{nij} = \text{Var. zero-um} = \begin{cases} 1, & \text{se o navio } n \text{ percorrer } (i,j) \\ 0, & \text{c/c} \end{cases}$$

$$y_{nj} = \text{var. zero-um} = \begin{cases} 1, & \text{se o navio } n \text{ efetua o trip } j \\ 0, & \text{c/c} \end{cases}$$

FORMULAÇÃO MATEMÁTICA (F5)

1 - Função objetivo: Minimizar os custos de efetuar os trips $i \in K$.

$$V(F5) = \text{Mínimo} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} c_{nij} x_{nij} + \sum_{n \in N} \sum_{j \in K} c_{nj} y_{nj}$$

2 - Restrições de atendimento de cada demanda numa única visita por um único navio.

$$\sum_{n \in N} \sum_{i \in K} x_{nij} = 1, \quad \forall j \in K$$

3 - Variáveis de fluxo e de decisão do problema:

$$x_{nij} \in \{0,1\}, y_{nj} \in \{0,1\}, \forall n \in N, \forall i \in K, \forall j \in PK(i)$$

4 - Ligação entre as variáveis de fluxo, decisão e de tempo:

$$t_{ni}^0 + Q_{nij} - t_{nj}^0 \leq (1 + x_{nij} - 2y_{nj}) M_{nij}, \text{ onde}$$

$$M_{nij} \geq b_i + Q_{nij} - a_j, \forall n \in N, \forall i \in K, \forall j \in PK(i)$$

Neste conjunto de restrições, achamos conveniente efetuar um esclarecimento mais detalhado da sua construção.

Pelas restrições (2) podemos afirmar que:

$$x_{nij} \leq 1, \forall n \in N, \forall i \in K, \forall j \in PK(i)$$

Assim, usando (2) e (3) teremos as seguinte possibilidades:

i) Quando $x_{nij} = 0$ e $y_{nj} = 0$, neste caso teremos:

$$t_{ni}^0 + Q_{nij} - t_{nj}^0 \leq M_{nij} \text{ (redundante)}$$

Isto é, se n não chega em j a partir de i , e n não efetua o trip j , não há necessidade de n satisfazer às restrições de tempo relacionadas ao trip j .

ii) Quando $x_{nij} = 1$ e $y_{nj} = 1$, neste caso teremos:

$t_{ni}^0 + Q_{nij} \leq t_{nj}^0$, ou seja, se o navio n chega em j a partir de i e efetua j , então se deve atender às restrições de tempo do trip $j \in K$.

iii) Quando $x_{nij} = 1$ e $y_{nj} = 0$, neste caso

$t_{ni}^0 + Q_{nij} - t_{nj}^0 \leq 2M_{nij}$ (redundante), isto é, se o navio n chega em j mas não efetua j , não existe a necessidade de n atender às restrições de tempo do trip j .

iv) Quando $x_{nij} = 0$ e $y_{nj} = 1$, aqui teremos

$$t_{ni}^0 + Q_{nij} - t_{nj}^0 \leq -M_{nij} \text{ (inviável)}$$

ou seja, proíbe-se a possibilidade do navio n efetuar o trip j , sem chegar a j .

v) Das restrições (2) temos que $0 \leq x_{nij} \leq 1$, $\forall n \in N$, $\forall i \in K$ e $\forall j \in PK(i)$. Portanto em (4) y_{nj} será impedido de tornar valor superior a 1.

5 - Atendimento dos tempos limites para concluir cada trip $j \in K$.

$$U_j \geq t_{nj}^0 + t_{nj} - (1 - y_{nj}) D_{nj}, \text{ onde } D_{nj} \geq b_j + t_{nj}, \\ \forall n \in N, \forall j \in K$$

6 - Restrições *time-windows* do trip $i \in K$.

$$a_i \leq t_{ni}^0 \leq b_i, \forall i \in K.$$

Portanto as restrições 1 a 6 descrevem o seguinte problema de programação linear mista.

$$V(F5) = \text{Min.} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} c_{nvj} x_{nij} + \sum_{n \in N} \sum_{j \in K} c_{nj} y_{nj} \quad (4.27)$$

$$\text{S.A.} \quad \sum_{n \in N} \sum_{i \in K} x_{nij} = 1, \quad \forall j \in K \quad (4.28)$$

$$x_{nij} \in (0, 1), \quad y_{nj} \in (0, 1), \quad \forall i \in K, \quad \forall n \in N, \quad \forall j \in K \quad (4.29)$$

$$t_{ni}^0 + Q_{nij} - t_{nj}^0 \leq (1 + x_{nij} - 2y_{nj}) M_{nij}, \quad \text{onde} \quad (4.30)$$

$$M_{nij} \geq b_i + Q_{nij} - a_j, \quad \forall n \in N, \quad \forall i \in K, \quad \forall j \in PK(i)$$

$$U_j \geq t_{nj}^0 + t_{nj} - (1 - y_{nj}) D_{nj}, \quad \text{onde} \quad (4.31)$$

$$D_{nj} \geq b_j + t_{nj}, \quad \forall n_j, \quad \forall n \in N, \quad \forall j \in K$$

$$a_i \leq t_{ni}^0 \leq b_i, \quad \forall n \in N, \quad \forall i \in K \quad (4.32)$$

IV-6) FORMULAÇÃO 6 (F6)

Na programação atual de navios em longo curso da (FP) (veja *BECKER(1987)*), existem alguns critérios que influenciam na escolha do tipo do navio a ser alocado para efetuar um determinado contrato (trip).

Esta influência pode ser causada por exemplo, por

fatores do tipo: distância a ser percorrida, tipo de carga, periodicidade e quantidade de carga transportada a cada etapa do contrato, fatores que são previamente conhecidos.

Desta forma, torna-se possível antecipar que tipo de navio deve ser alocado para atender a um determinado trip.

Queremos mostrar com isso, que o problema de roteamento e scheduling de navios heterogêneos, pode muitas vezes ser particionado num conjunto de subproblemas de frotas homogêneas.

Neste esquema, inicialmente particionamos a frota global de navios N , em p - sub frotas menores e homogêneas N_i , com $N = \bigcup_{i=1}^p N_i$ e $N_i \cap N_j = \emptyset$ se $i \neq j$.

De posse dos conjuntos $N_i \subset N$, particionamos igualmente o conjunto original de trips K , também em p - sub-conjuntos independentes $K_i \subset K$, com $K = \bigcup_{i=1}^p K_i$, de modo que cada N_i tenha em correspondência um conjunto $K_i \subset K$ formado pelos trips *mais adequados* aos navios $n \in N_i$.

A solução global obtida resolvendo os p - sub-problemas homogêneos poderá não mais ser a solução ótima, desde que os critérios para partição de K e N impedem o atendimento de trips $k \in K_j$ por navios $n \in N_i$, se $i \neq j$.

Uma alternativa para tentar separar este problema seria

supor que os conjuntos K_i não fossem necessariamente disjuntos uns dos outros.

Por exemplo, se existir um trip $k \in K$ que se encaixa em vários subconjuntos K_i , poderíamos pensar em manter este trip k em cada um desses subconjuntos K_i .

As implicações deste procedimento são duas:

i) A possibilidade de obter uma solução mais próxima do ótimo.

ii) A solução global contém um mesmo trip em mais de um schedule sem necessidade.

Para evitar a ocorrência deste último caso, após obter a solução global, poderíamos efetuar uma triagem no caso de ocorrer redundâncias do tipo ii).

NOTAÇÃO DA FORMULAÇÃO (F6)

x_{ij} = número de navios que percorre diretamente o arco (i,j)

$K(n)$ = o trip efetuado mais recentemente pelo navio n nos horizontes de planejamentos anteriores.

t_i^0 = variável de tempo para um navio chegar a origem do

trip i , pronto para operar.

t_i^e = tempo para um navio efetuar o trip i

$t_{i,j}$ = tempo para um navio percorrer em lastro, o trecho (i,j) .

NN_i = número de navios necessários para atender a demanda do trip i

N_r = subconjunto de navios homogêneos do tipo r .

K_r = subconjunto de trips *mais viáveis* ao tipo de navio r .

$C_{i,j}$ = custo para percorrer em lastro o arco (i,j)

$K_r^+ = K_r \cup \left(\bigcup_{n \in N_r} K(n) \right) = K_r$ somado aos trips efetuados mais recentemente pelos navios $n \in N_r$.

FORMULAÇÃO MATEMÁTICA DE (F6)

$$V(F6) = \text{Mínimo} \sum_{i \in K_n} \sum_{j \in K_r} c_{i,j} x_{i,j} \quad (4.33)$$

$$\text{S.A.} \quad \sum_{j \in K_n} x_{k(n),j} = 1, \quad \forall n \in N_r \quad (4.34)$$

$$\sum_{i \in K_r} x_{i,j} = NN_j, \quad \forall j \in K_r \quad (4.35)$$

$$\sum_{j \in K_r} x_{i,j} \leq NN_i, \quad \forall i \in K_r \quad (4.36)$$

$$\text{Se } x_{i,j} > 0 \Rightarrow t_i^0 + t_i^e + t_{i,j} \leq t_j^0, \quad \forall i \in K_r^+, \quad \forall j \in K_r \quad (4.37)$$

$$a_i \leq t_i^0 \leq b_i, \quad \forall i \in V_{K_r} \quad (4.38)$$

$$x_{ij} \geq 0 \text{ e inteiro, } \forall i \in K_r^+, \forall j \in K_r \quad (4.39)$$

ANÁLISE DAS RESTRICÇÕES DE (F6)

A função objetivo (4.33) minimiza o custo total produzido pelas viagens ociosas.

O conjunto de restrições (4.34) determina que cada navio N_r deve percorrer exatamente um arco de ligação entre dois horizontes de planejamento consecutivos.

Com isso, exige-se que cada navio efetue ao menos um trip do atual horizonte de planejamento.

(4.35) determina que o número de navios que chega em j é exatamente o número de navios necessários para atender à demanda de j .

Portanto, nesta formulação supomos que não exista nenhum navio n parado em j para atender j no início do planejamento. Caso isso ocorra na prática, tal situação é contornada com a criação de um trip artificial i origem do arco (i, j) cujo custo de transporte é $c_{ij} = 0$ e o tempo de percurso $t_{ij} = 0$.

Tal procedimento pode ser feito previamente conhecendo apenas cada trip, suas origens e destinos. Em particular adota-se este tipo de procedimento se a origem de um trip j for igual ao destino de um trip i , anterior e adjacente a j .

O conjunto de restrições (4.36) nos diz que o número de navios que parte do trip $i \in K_r$ é menor ou igual ao número de navios requisitados para atender o trip i . A desigualdade é devido à possibilidade de algum navio permanecer em i , após efetuar i .

As restrições (4.37) conectam as variáveis de fluxo $x_{i,j}$ com as variáveis contínuas de tempo t_i^o e exige que se algum navio percorrer (i,j) então os tempos iniciais para efetuar j devem ser cumpridos.

Em particular (4.37) proíbe combinar o trip i consigo mesmo.

(4.38) são as restrições do tipo *time-window* dos trips $i \in K_r$ e finalmente (4.39) são as variáveis inteiras $x_{i,j}$ do problema.

DIFICULDADES DA FORMULAÇÃO (F6)

A dificuldade da formulação (F6) está na interpretação do conjunto de restrições (4.37) já que mesmo (F6) sendo um problema com frota homogênea, a posição inicial de cada navio N_r é, na maioria dos casos, diferente um do outro.

Assim, t_i^0 possui na realidade valores distintos para cada navio $n \in N_r$, bem como os custos de chegada num determinado trip $i \in K_r$.

ANÁLISE E PROPOSTA PARA AS RESTRIÇÕES (4.37) DE (F6)

As restrições (4.37), se j é o primeiro trip de um schedule s , qual navio $n \in N_r$ deve ser a ele alocado?

Esta dúvida surge na medida que os navios de N_r mesmo sendo homogêneos, podem ser localizados em pontos distintos um do outro no início de um planejamento, acarretando com isso, custos e tempo iniciais diferentes.

Em outras palavras, cada navio $n \in N_r$ pode estar em depósitos distintos, gerando com isso um problema com frota homogênea mas de múltiplos depósitos.

A) A QUESTÃO DOS CUSTOS INICIAIS: Esta questão pode ser resolvida após a geração da solução do problema.

De fato, após conhecermos o conjunto de schedules da solução final (de cardinalidade igual a N_r), resta definirmos para cada schedule, qual navio alocar.

Um critério lógico é o de designar o navio cujo custo de chegar à origem do trip inicial do schedule seja mínimo (estes custos são previamente conhecidos) e que ainda não tenha sido alocado a nenhum schedule.

B) A QUESTÃO DOS TEMPOS INICIAIS:

O procedimento imediato é tomar

$$t_j^0 = e_j^0 = \max_{k \in K_r} (e_k^0), \text{ neste caso, qualquer navio de } N_r$$

estará apto a se candidatar a efetuar o trip j .

Outra alternativa seria a de tentar reorganizar os conjuntos N_r , particionando se necessário estes conjuntos em subconjuntos menores M_r tal que $N_r = \cup_r M_r$.

Porém, para não pulverizar N , conservamos para cada M_r o conjunto de trips K_r correspondente a N_r .

Com isso, a solução final de (F6) conterá, possivelmente, schedules efetuando um trip mais de uma vez ou vários schedules solução contendo um mesmo trip.

Mas isso pode ser em parte superado, resolvendo um problema do tipo set-covering (como já visto em (F1)), onde as colunas da matriz set-covering são os schedules solução de (F6).

CONCLUSÕES DE (F6)

O aspecto fundamental da formulação (F6) é a *redução drástica* nas dimensões do problema a ser solucionado tornando viável neste caso o uso de técnicas exatas para a sua resolução.

Mostramos a seguir, um exemplo com $|K|=50$ trips, $|N|=30$ navios, $|N_r|=5$ navios, $|K_r|=10$ trips.

FORMULAÇÃO	NÚMERO DE VARIÁVEIS ORIGINAIS	NÚMERO DE RESTRIÇÕES	NÚMERO VARIÁVEIS INCLUINDO AS DE FOLGA
(F1)*	$ S $	80	$ S + 80$
(F2)	154.500	379.500	532.550
(F3)	3.000	154.600	157.600
(F4)	247.200	254.510	499.310
(F5)	78.000	79.600	157.550
(F6)	110	195	295

(*) (F1) em relação a segunda etapa.

DIMENSÕES DOS PROBLEMAS (F1) A (F6)

TABELA 4.2

IV-7) FORMULAÇÃO 7 (F7)

Esta formulação trabalha somente com as variáveis de fluxo e de tempo.

São efetuadas algumas simplificações em relação a formulações mais abrangentes como (F2).

Supomos que a velocidade seja única para os navios e que cada trip do atual horizonte de planejamento possa ser atendido por um único navio numa única visita.

NOTAÇÃO DE (F7)

N = conjunto de navios heterogêneos da frota

K = conjunto de trips do atual horizonte de planejamento

KAP = subconjunto de trips do planejamento anterior, formado pelos trips mais recentemente efetuados por cada navio

$$\bar{K} = (K) \cup (KAP)$$

N_i = conjunto de navios $n \in N$, que atendem às restrições locais R (veja notação de (F2)) $n \in N_i$, significa também que n consegue atender à demanda de i numa única visita

c_{nij} = custo operacional do navio $n \in N$ efetuar o trecho (i, j) em lastro mais o custo do navio n efetuar o trip $j \in K$, com $i \in \bar{K}$

c_{njt} = custo constante igual para todo $n \in N$, $\forall j \in K$, onde t é destino final de cada schedule

U_j - tempo limite para efetuar o trip j

t_{ni} = tempo para o navio n efetuar o trip i

t_{nij} = tempo para o navio n percorrer em lastro o trecho (i, j)

t_{njt} = tempo constante igual para todo $n \in N$, $\forall j \in K$

t_{ni}^o = variável contínua de tempo, necessário para o navio $n \in N$ aportar na origem do trip $i \in K$, pronto para operar

x_{nij} = var. binária zero-um $\begin{cases} 1, & \text{se } n \text{ percorre } (i, j) \\ 0, & \text{c/c} \end{cases}$

$[a_i, b_i]$ - *time-window* do trip i , intervalo dentro do qual o navio alocado a este trip deve aportar na sua origem pronto para operar.

FORMULAÇÃO MATEMÁTICA DE (F7)

$$V(F7) = \text{Mínimo} \sum_{n \in N} \sum_{i \in \bar{K}} \sum_{j \in K} c_{nij} x_{nij} \quad (4.40)$$

$$\text{S.A.} \sum_{i \in \bar{K}} x_{nij} = 1, \forall j \in \bar{K}, \forall n \in N \quad (4.41)$$

$$\sum_{i \in \bar{K}} x_{nji} = 1, \forall j \in \bar{K}, \forall n \in N \quad (4.42)$$

$$x_{nij} \geq 0, \forall n \in N, \forall i \in \bar{K}, \forall j \in \bar{K} \quad (4.43)$$

$$t_{ni}^0 + t_{ni} + t_{nij} - t_{nj}^0 \leq (1 - x_{nij}) M_{nij}, \text{ onde} \quad (4.44)$$

$$M_{nij} \geq b_i + t_{ni} + t_{nij} - a_j, \forall n \in N, \forall i \in \bar{K}, \forall j \in \bar{K}$$

$$U_j \geq t_{nj}^0 + t_{nj} (1 - \sum_{i \in \bar{K}} x_{nij}) N_{nj}, \text{ onde} \quad (4.45)$$

$$N_{nj} \geq b_j + t_{nj}, \forall n \in N, \forall j \in \bar{K}$$

$$a_i \leq t_{ni}^0 \leq b_i, \forall n \in N, \forall i \in \bar{K} \quad (4.46)$$

$$x_{nij} \in \{0, 1\}, \forall n \in N, \forall i \in \bar{K}, \forall j \in \bar{K} \quad (4.47)$$

ANÁLISE DAS RESTRICÇÕES DE (F7)

As restrições (4.40) a (4.43) descrevem um problema de roteamento puro conhecido na literatura como o N-problema de fluxo de custo mínimo.

Os conjuntos de restrições (4.44) a (4.47) formam um problema de scheduling com restrições *time-window*.

CAPÍTULO V

GERAÇÃO DE LIMITES INFERIORES:

NOVAS RELAXAÇÕES E DECOMPOSIÇÕES LAGRANGEANAS

INTRODUÇÃO

Apresentamos aqui algumas técnicas para obter limites inferiores do problema (FP) utilizando relaxações e decomposições lagrangeanas.

A geração dos limites inferiores de cada relaxação equivale a resolver um problema do tipo *máx/min* usando por exemplo métodos do tipo subgradiente.

São propostas várias relaxações para algumas das formulações desenvolvidas no capítulo III analisando sempre que possível sua performance quanto à qualidade do limite gerado, complexidade do problema relaxado e a possibilidade de obter soluções viáveis ou *quase-viáveis* para o problema original.

5.1 - ANÁLISE DA FORMULAÇÃO 1 (F1):

Considere a formulação (F1) descrita no capítulo III.

$$V(F1) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (5.1)$$

$$\text{S.A.} \quad \sum_{s \in S} e_{ks} x_s \geq 1, \forall k \in K \quad (5.2)$$

$$\sum_{s \in S(n)} x_s \leq 1, \forall n \in N \quad (5.3)$$

$$x_s \in \{0, 1\}, \forall s \in S \quad (5.4)$$

As relaxações da formulação F1 são efetuadas em duas etapas:

Na primeira, é introduzida a técnica de *relaxação lagrangeana convencional* introduzida por A. GEOFFRION (1974) e na etapa posterior trabalhamos com a relaxação conhecida como *Decomposição Lagrangeana* (veja GUINARD e KIM (1987 a, 1987b)).

TÉCNICAS DE RELAXAÇÃO LAGRANGEANA CONVENCIONAL

APLICADO AO PROBLEMA (FP).

Usando relaxações Lagrangeanas convencionais, podemos formular através de F1, dois problemas relaxados, um relaxando as restrições (5.2) e noutro as relaxações (5.3), com multiplicadores lagrangeanos $\mu_k \geq 0, \forall k \in K$ e $\omega_n \geq 0, \forall n \in N$ respectivamente.

R1) RELAXAÇÃO LAGRANGEANA CONVENCIONAL 1:

Dualizando as restrições de recobrimento (5.2) da formulação F1 com multiplicadores $\mu_k \geq 0$, $\forall k \in K$, teremos o seguinte problema relaxado:

$$V(R1_{\mu}) = \text{Mínimo} \sum_{s \in S} (c_s - \sum_{k \in K} e_{ks} \mu_k) x_s + \sum_{k \in K} \mu_k \quad (5.5)$$

$$\text{S.A.} \quad \sum_{s \in S(n)} x_s \leq 1, \quad \forall n \in N \quad (5.6)$$

$$x_s \in \{0, 1\}, \quad \forall s \in S \quad (5.7)$$

Cada problema $(R1_{\mu})$ com o vetor multiplicador $\mu = (\mu_k)$ fixado, onde $\mu_k \geq 0$, $\forall k \in K$ pode ser resolvido por uma simples inspeção nos coeficientes $\hat{c}_s = c_s - \sum_{k \in K} e_{ks} \mu_k$ da função objetivo de $(R1_{\mu})$.

Para isso, basta escolher para cada navio $n \in N$ os valores:

$$\hat{c}_{s(n)} = \text{mínimo} (c_s - \sum_{k \in K} e_{ks} \mu_k, \quad s \in S(n)) \text{ e}$$

alocar os seguintes valores para cada variável $x_{s(n)}$, $s \in S(n)$.

- Se $\hat{c}_{s(n)} < 0$ faça

$$x_{s(n)} = 1 \text{ e } x_k = 0, \quad \forall k \in S(n) \text{ com } s \neq k$$

- Se $\hat{c}_s(n) \neq 0$ faça

$$x_s = 0, \forall s \in S(n)$$

INTERPRETAÇÃO DOS COEFICIENTES DE CUSTO DE (R1 μ)

No problema (FP), podemos chamar:

c_s = custo direto de efetuar o schedule $s \in S(n)$ pelo navio $n \in N$.

$\sum_{k \in K} e_{ks} \mu_k$ = custo alternativo de efetuar os trips k pertencentes ao schedule $s \in S(n)$, através de outros schedules que o contenha.

Assim, se para um determinado schedule $s \in S(n)$ ocorrer:

$$1) \quad c_s \geq \sum_{k \in K} e_{ks} \mu_k \rightarrow \text{n\~{a}o vale a pena alocar o navio } n \in N$$

para efetuar os trips pertencentes ao schedule $s \in S(n)$. Isto é, é preferível (mais barato) que os trips k do schedule $s \in S(n)$ sejam efetuados por outros navios (um ou mais) cujos schedules os contenham.

Isto significa que a rota direta é "cara" ou que a rota alternativa é "barata" e neste caso designamos os valores:

$x_s = 0$, (o navio n não efetua o schedule $s \in S(n)$, mas não fica necessariamente ocioso).

ii) $c_s < \sum_{k \in K} e_{ks} \mu_k \Rightarrow$ vale a pena efetuar os trips k pertencentes a s , via schedule s .

Ou seja, é vantajoso efetuar os trips $k \in s \in S(n)$ pelo navio n .

Neste caso fazemos:

$$x_s = 1 \text{ e } x_k = 0 \quad \forall k \in S(n) \text{ com } k \neq s$$

Naturalmente cada valor de μ fornece um limite inferior $V(R1\mu)$ de $V(F1)$.

O melhor desses limites inferiores de $V(F1)$ nesta relaxação é dado por:

$$R1 \quad \left\{ V(R1) = \underset{\mu \geq 0}{\text{Máximo}} \left\{ V(R1\mu) \right\} \right. \quad (5.8)$$

RII RELAXAÇÃO LAGRANGEANA CONVENCIONAL 2:

Ao dualizar as restrições (5.3) de (F1) com multiplicadores $w = (w_n)$, com $w_n \geq 0$, $\forall n \in N$ geramos o seguinte problema relaxado:

$$V(R2w) = \underset{\text{Mínimo}}{\sum_{n \in N} \sum_{s \in S(n)} (c_s + w_n) x_s} - \sum_{n \in N} w_n \quad (5.9)$$

$$\text{S.A.} \quad \sum_{s \in S} e_{ks} x_s \geq 1, \quad \forall k \in K \quad (5.10)$$

$$x_s \in \{0, 1\}, \forall s \in S \quad (5.11)$$

$$\text{onde } e_{ks} \in \{0, 1\}, \forall k \in K, \forall s \in S$$

O problema primal (R2W) com o vetor $W \geq 0$ fixado é um problema clássico de programação linear inteira conhecido como o problema de recobrimento ("set covering problem") ou (SCP).

O (SCP) embora sendo de complexidade NP - completo, possui algoritmos com bons desempenhos práticos como os desenvolvidos por *Balas e Ho (1980)*.

No problema (R2W), a melhor das cotas inferiores $V(R2w)$ com $w \geq 0$ será obtido resolvendo o seguinte problema do tipo *maxmin*:

$$R2 \quad \left\{ V(R2) = \underset{w \geq 0}{\text{máximo}} \{ V(R2w) \} \right\}$$

Ambos os problemas (R1) e (R2) podem ser resolvidos usando, por exemplo, os métodos do tipo subgradiente que se caracteriza pela sua simplicidade e eficiência neste tipo de problemas.

Após desenvolver as duas relaxações lagrangeanas convencionais (R1) e (R2), já estamos em condições de efetuar as primeiras comparações dos limites inferiores obtidos até o momento para $V(F1)$.

Sabemos pela teoria da relaxação lagrangeana que os limites inferiores $V(R1)$ e $V(R2)$ se situam entre o limite inferior obtido pela relaxação linear de $(F1)$ denotado $(F1L)$ e o valor ótimo de $(F1)$, isto é:

$$\begin{aligned} V(F1L) &\leq V(R1) \leq V(F1) \\ V(F1L) &\leq V(R2) \leq V(F1) \end{aligned} \quad (5.13)$$

Além disso, como o problema $(R1\mu)$ possui a propriedade de integralidade, podemos concluir que:

$$\begin{aligned} V(R1) &= V(F1L) \\ V(F1L) &= V(R1) \leq V(R2) \leq V(F1) \end{aligned} \quad (5.14)$$

TÉCNICAS DE DECOMPOSIÇÃO LAGRANGEANA APLICADAS AO

PROBLEMA $(F1)$

Dentro da classe de relaxações conhecidas como decomposições lagrangeanas, propomos vários esquemas que melhor se adaptam ao problema $(F1)$.

DI) DECOMPOSIÇÃO LAGRANGEANA 1:

Esta é a versão clássica de decomposição lagrangeana

onde o problema original (F1) é reescrito na seguinte forma equivalente ($\overline{F1}$):

$$V(\overline{F1}) = V(F1) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (5.15)$$

$$\text{S.A.} \sum_{s \in S} e_{ks} x_s \geq 1, \quad \forall k \in K \quad (5.16)$$

$$\sum_{s \in S(n)} y_s \leq 1, \quad \forall n \in N \quad (5.17)$$

$$x_s = y_s, \quad \forall s \in S \quad (5.18)$$

$$x_s \in (0, 1), y_s \in (0, 1), \quad \forall s \in S \quad (5.19)$$

e onde $e_{ks} \in (0, 1) \quad \forall k \in K, \forall s \in S$

As variáveis y_s introduzidas em $F1$ são conhecidas como cópias das variáveis x_s .

Dualizando as restrições de acoplamento (5.18) de ($\overline{F1}$) com multiplicadores irrestritos α_s , $\forall s \in S$, obtemos o seguinte problema relaxado:

$$V(D1\alpha) = \text{Mínimo} \sum_{s \in S} (c_s + \alpha_s) x_s - \sum_{s \in S} \alpha_s y_s \quad (5.20)$$

$$\text{S.A.} \sum_{s \in S} e_{ks} x_s \geq 1, \quad \forall k \in K \quad (5.21)$$

$$\sum_{s \in S(n)} y_s \leq 1, \quad \forall n \in N \quad (5.22)$$

$$x_s \in \{0, 1\}, \forall s \in S \quad \text{e} \quad y_s \in \{0, 1\}, \forall s \in S \quad (5.23)$$

A propriedade fundamental das técnicas de Decomposição lagrangeanas pode ser notada no problema relaxado $(D1\alpha)$. De fato, $(D1\alpha)$ pode ser decomposto em dois subproblemas menores e independentes, um com as variáveis x_s e outro com y_s , como descritos a seguir:

$$V(D1X\alpha) = \text{Mínimo} \sum_{s \in S} (c_s + \alpha_s) x_s \quad (5.24)$$

$$\text{S.A.} \sum_{s \in S} a_{ks} x_s \geq 1, \quad \forall k \in K \quad (5.25)$$

$$x_s \in \{0, 1\}, \quad \forall s \in S \quad (5.26)$$

e

$$V(D1Y\alpha) = \text{Mínimo} - \sum_{s \in S} \alpha_s y_s = \text{Máximo} \sum_{s \in S} \alpha_s y_s \quad (5.27)$$

$$\text{S.A.} \sum_{s \in S(n)} y_s \leq 1, \quad \forall n \in N \quad (5.28)$$

$$y_s \in \{0, 1\}, \quad \forall s \in S \quad (5.29)$$

Assim, para cada vetor multiplicador $\alpha = (\alpha_s)$ onde α_s é irrestrito em sinal para todo $s \in S$, o valor $V(D1\alpha)$ será dado pela soma dos valores de $V(D1X\alpha)$ com $V(D1Y\alpha)$.

$$V(D1\alpha) = V(D1X\alpha) + V(D1Y\alpha), \quad \forall \alpha \quad (5.30)$$

E o melhor limite inferior de $V(F1)$ nesta decomposição é dado por:

$$D1 \quad \left\{ V(D1) = \underset{\alpha}{\text{Máximo}} \left\{ V(D1\alpha) \right\} \right\} \quad (5.31)$$

Comparativamente aos problemas anteriores $(R1_{\mu})$ e $(R2w)$ obtidos por relaxações lagrangeanas convencionais, podemos observar que:

- O problema $(D1X\alpha)$ é um problema de recobrimento (SCP) como $(R2w)$, enquanto que $(D1Y\alpha)$ é um problema que pode ser resolvido por uma simples inspeção nos coeficientes da sua função objetivo como em $(R1\mu)$.

- O limite obtido por $(D1)$ está entre o melhor limite obtido via relaxações lagrangeanas convencionais e o valor ótimo de $(F1)$ (veja *M. Guinard, e Kim (1987a) e (1987b)*), isto é :

$$\text{Máximo } \{V(R1), V(R2)\} \leq V(D1) \leq V(F1) \quad (5.32)$$

- Em particular, para o problema $(F1)$ usando a propriedade $(A14)$ podemos afirmar que:

$$\boxed{V(F1L) = V(R1) \leq V(R2) = V(D1) \leq V(F1)} \quad (5.33)$$

CONCLUSÕES DE (D1)

A técnica de decomposição lagrangeana clássica aplicada ao problema (F1) gera um problema relaxado de decomposição (D1) que fornece um limite inferior igual ao melhor limite obtido via relaxação lagrangeana convencional (veja *GEOFFRION (1974)*).

A igualdade e não a superioridade decorre do fato de um dos subproblemas de $(D1\alpha)$ possuir a propriedade de integralidade (veja *M. GUINARD e KIM (1987a) e (1987b)*).

Mas mesmo assim, esta técnica mostra vantagens como a de conservar todas as restrições do problema original (F1) em um dos subproblemas $(D1X\alpha)$ ou $(D1Y\alpha)$ ao contrário das técnicas de relaxação lagrangeana convencionais onde ao menos uma restrição original é suprimida do problema relaxado.

Com isso, a possibilidade de obtermos uma solução viável para (F1) é maior via decomposição lagrangeana justificando com isso o seu uso nestas situações.

Por outro lado, ao gerarmos uma cópia para cada variável do problema original, e acrescentando as restrições de acoplamento do tipo (5.18), criamos um igual número de multiplicadores para o problema dual lagrangeano, o que pode dificultar nos problemas de grande porte, o uso de métodos

do tipo subgradiente.

O bom senso recomenda nestas situações atualizar o procedimento de decomposição lagrangeana procurando reduzir o número de multiplicadores no problema dual lagrangeano como propomos a seguir:

DI) DECOMPOSIÇÃO LAGRANGEANA 2:

Uma proposta para reduzir o número de multiplicadores do problema (Di) é a de gerar cópias y_s de x_s para cada $s \in S$ mas exigindo que a igualdade $x_s = y_s$ seja explicitado somente para $s \in S_1 \subset S$.

Nos problemas reais, os critérios que definem se um schedule s pertence ou não ao subconjunto S_1 são inúmeros, podendo se basear por exemplo nos valores dos coeficientes de x_s na função objetivo ou então nas informações prévias do problema, dizendo que determinado schedule s tem maiores possibilidades de estar na solução final do problema.

No problema (FP) (veja *BECKER (1987)*) podemos citar os seguintes candidatos de $s \in S_1$:

i) $x_s = y_s$, onde s é o único schedule de S que contém um determinado trip $k \in K$.

ii) $x_s = y_s$, onde s é o único schedule de $S(n)$ para algum navio $n \in N$. A justificativa neste caso é que na prática, com as disponibilidades atuais (veja BECKER (1987)), as melhores soluções viáveis necessitam alocar todos os navios disponíveis da frota e mais alguns afretados no mercado.

iii) $x_s = y_s$, onde o custo relativo de efetuar o schedule s seja um dos $|S_1|$ - menores custos do problema.

Assim, baseado nas propostas i), ii), iii) podemos então reescrever o problema (F1) como:

$$V(D1) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (5.35)$$

$$\text{S.A.} \quad \sum_{s \in S} a_{ks} x_s \geq 1, \quad \forall k \in K \quad (5.36)$$

$$\sum_{s \in S(n)} y_s \leq 1, \quad \forall n \in N \quad (5.37)$$

$$x_s = y_s, \quad \forall s \in S_1 \subseteq S \quad (5.38)$$

$$x_s \in (0, 1), y_s \in (0, 1), \quad \forall s \in S \quad (5.39)$$

Efetuando um procedimento análogo ao do caso anterior (D1) podemos gerar um problema do tipo maxmin (D2) com $V(D2) = \text{máx} \{V(D2/\beta)\}$ e usando o fato de (D2) ser relaxação de (D1) podemos concluir que:

$$V(\text{FIL}) \leq V(\text{D2}) \leq V(\text{D1}) \leq V(\text{F1})$$

(5.40)

Uma generalização dos casos (D1) e (D2) é dado a seguir:

DIII) DECOMPOSIÇÃO LAGRANGEANA 3:

REINOSO e MACULAN (1988), apresentaram uma proposta para reduzir o número de multiplicadores do problema dual lagrangeano para um problema geral de Programação linear inteira.

Adaptando esta proposta ao nosso problema (F1) teremos o seguinte resultado.

Considere SA e SB subconjuntos disjuntos de S com $S = SA \cup SB$, onde S é o conjunto de todos os schedules viáveis do problema (FF).

$$V(G2) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (5.41)$$

$$\text{S.A.} \quad \sum_{s \in SA} e_{k_s} y_s + \sum_{s \in SB} e_{k_s} y_s \geq 1, \quad \forall k \in K \quad (5.42)$$

$$y_s = x_s, \quad \forall s \in SA \quad (5.43)$$

$$\sum_{s \in SB} e_{k_s} y_s = \sum_{s \in SB} e_{k_s} x_s, \quad \forall k \in K \quad (5.44)$$

$$\sum_{s \in S(n)} x_s \leq 1, \quad \forall n \in N \quad (5.45)$$

$$x_s \in \{0, 1\}, y_s \in \{0, 1\}, \quad \forall s \in S \quad (5.46)$$

e onde $e_{ks} \in (0, 1)$, $\forall k \in K$, $\forall s \in S$

No problema (G2), relaxando as restrições de acoplamento (5.43) e (5.44) obtemos $|SA| + |K|$ multiplicadores para o problema dual lagrangeano.

No caso particular onde $SA = \emptyset$ ou $S = SB$, teríamos a seguinte configuração:

$$V(G3) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (5.47)$$

$$\text{S.A.} \quad \sum_{s \in S} e_{ks} y_s \geq 1, \quad \forall k \in K \quad (5.48)$$

$$\sum_{s \in S} e_{ks} y_s = \sum_{s \in S} e_{ks} x_s, \quad \forall k \in K \quad (5.49)$$

$$\sum_{s \in S(n)} x_s \leq 1, \quad \forall n \in N \quad (5.50)$$

$$x_s \in (0, 1), y_s \in (0, 1), \quad \forall s \in S \quad (5.51)$$

Note que em (G3), o número de multiplicadores duais é de somente $|K|$ = número de trips do problema original que certamente é bem menor que $|S|$ = número de schedules viáveis do problema (F1).

Além disso, o mesmo procedimento aplicado a (G3) poderia ser feito copiando agora as restrições (5.45) ao invés de (5.42). Neste caso, estaríamos gerando o seguinte

problema relaxado:

Considere $N = N1 \cup N2$, com $N1 \cap N2 = \emptyset$.

$$V(04) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (5.52)$$

$$\text{S.A.} \quad \sum_{s \in S} e_{ks} x_s \geq 1, \quad \forall k \in K \quad (5.53)$$

$$\sum_{s \in S(n1)} y_s + \sum_{s \in S(n2)} y_s \leq 1, \quad \forall n1 \in N1 \text{ e } \forall n2 \in N2 \quad (5.54)$$

$$y_s = x_s, \quad \forall s \in S(n1) \text{ e } \forall n1 \in N1 \quad (5.55)$$

$$\sum_{s \in S(n2)} x_s = \sum_{s \in S(n2)} y_s, \quad \forall n2 \in N2 \quad (5.56)$$

$$x_s \in (0, 1), \quad y_s \in (0, 1), \quad \forall s \in S \quad (5.57)$$

No caso particular onde $N1 = \emptyset$ ou $N = N2$ temos a seguinte configuração para (04):

$$V(05) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (5.58)$$

$$\text{S.A.} \quad \sum_{s \in S} e_{ks} x_s \geq 1, \quad \forall k \in K \quad (5.59)$$

$$\sum_{s \in S(n)} y_s \geq 1, \quad \forall n \in N \quad (5.60)$$

$$\sum_{s \in S(n)} x_s = \sum_{s \in S(n)} y_s, \quad \forall n \in N \quad (5.61)$$

$$x_s \in \{0, 1\}, y_s \in \{0, 1\}, \forall s \in S \quad (5.62)$$

Dualizando as restrições de acoplamento (5.61) obtemos $|N|$ multiplicadores lagrangeanos que é exatamente o número de navios da frota que obviamente é bem menor ao número de schedules viáveis do problema original dado por $|S|$.

Nos problemas (D3) e (D5), ao relaxarmos as suas respectivas restrições de acoplamento, obtemos os seguintes problemas relaxados: denotados por (D3 γ) e (D5 η) respectivamente com γ e η irrestritos em sinal.

Os melhores limites inferiores de cada relaxação são dados por:

$$(D3) \quad \left\{ V(D3) = \underset{\gamma}{\text{máximo}} \{ V(D3\gamma) \} \right\} \quad (5.63)$$

e

$$(D5) \quad \left\{ V(D5) = \underset{\eta}{\text{máximo}} \{ V(D5\eta) \} \right\} \quad (5.64)$$

onde γ e η são multiplicadores irrestritos em sinal com cardinalidade $|K|$ e $|N|$, respectivamente.

Além disso, se (R2) é a relaxação lagrangeana convencional dualizando as restrições $\sum_{s \in S(n)} x_s \leq 1, \forall n \in N$ então valem os seguintes resultados:

LEMA 5.1. $V(D5) = V(D1)$

Prova: É fácil verificar que valem as desigualdades

$$V(R2) \leq V(D5) \leq V(D1) \leq V(F1)$$

Além disso já vimos anteriormente em (5.34) que $V(D1) = \text{máximo} (V(R1), V(R2)) = V(R2)$

Portanto

$$V(D1) = V(R2) \Rightarrow V(D1) \leq V(D5) \leq V(D1) \Rightarrow V(D1) = V(D5).$$

LEMA 5.2. No problema (F1) se uma das relaxações lagrangeanas convencionais possuir a propriedade de integralidade, então a seguinte desigualdade é satisfeita.

$V(D3) \leq V(D5)$, se (R1) possui a propriedade de integralidade
ou

$V(D5) \leq V(D3)$, se (R2) possui a propriedade de integralidade

Prova: No problema (F1) a relaxação (R1) possui a propriedade de integralidade e pelo Lema 5.1., temos que $V(D5) = V(D1) = V(R2)$.

Além disso, (D3 γ) é uma relaxação de (D1 α)

Logo $V(D3) \leq V(D1)$

Portanto $V(D3) \leq V(D1) = V(D5)$ ou

$$V(D3) \leq V(D5)$$

COROLÁRIO 5.1: Duas decomposições lagrangeanas do tipo definido DIII) geradas de um mesmo problema original são comparáveis se uma delas possui a propriedade de integralidade.

Finalmente, note que os problemas sem relaxação (G1), (G2), (G3), (G4), (G5) e (F1) são equivalentes no sentido de possuírem o mesmo valor ótimo, isto é, $V(G1) = V(G2) = V(G3) = V(G4) = V(G5) = V(F1)$.

Para concluir a análise da formulação (F1), mostramos a seguir, comparações intuitivas entre os problemas (F1), (D1), (D3), (R1), (R2) e (FIL) relativamente ao *tradeoff* entre:

- i) *solução viável VS valor ótimo.*
- ii) *solução viável VS complexidade do Problema Primal*
- iii) *solução viável VS complexidade do Problema Dual*

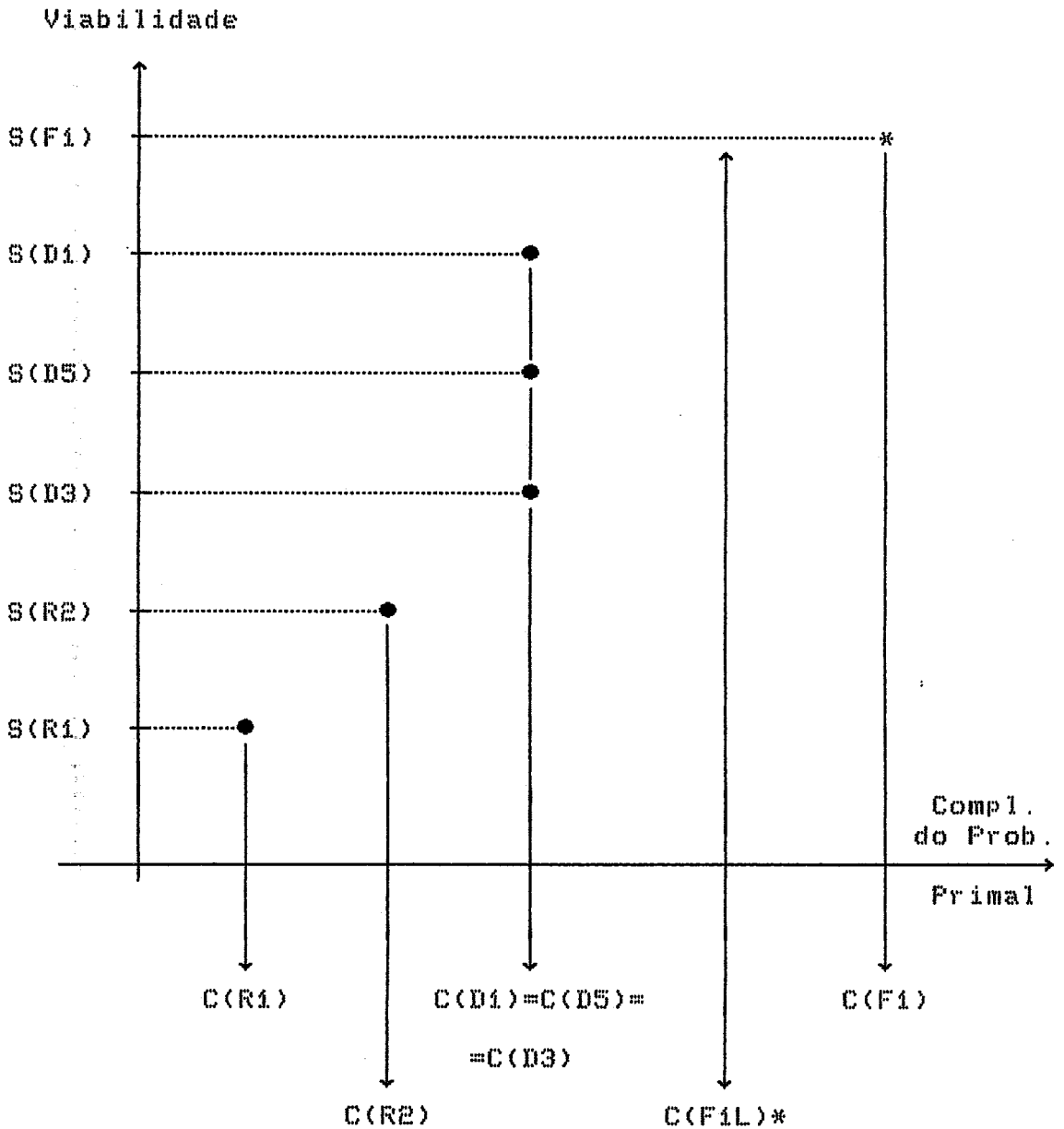
$-c(x) < c(y) \Rightarrow$ complexidade de x é inferior à de y .

$-s(x) < c(y) \Rightarrow$ viabilidade de x é inferior à de y .

$-v(x) < v(y) \Rightarrow$ valor ótimo de x é inferior à de y .

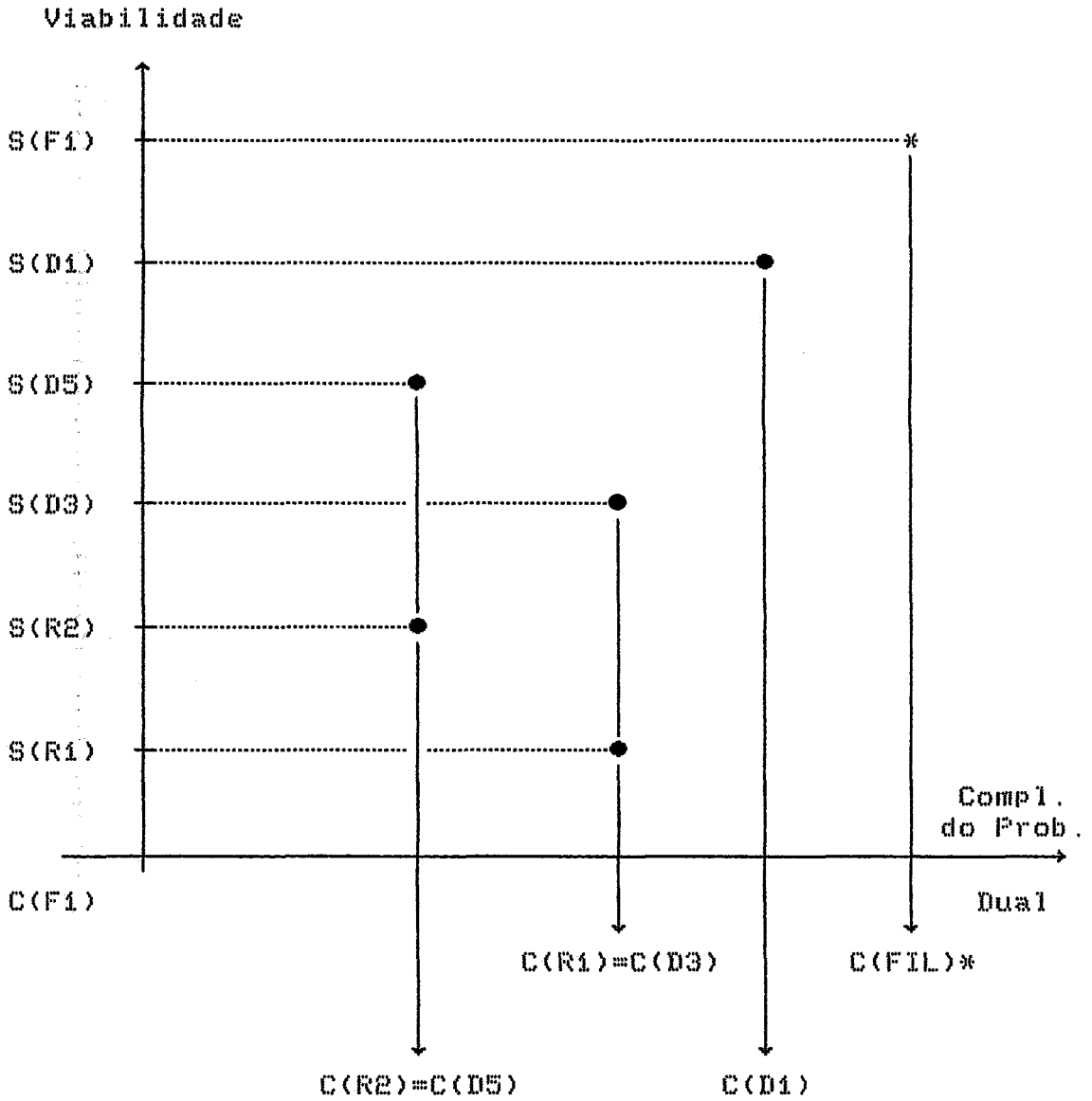
TABELA 5.2:

TRADEOFF ENTRE VIABILIDADE E COMPLEXIDADE DO PROBLEMA PRIMAL



(*) $S(FIL) \leq S(F1)$, mas $S(FIL)$ não pode ser previamente comparada com $S(D1)$, $S(D3)$, $S(D5)$, $S(R1)$ e $S(R2)$.

TABELA 5.3.
 TRADEOFF ENTRE VIABILIDADE DO PROBLEMA PRIMAL E
 COMPLEXIDADE DO PROBLEMA DUAL.



(*) $S(FIL) \leq S(F1)$, mas $S(F1)$ não é comparável previamente com os valores de $S(D1)$, $S(D3)$, $S(D5)$, $S(R1)$, $S(R2)$.

CONCLUSÕES FINAIS DE (F1)

Os gráficos das tabelas 5.1, 5.2, 5.3 nos mostram que:

A técnica de decomposição lagrangeana proposta por *MONIQUE GUINARD e KIM (1987a) e (1987b)* torna possível resolver problemas de programação linear com um elevado número de restrições, acenando ainda com possíveis soluções viáveis para o problema original.

O aspecto negativo da versão clássica desta decomposição lagrangeana em alguns casos é a presença de um elevado número de multiplicadores no problema dual lagrangeano.

Em termos do limite inferior gerado, ele é superior às relaxações lagrangeanas convencionais e a outras decomposições do tipo (D3) ou (D5).

As versões propostas por *REINOSO e MACULAN (1988)* (Problemas (D3) e (D5)) possibilitam resolver problemas de programação linear com elevado número de restrições e variáveis.

A possibilidade de gerar soluções viáveis para o problema original é menor que na versão clássica de *M. GUINARD e KIM (1987a) e (1987b)*, mas é maior que nas

técnicas de relaxações lagrangeanas convencionais. Também em termos do limite gerado, ele se situa entre aos obtidos via decomposições clássicas de *GUINARD e KIM (1987)* e as obtidas nas relaxações lagrangeanas convencionais.

Os resultados dos Lemas 5.1. e 5.2. possibilitam avaliar a priori em alguns casos, a melhor decomposição lagrangeana dentre as decomposições possíveis de um determinado problema.

5.2: ANÁLISE DA FORMULAÇÃO 2 (F2):

A formulação (F2) apresentada no capítulo III foi desenvolvida sem maiores preocupações em simplificar a representação do modelo real.

De fato, a única simplificação evidente no modelo é a forma de gerar as variáveis contínuas de tempo onde supomos velocidades constantes para cada tipo de operação (viagem em lastro, efetuar um trip etc).

Esta aproximação com a realidade gerou um modelo de maior complexidade, dificultando a princípio o uso de técnicas exatas para a sua solução.

Ainda assim, a análise teórica dessas técnicas permaneceu válida neste problema como procuramos mostrar a seguir.

Considere a formulação (F2) como desenvolvida no capítulo III

$$\begin{aligned}
 V(F2) = \text{Mínimo} & \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in PK(i)} c_{nvi} x_{nvi} + \\
 & + \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} y_{nvi}
 \end{aligned} \tag{5.65}$$

$$\text{S.A.} \quad \sum_{n \in N_i} \sum_{v \in V} q_{nvi} y_{nvi} \geq d_i, \quad \forall i \in K \quad (5.66)$$

$$\sum_{i \in AK(j)} \sum_{v \in V} x_{nvi} = \sum_{v \in V} y_{nvj} \leq 1, \quad \forall j \in K, \quad \forall n \in N \quad (5.67)$$

$$x_{nvi} \in (0, 1), \quad y_{nvj} \in (0, 1), \quad \forall n \in N, \quad \forall i \in K, \quad \forall j \in K, \quad \forall v \in V \quad (5.68)$$

$$t_{ni}^0 + t_{nvi} y_{nvi} + t_{nv'ij} - t_{nj}^0 \leq (2 - \sum_{v \in V} (y_{nvj} + x_{nvi})) M_{nij}$$

(5.69)

onde

$$M_{nij} \geq b_i + t_{nvi} + t_{nv'ij} - a_j, \quad \forall n \in N, \quad \forall i \in K, \quad \forall j \in K, \quad \forall v \in V, \quad \forall v' \in V$$

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} y_{nvj}) N_{nvj}, \quad (5.70)$$

onde

$$N_{nvj} \geq b_j + t_{nvj}, \quad \forall j \in K, \quad \forall n \in N, \quad \forall v \in V$$

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall n \in N, \quad \forall j \in K \quad (5.71)$$

TÉCNICA DE RELAXAÇÃO LAGRANGEANA CONVENCIONAL APLICADA AO PROBLEMA (F2)

RIII) RELAXAÇÃO LAGRANGEANA CONVENCIONAL 3:

Analisamos aqui, o comportamento de (F2) ao relaxarmos as restrições (5.69) e parte de (5.67) (as igualdades) gerando multiplicadores duais $\alpha_{nvv'ij} \geq 0$ e β_{nj}

irrestritos em sinal, $\forall n \in N, \forall v, v' \in V, \forall i \in K, \forall j \in K$.

O problema relaxado é descrito como:

$$\begin{aligned}
 V(R3\gamma) = \text{Mínimo} & \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in PK(i)} c_{nvij} x_{nvij} + \\
 + \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nv} y_{nvi} & + \sum_{n \in N} \sum_{v \in V} \sum_{v' \in V} \sum_{i \in K} \sum_{j \in PK(i)} \alpha_{nvv'ij} \\
 (t_{ni}^0 + t_{nvi} y_{nvi} + t_{nv'ij} - t_{nj}^0 - (2 - \sum_{v \in V} (y_{nvj} + x_{nvij}))) & \\
 M_{nij} + \sum_{n \in N} \sum_{j \in K} \beta_{nj} (\sum_{i \in AK(j)} \sum_{v \in V} x_{nvij} - \sum_{v \in V} y_{nvj}) &
 \end{aligned}$$

$$\text{S.A. } (5.66), (5.68), (5.70), (5.71) \quad e$$

$$\sum_{v \in V} y_{nvj} \leq 1, \forall n \in N, \forall j \in K \quad (5.72)$$

onde $\gamma = (\alpha, \beta)$

ou ainda,

$$\begin{aligned}
 V(R3\gamma) = \text{Mínimo} & \sum_{n \in N} \sum_{i \in K} \sum_{v \in V} \sum_{j \in PK(i)} (c_{nvij} + \\
 + \sum_{v' \in V} \alpha_{nvv'ij} M_{nij} + \beta_{nj}) x_{nvij} & + \sum_{n \in N} \sum_{v \in V} \sum_{j \in PK(i)} c_{nvj} + \\
 + \sum_{i \in K} \sum_{v' \in V} \alpha_{nvv'ij} (t_{nvi} + M_{nij}) - \beta_{nj} \sum y_{nvj} & + \\
 + \sum_{n \in N} \sum_{v \in V} \sum_{v' \in V} \sum_{i \in K} \sum_{j \in PK(i)} \alpha_{nvv'ij} [t_{ni} - t_{nj}] & + \\
 + \sum_{n \in N} \sum_{v \in V} \sum_{v' \in V} \sum_{i \in K} \sum_{j \in PK(i)} \alpha_{nvv'ij} (t_{nv'ij} - 2M_{nij}) & \quad (5.73)
 \end{aligned}$$

$$\text{S.A. } \sum_{n \in N} \sum_{v \in V} q_{ni} y_{nvi} \geq d_i, \forall i \in K \quad (5.74)$$

$$\sum_{v \in V} y_{nvj} \leq 1, \forall n \in N, \forall j \in K \quad (5.75)$$

$$x_{nvi j} \in \{0, 1\}, y_{nvj} \in \{0, 1\}, \forall n \in N, \forall v \in V, \forall i \in K, \forall j \in PK(i) \quad (5.76)$$

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} y_{nvj}) N_{nvj}, \quad (5.77)$$

$$\text{onde } N_{nj} \geq b_j + t_{nvj}, \forall n \in N, \forall j \in K, \forall v \in V$$

$$a_j \leq t_{nj}^0 \leq b_j, \forall n \in N, \forall j \in K \quad (5.78)$$

$$\text{onde } \gamma = (\alpha, \beta)$$

Definindo:

$$\hat{c}_{nvi j} = c_{nvi j} + \sum_{v' \in V} \alpha_{nvv'ij} M_{nij} + \beta_{nj}$$

$$\hat{c}_{nvj} = c_{nvj} + \sum_{i \in K} \sum_{v' \in V} \alpha_{nvv'ij} (t_{nvi} + M_{nij}) - \beta_{nj}$$

$$KU = \sum_{n \in N} \sum_{v \in V} \sum_{v' \in V} \sum_{i \in K} \sum_{j \in PK(i)} \alpha_{nvv'ij} (t_{nv'ij} - 2M_{nij})$$

O problema (R37) pode ser reescrito na forma:

$$\begin{aligned} V(R37) = \text{Min. } & \sum_n \sum_v \sum_{i,j} \hat{c}_{nvi j} x_{nvi j} + \sum_n \sum_v \sum_j \hat{c}_{nvj} y_{nvj} + \\ & + \sum_{n \in N} \sum_{v \in V} \sum_{v' \in V} \sum_{i \in K} \sum_{j \in PK(i)} \alpha_{nvv'ij} (t_{ni}^0 - t_{nj}^0) + KU \end{aligned} \quad (5.79)$$

$$\text{S.A. } \sum_{n \in Ni} \sum_{v \in V} q_{ni} y_{nvi} \geq d_i, \forall i \in K \quad (5.80)$$

$$\sum_{v \in V} y_{nvi} \leq 1, \quad \forall n \in N, \quad \forall i \in K \quad (5.81)$$

$$x_{nvi} \in \{0, 1\}, \quad y_{nvj} \in \{0, 1\}, \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K, \quad \forall j \in K \quad (5.82)$$

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} y_{nvj}) N_{nvj}, \quad (5.83)$$

$$\text{onde } N_{nvj} \geq b_j + t_{nvj}, \quad \forall n \in N, \quad \forall v \in V, \quad \forall j \in K$$

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall n \in N, \quad \forall j \in K \quad (5.84)$$

Portanto o problema dual lagrangeano gerado a partir do problema relaxado (R3 γ) é descrito como:

$$R3 \quad (V(R3) = \underset{\gamma}{\text{Máximo}} (R3\gamma)) \quad (5.85)$$

$$\text{onde } \gamma = (\alpha, \beta) \text{ com } \alpha = (\alpha_{nvv'ij}), \quad \beta = (\beta_{nj}) \text{ e}$$

$$\alpha_{nvv'ij} \geq 0, \quad \beta_{nj} \text{ irrestritos em sinal}$$

$$\forall n \in N, \quad \forall v, v' \in V, \quad \forall i \in K, \quad \forall j \in K.$$

O problema (R3 γ) descrito por (5.79) a (5.84) depende somente das variáveis de decisão y_{nvi} e das variáveis contínuas de tempo t_{ni}^0 .

Uma possibilidade para reduzir a sua dimensão seria relaxar também as restrições de acoplamento (5.83), obtendo com isso dois subproblemas menores e independentes, um em função dos y_{nvi} e o outro em função dos t_{ni}^0 .

A inconveniência deste último procedimento está no aumento considerável do número de multiplicadores que em (R3 γ) já é muito elevado.

Mostramos a seguir, o nível de redução da relaxação lagrangeana (R3 γ) com o vetor γ fixado.

FORMULAÇÃO	Nº RESTRIÇÕES	Nº VARIÁVEIS ORIGINAIS	Nº VARIÁVEIS INCLUINDO FOLGA
(F2)	309.050	154.500	462.050
(R3 γ)	7.550	4.500	12.050

Tabela 5.2.1: Exemplo com $|N| = 30$, $|K| = 50$ e $|V| = 2$.

TÉCNICAS DE DECOMPOSIÇÃO LAGRANGEANA APLICADA AO PROBLEMA (F2)

DI) DECOMPOSIÇÃO LAGRANGEANA 3:

A decomposição lagrangeana proposta inicialmente é a versão clássica (Veja *Guinard e Kim* (1987a) e (1987b)) aplicada ao problema relaxado (R3 γ).

Este procedimento gera uma cópia z_{nvi} para cada variável de decisão y_{nvi} , acrescida das restrições de

igualdade $z_{nvi} = y_{nvi}$, $\forall n \in N$, $\forall v \in V$, $\forall i \in K$.

$$\text{Considere } X = \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in PK(i)} \hat{c}_{nvi} x_{nvi} + KU \quad (5.86)$$

$$\begin{aligned} V(\overline{R37}) = \text{Min } X + \sum_{n \in N} \sum_{v \in V} \sum_{j \in K} \hat{c}_{nvj} y_{nvj} + \\ + \sum_{n \in N} \sum_{v \in V} \sum_{v' \in V} \sum_{i \in K} \sum_{j \in PK(i)} \alpha_{nvv'ij} (t_{ni}^0 - t_{nj}^0) \end{aligned} \quad (5.87)$$

$$\text{S.A. } \sum_{n \in N} \sum_{v \in V} q_{ni} y_{nvi} \geq d_i, \quad \forall i \in K \quad (5.88)$$

$$\sum_{v \in V} y_{nvi} \leq 1, \quad \forall i \in K, \quad \forall n \in N \quad (5.89)$$

$$y_{nvi} \in (0, 1), \quad z_{nvi} \in (0, 1), \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K \quad (5.90)$$

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} z_{nvj}) N_{nvj}, \quad \text{onde} \quad (5.91)$$

$$N_{nvj} \geq b_j + t_{nvj}, \quad \forall n \in N, \quad \forall j \in K, \quad \forall v \in V$$

$$y_{nvj} = z_{nvj}, \quad \forall n \in N, \quad \forall v \in V, \quad \forall j \in K \quad (5.92)$$

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall n \in N, \quad \forall j \in K \quad (5.93)$$

Relaxando as restrições de acoplamento (5.92), obtemos o seguinte problema relaxado:

$$\begin{aligned} V(D37) = x + \text{Mínimo } \sum_{n \in N} \sum_{v \in V} \sum_{j \in K} \hat{c}_{nvj} y_{nvj} + \\ + \sum_{n \in N} \sum_{v \in V} \sum_{v' \in V} \sum_{i \in K} \sum_{j \in PK(i)} \alpha_{nvv'ij} (t_{ni}^0 - t_{nj}^0) + \\ + \sum_{n \in N} \sum_{v \in V} \sum_{j \in K} \mu_{nvj} (y_{nvj} - z_{nvj}) \end{aligned}$$

S.A. (5.88), (5.89), (5.90), (5.91), (5.93)

onde $\eta = (\gamma, \mu)$ e $\mu = (\mu_{nvj})$ com μ_{nvj} irrestritos em sinal para todo $n \in N, v \in V, j \in K$.

Observe que o problema (D3 η) pode ser decomposto em dois subproblemas menores e independentes sendo um em função das variáveis y_{nvj} e o outro em função de z_{nvj} e t_{nj}^0 como mostramos a seguir:

$$V(D3Y\eta) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{j \in K} (\hat{c}_{nvj} + \mu_{nvj}) y_{nvj} \quad (5.94)$$

$$\text{S.A.} \quad \sum_{n \in N_i} \sum_{v \in V} q_{ni} y_{nvi} \geq d_i, \quad \forall i \in K \quad (5.95)$$

$$\sum_{v \in V} y_{nvi} \leq 1, \quad \forall n \in N, \forall i \in K \quad (5.96)$$

$$y_{nvi} \in (0, 1), \quad \forall n \in N, \forall v \in V, \forall i \in K \quad (5.97)$$

e

$$V(D3T\eta) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{v' \in V} \sum_{i \in K} \sum_{j \in PK(i)} \alpha_{nvv'ij} (t_{ni}^0 - t_{nj}^0) - \sum_{n \in N} \sum_{v \in V} \sum_{j \in K} \mu_{nvj} z_{nvj} \quad (5.98)$$

$$\text{S.A.} \quad U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} z_{nvj}) N_{nvj}, \quad (5.99)$$

$$\text{onde} \quad N_{nvj} \geq b_j + t_{nvj}, \quad \forall n \in N, \forall v \in V, \forall j \in K$$

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall n \in N, \forall j \in K \quad (5.100)$$

$$z_{nvj} \in (0, 1), \quad \forall n \in N, \forall v \in V, \forall j \in K \quad (5.101)$$

Assim, para cada vetor multiplicador $\eta = (\alpha, \beta, \mu)$ fixado, onde $\alpha = (\alpha_{nvv'ij})$, $\beta = (\beta_{nj})$, $\mu = (\mu_{nvj})$ e $\alpha_{nvv'ij} \geq 0$ e β_{nj} e μ_{nvj} são irrestritos em sinal teremos:

$$V(D3\eta) = X + V(D3Y\eta) + V(D3T\eta) \quad (5.102)$$

e

$$V(D3) = \underset{\eta}{\text{Máximo}} (V(D3\eta)) \quad (5.103)$$

Mostramos a seguir, o nível de redução nos problemas primais (R3Y) e (D3 η) frente ao problema original (F2).

FORMULAÇÃO	Nº RESTRIÇÕES	Nº VARIÁVEIS ORIGINAIS	Nº VARIÁVEIS INCLUINDO AS DE FOLGA
(F2)	309.050	154.500	462.050
(R3Y)	7.550	4.500	12.050
(D3Y η)	100	3.000	3.100
(D3T η)	6.000	1.500	7.500

Tabela (5.2.2): Exemplo com $|N| = 30$ navios,
 $|K| = 50$ trips, $|V| = 2$ velocidades.

Os números da tabela (5.2.2) mostram uma redução significativa nas dimensões dos problemas primais gerados pelas técnicas de decomposição lagrangeana frente aos gerados via relaxação lagrangeana convencional.

A dificuldade nos dois procedimentos de relaxação fica por conta do elevado número de multiplicadores duais gerados ao relaxar conjuntos de restrições de (F2).

Pensando nisso, apresentamos a seguir algumas técnicas alternativas de decomposição lagrangeana com o propósito de reduzir o número de variáveis duais.

Um desses procedimentos, já citado em (F1), consiste em gerar cópias para todas as variáveis y_{nvj} , mas relaxando a exigência de que todas as cópias sejam iguais às variáveis originais correspondentes, ao invés disso, supõe-se apenas que uma combinação linear das variáveis originais seja igual a uma combinação das cópias correspondentes, como mostrado a seguir:

DII) DECOMPOSIÇÃO LAGRANGEANA 4:

Considere o problema relaxado $(\overline{R3\gamma})$ substituindo as restrições (5.92) pelas igualdades:

$$\sum_{v \in V} y_{nvj} = \sum_{v \in V} z_{nvj}, \quad \forall n \in N, \quad \forall j \in K \quad (5.104)$$

Obtendo o seguinte problema:

$$(DP4\gamma) \quad \left\{ \begin{array}{l} V(DP4\gamma) = V(\overline{R3\gamma}) \quad (\text{definido em (5.87)}) \\ \text{S.A. (5.88) até (5.91), (5.93) e (5.104)} \end{array} \right.$$

Relaxando as restrições (5.104) com multiplicadores correspondentes θ_{n_j} irrestritos em sinal, geramos um novo problema relaxado denotado por $(D4\bar{\eta})$, onde $\bar{\eta} = (\gamma, \theta)$

Não é difícil perceber que valem as seguintes afirmativas:

i) $(D4\bar{\eta})$ possui número menor de multiplicadores do que $(D3\eta)$. De fato, dimensão $\eta = \text{dimensão } (\gamma + \mu)^* = \text{dimensão } (\gamma + 2\theta) > \text{dimensão } (\gamma + \theta) = \text{dimensão } (\bar{\eta})$.

(*) Onde $\mu = (\mu_{nv_j})$ e $\theta = (n_j)$ onde $|V| = 2$

ii) Chamando D4 $\left[\begin{array}{l} V(D4) = \underset{\bar{\eta}}{\text{Máximo}} \{V(D4\bar{\eta})\} \end{array} \right]$

Então $V(D4) \leq V(D3)$

De fato, $(D4)$ é uma relaxação de $(D3)$.

Mostramos a seguir, o número de multiplicadores lagrangeanos necessários para cada problema relaxado analisado nesta seção:

PROBLEMA	VETOR MULTIPLICADOR	DIMENSÃO DO VETOR MULTIPLICADOR
(R3 γ)	$\gamma = (\alpha_{nvv'lj}, \beta_{nj})$	301.500
(D3 η)	$\eta = (\alpha_{nvv'lj}, \beta_{nj}, \mu_{nvj})$	304.500
(D4 $\bar{\eta}$)	$\bar{\eta} = (\alpha_{nvv'lj}, \beta_{nj}, \theta_{nj})$	303.000

Tabela (5.2.3): Exemplo com $|N| = 30$, $|K| = 50$, $|V| = 2$

CONCLUSÕES DE (F2)

Os resultados das tabelas (5.2.2) e (5.2.3) mostram que embora as técnicas de relaxação e decomposição lagrangeana reduzam drasticamente as dimensões dos problemas primais (Tabela 5.2.2) tornando-os viáveis em termos de implementação, o mesmo otimismo não é observado com respeito à resolução global dos problemas lagrangeanos, devido ao elevado número de multiplicadores (Tabela 5.2.3).

A explicação para tais resultados é óbvia, basta recordarmos que na relaxação lagrangeana convencional (R3 γ), relaxamos as restrições (5.69) de (F2) cuja cardinalidade é muito elevada, trazendo como consequência este desequilíbrio entre o número de variáveis primais e duais nos problemas relaxados.

Uma consequência desse impacto é a indiferença (em número de multiplicadores) de se trabalhar com relaxações convencionais e as decomposições lagrangeanas (Tabela 5.2.3).

Surge então a necessidade de rever a formulação (F2) procurando torná-lo mais simples sem se distanciar do problema real, como mostraremos posteriormente.

5.C - ANÁLISE DA FORMULAÇÃO 3 (F3)

A formulação (F3) trabalha somente com as variáveis de decisão (X_{nvi}) e com as variáveis contínuas de tempo (t_{ni}^o).

$$V(F3) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} x_{nvi} \quad (5.105)$$

$$\text{S.A.} \sum_{n \in N} \sum_{v \in V} q_{ni} x_{nvi} \geq D_i, \quad \forall i \in K \quad (5.106)$$

$$\sum_{v \in V} x_{nvi} \leq 1, \quad \forall n \in N, \quad \forall i \in K \quad (5.107)$$

$$x_{nvi} \in \{0,1\}, \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K \quad (5.108)$$

$$t_{ni}^o + t_{nvi} + t_{nij} - t_{nj}^o \leq (2 - \sum_{v \in V} x_{nvi} - \sum_{v \in V} x_{nvj}) M_{nvi j}$$

onde $M_{nvi j} \geq b_i + t_{nvi} + t_{nij} - a_j$, (5.109)

$$\forall n \in N, \quad \forall v \in V, \quad \forall i \in K, \quad \forall j \in PK(i)$$

$$a_i \leq t_{ni}^o \leq b_i, \quad \forall n \in N, \quad \forall i \in K \quad (5.110)$$

$$U_i \geq t_{ni}^o + \sum_{v \in V} t_{nvi} x_{nvi}, \quad \forall i \in K, \quad \forall n \in N \quad (5.111)$$

REDUÇÃO DO PROBLEMA (F3) A UM PROBLEMA DE SELEÇÃO MÚLTIPLA COM RESTRIÇÕES DE "SCHEDULING"

Com o intuito de viabilizar o uso de técnicas de relaxação e decomposição lagrangeana em (F3), podemos supor sem perda de generalidade, uma restrição adicional de que a demanda de cada trip $i \in K$ seja atendida numa única visita e por um único navio $n \in N$.

Do ponto de vista prático, isto pode ser resolvido da seguinte forma: se um trip i possuir uma demanda D_i superior à capacidade de qualquer navio $n \in N$, referente à carga do trip i , dividimos D_i em um conjunto de novos trips com demandas $d_i^j < D_i$, todos ocupando o mesmo espaço físico do trip original i e tais que $\sum_j d_i^j = D_i$. Além disso, na geração dos conjuntos N_k incluímos a condição $q_{nk} \geq D_k$ para algum navio $n \in N$.

Tal modificação equivale a substituir as restrições (5.106) e (5.107) de (F3) pelas restrições

$$\sum_{n \in N_i} \sum_{v \in V} x_{nvi} = 1, \quad \forall i \in K$$

A formulação resultante, denotada por (F3M) será da forma:

$$V(F3M) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} x_{nvi} \quad (5.112)$$

$$\text{S.A.} \quad \sum_{n \in N_i} \sum_{v \in V} x_{nvi} = 1, \quad \forall i \in K \quad (5.113)$$

$$x_{nvi} \in (0,1), \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K \quad (5.114)$$

$$t_{ni}^0 + t_{nvi} + t_{nij} - t_{nj}^0 \leq (2 - \sum_{v \in V} x_{nvi} - \sum_{v \in V} x_{nvj}) M_{nvi j}$$

onde $M_{nvi j} \geq b_i + t_{nvi} + t_{nij} - a_j$, (5.115)

$$\forall n \in N, \quad \forall v \in V, \quad \forall i \in K, \quad \forall j \in PK(i)$$

$$a_i \leq t_{ni}^0 \leq b_i, \quad \forall i \in K, \quad \forall n \in N \quad (5.116)$$

$$U_i \geq t_{ni}^0 + \sum_{v \in V} t_{nvi} x_{nvi}, \quad \forall i \in K, \quad \forall n \in N \quad (5.117)$$

Na formulação (F3M), as restrições de roteamento (5.112) a (5.114) definem um problema de seleção múltipla que pode ser resolvido por inspeção, escolhendo para cada trip $i \in K$, um $x_{n^* v^* i} = 1$, com $n^* \in N$, $v^* \in V$ tais que $C_{n^* v^* i} = \text{mínimo} \{c_{nvi}, n \in N, v \in V\}$ e fixando $x_{nvi} = 0$ para os outros $n \in N, v \in V$.

Mostramos, a seguir, o uso de técnicas de relaxação lagrangeana convencional e de algumas decomposições lagrangeanas aplicadas ao problema (F3M).

RIV) RELAXAÇÃO LAGRANGEANA CONVENCIONAL 4:

Relaxamos aqui, uma das restrições de acoplamento de (F3M), a saber o conjunto (5.115).

Considere $\alpha_{nvi j} \geq 0$, os multiplicadores lagrangeanos associados às restrições (5.115). Então o problema relaxado será da forma:

$$V(R4\alpha) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} x_{nvi} + \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in K} \alpha_{nvi j} (t_{ni}^0 + t_{nvi} + t_{nij} - t_{nj}^0 - (2 - \sum_{v \in V} x_{nvi} - \sum_{v \in V} x_{nvj}) M_{nvi j})$$

S.A. (5.113), (5.114), (5.116), (5.117)

ou

$$V(R4\alpha) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \left[c_{nvi} + \sum_{j \in K} (\alpha_{nvi j} M_{nvi j} + \alpha_{nvji} M_{nvji}) \right] x_{nvi} + \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in K} \left[(\alpha_{nvi j} - \alpha_{nvji}) t_{ni}^0 + \alpha_{nvi j} [(t_{nvi} + t_{nij} - 2M_{nvi j})] \right]$$

S.A. (5.113), (5.114), (5.116), (5.117)

Em (R4 α) para cada valor de $\alpha = (\alpha_{nvi j})$ fixado obtemos um limite inferior do valor de $V(F3M)$ e como numa solução viável de (F3M) as quantidades:

$$\sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in K} \alpha_{nvi j} (t_{ni}^0 + t_{nvi} + t_{nij} - t_{nj}^0 - (2 - \sum_{v \in V} x_{nvj} - \sum_{v \in V} x_{nvj}) M_{nvi j})$$

são não positivos, então estamos interessados em gerar um vetor multiplicador $\alpha = (\alpha_{nvi j}) \geq 0$, tal que:

$$(R4) \quad \left(\begin{array}{l} V(R4) = \text{Máximo } (V(R4\alpha)) \\ \alpha \geq 0 \end{array} \right) \quad (5.118)$$

(5.118) é um problema do tipo maxmin e pode ser resolvido utilizando por exemplo, métodos do tipo subgradiente.

Para o exemplo dado anteriormente com $|K| = 50$, $|N| = 30$ e $|V| = 2$, o problema (R4) terá as seguintes dimensões:

	NÚMERO DE RESTRIÇÕES	NÚMERO DE VARIÁVEIS	NÚMERO VARIÁVEIS COM AS DE FOLGA
PROBLEMA PRIMAL	4550	4500	9000
PROBLEMA DUAL	—	150000	—

Tabela 5.3.1

CONCLUSÕES DE (R4)

Na tabela (5.3.1), observamos que as técnicas de relaxação lagrangeana convencional geram um problema primal de dimensões aceitáveis, mas por outro lado, com um número de multiplicadores lagrangeanos muito elevado.

A dificuldade em viabilizar (F3M) é a dimensão das restrições (5.115) que no exemplo com $|K| = 50$, $|N| = 30$ e $|V| = 2$, fornece 150.000 restrições ou multiplicadores lagrangeanos após uma relaxação lagrangeana.

Para superar este problema, propomos a seguir técnicas que procuram suavizar o impacto de (5.115) no problema (F3M).

UMA TÉCNICA PARA CONVERTER PROBLEMAS DE FROTA HETEROGÊNEA EM PROBLEMAS DE FROTA HOMOGÊNEA USANDO RELAXAÇÃO LAGRANGEANA CONVENCIONAL.

Note que o problema (F3M) a princípio não pode ser decomposto em um conjunto de subproblemas independentes com frota homogênea.

Isto ocorre devido à presença das restrições (5.113) de (F3M) que não pode ser representado separadamente para cada

tipo de navio $n \in N$.

A nossa proposta é a de convertero problema (F3M) em $|N|$ subproblemas menores, independentes e com frota homogênea, através de uma relaxação lagrangeana conveniente.

Com isso, estaremos reduzindo drasticamente a dimensão do problema a ser resolvido.

Considere o problema (F3M) reescrito da seguinte forma:

$$V(\text{F3M}) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} x_{nvi} \quad (5.119)$$

$$\text{S.A.} \quad \sum_{n \in N_i} \sum_{v \in V} x_{nvi} = 1, \quad \forall i \in K \quad (5.120)$$

$$x_{nvi} \in (0,1), \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K \quad (5.121)$$

$$t_{1j}^0 + t_{1vi} + t_{1ij} - t_{1j}^0 \leq (2 - \sum_{v \in V} x_{1vi} - \sum_{v \in V} x_{1vj}) M_{1vij} \quad (5.121^1)$$

$$t_{2i}^0 + t_{2vi} + t_{2ij} - t_{2j}^0 \leq (2 - \sum_{v \in V} x_{2vi} - \sum_{v \in V} x_{2vj}) M_{2vij} \quad (5.121^2)$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$t_{|N|c}^0 + t_{|N|vi} + t_{|N|ij} - t_{|N|j}^0 \leq (2 - \sum_{v \in V} x_{|N|vi} - \sum_{v \in V} x_{|N|vj}) M_{|N|vij} \quad (5.121^{|N|})$$

$$\text{onde } M_{nvi j} \geq b_i + t_{nvi} + t_{nij} - a_j, \\ \forall n \in N, \forall v \in V, \forall i \in K, \forall j \in PK(i)$$

$$a_i \leq t_{1i}^0 \leq b_i, \quad \forall i \in K \quad (5.122^1)$$

$$a_i \leq t_{2i}^0 \leq b_i, \quad \forall i \in K \quad (5.122^2)$$

$$: \quad : \quad :$$

$$: \quad : \quad :$$

$$. \quad . \quad .$$

$$a_i \leq t_{|N| i}^0 \leq b_i, \quad \forall i \in K \quad (5.122^{|N|})$$

$$U_i \geq t_{1i}^0 + \sum_{v \in V} t_{1vi} x_{1vi}, \quad \forall i \in K \quad (5.123^1)$$

$$U_i \geq t_{2i}^0 + \sum_{v \in V} t_{2vi} x_{2vi}, \quad \forall i \in K \quad (5.123^2)$$

$$: \quad : \quad :$$

$$: \quad : \quad :$$

$$. \quad . \quad .$$

$$U_i \geq t_{|N| i}^0 + \sum_{v \in V} t_{|N| vi} x_{|N| vi}, \quad \forall i \in K \quad (5.123^{(N)})$$

R.V) RELAXAÇÃO LAGRANGEANA CONVENCIONAL 5:

Relaxando as restrições (5.120) de (F3M), obtemos o seguinte problema denotado por (R5/β).

$$V(R5/\beta) = \text{Mín.} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} (c_{nvi} - \beta_i) x_{nvi} + \sum_{i \in K} \beta_i \quad (5.124)$$

$$\text{S.A. (5.121) - (5.123)}^{|N|}$$

onde $\beta = (\beta_i)$ são multiplicadores irrestritos em sinal correspondentes às equações (5.120)

O problema (R5/ β) por sua vez, pode ser decomposto em $|N|$ subproblemas independentes como mostrado a seguir.

SUBPROBLEMA (N=1) : R5/ $\beta(1)$

$$V(R5/\beta(1)) = \text{Mín.} \sum_{v \in V} \sum_{i \in K} (c_{1vi} - \beta_i) x_{1vi} + \sum \beta_i \quad (5.125)$$

$$\text{S.A. } x_{1vi} \in (0,1), \forall v \in V, \forall i \in K \quad (5.126)$$

$$t_{1i}^0 + t_{1vi} + t_{1ij} - t_{1j}^0 \leq (2 - \sum_{v \in V} x_{1vi} - \sum_{v \in V} x_{1vj}) M_{1vij} \quad (5.127)$$

onde

$$M_{1vij} \geq b_i + t_{1vi} + t_{1ij} - a_j, \forall v \in V, \forall i \in K, \forall j \in K$$

$$a_i \leq t_{1i}^0 \leq b_i, \forall i \in K \quad (5.128)$$

$$U_i \geq t_{1i} + \sum_{v \in V} t_{1vi} x_{1vi}, \forall i \in K \quad (5.129)$$

De forma análoga podemos definir os $|N| - 1$ subproblemas restantes.

Portanto, o valor ótimo do problema primal (R5B) com β fixado, pode ser obtido somando os valores ótimos dos subproblemas $R5\beta(n)$, $n = \overline{1, |N|}$.

Deu seja,

$$V(R5\beta) = \sum_{n=1}^{|N|} V(R5\beta(n)) \quad (5.130)$$

E a melhor cota inferior dentre os vários $V(R5\beta)$, com β irrestrito em sinal é dado por

$$(R5) \quad V(R5) = \underset{\beta}{\text{Máximo}} (V(R5\beta)) \quad (5.131)$$

Além disso, da teoria de relaxação lagrangeana, temos que:

$$V(F3M) \geq V(R5) \geq V(F3ML) \quad (5.132)$$

onde (F3ML) é o problema (F3M) com relaxação linear.

CONCLUSÕES DE (F3M)

A técnica de converter problemas de roteamento com frota heterogênea em $|N|$ subproblemas com frota homogênea

via relaxação lagrangeana convencional reduz drasticamente as dimensões do problema a ser resolvido como nos mostra a tabela (5.3.2) a seguir:

PROBLEMA	NÚMERO RESTRIÇÕES	NÚMERO VARIÁVEIS PRIMAS ORIGINAIS	NÚMERO VARIÁVEIS PRIMAS C/ FOLGA	NÚMERO DE MULTIPLICADORES DUAS
(F3M)	154.550	4500	159.000	0
(R4 α)	4.550	4500	9.000	150.000*
(R5 β)	5.150	150	5.300	5.050**

Exemplo com $|K| = 50$ trips, $|N| = 30$ navios e
 $|V| = 2$ velocidades

Tabela 5.3.2

(*) Problema dual lagrangeano (R4)

(**) Problema dual lagrangeano (R5)

Além da redução das dimensões obtida em (R5) a qualidade da cota inferior de $V(F3M)$ obtida nesta relaxação possivelmente será superior ao obtido em (R4) já que em (R5) relaxamos um conjunto de restrições "simples", a saber $(\sum_{n \in N} \sum_{v \in V} x_{nvi} = 1)$ enquanto em (R4) as restrições relaxadas são do tipo "complicadas".

$$(t_{ni}^0 + t_{nvi} + t_{nij} - t_{nj}^0) \leq (2 - \sum_{v \in V} x_{nvi} - \sum_{v \in V} x_{nvj}) M_{nvi j}$$

E a prática nos mostra que deva valer as desigualdades:

$$V(R5) \geq R(R4) \quad \text{ou seja,}$$

$$V(F3M) \geq V(R5) \geq V(R4) \geq V(F3ML)$$

Finalmente, o fato de em (R5) necessitarmos a solução de $|N|$ subproblemas ao invés de somente um como em (R4), não traz maiores problemas já que em geral $|N| \leq 40$ e além disso $|N| \lll |K|$, ou seja a ordem de complexidade do problema é dominado pelo tamanho de $|K|$.

5.4 - ANÁLISE DA FORMULAÇÃO 5 (F5)

Nesta formulação supomos adicionalmente que a velocidade de cada navio da frota é única e que cada trip deve ser atendido numa única visita e por um único navio.

Na prática estas simplificações podem ter a seguinte configuração:

A velocidade única, sendo a velocidade mais reduzida, isso evitaria riscos de obter uma solução inviável e além disso daria aos operadores dos navios maior liberdade em alterar velocidades durante o percurso desde que satisfaça às restrições de time-windows e de tempo limite para concluir as tarefas subsequentes do schedule.

A restrição de única visita pode ser contornada particionando a demanda que não possa ser atendida por um único navio numa única visita, esta divisão gera um novo trip para cada parcela da partição como proposto em formulações anteriores.

FORMULAÇÃO MATEMÁTICA (F5)

$$V(F5) = \text{Mín.} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} c_{nij} x_{nij} + \sum_{n \in N} \sum_{j \in K} c_{nj} y_{nj} \quad (5.133)$$

$$\text{S.A.} \quad \sum_{n \in N} \sum_{i \in K} x_{nij} = 1, \quad \forall j \in K \quad (5.134)$$

$$x_{nij} \in (0,1), \quad y_{nj} \in (0,1), \quad \forall n \in N, \forall i \in K, \forall j \in P_k(i) \quad (5.135)$$

$$t_{ni}^0 + Q_{nij} - t_{nj}^0 \leq (1 + x_{nij} - 2y_{nj}) M_{nij}, \quad \text{onde} \quad (5.136)$$

$$M_{nij} \geq b_i + Q_{nij} - a_j, \quad \forall n \in N, \forall i \in K, \forall k \in P_k(i)$$

$$U_j + t_{nj}^0 + t_{nj} - (1 - y_{nj}) N_{nj}, \quad \text{onde} \quad (5.137)$$

$$N_{nj} \geq b_j + t_{nj}, \quad \forall n \in N, \forall j \in K$$

$$a_i \leq t_{ni}^0 \leq b_i, \quad \forall n \in N, \forall i \in K \quad (5.138)$$

No problema (F5) poderíamos, como em outras formulações (F7), trabalhar somente com as variáveis de fluxo x_{nij} ou somente com as variáveis de decisão y_{nj} além das variáveis contínuas de tempo (F3).

No entanto preferimos manter aqui ambas as variáveis para que a seguinte situação real seja satisfeita sem a necessidade de alterar a formulação e/ou a notação inicial.

Mantendo somente x_{nij} , estaríamos exigindo no problema (FP) que todo navio $n \in N$ que aportasse na origem do trip $j \in K$ obrigatoriamente atendesse o trip j .

Mas uma situação que pode ocorrer é que a origem do trip j seja exatamente o destino de algum trip i , anterior

multiplicadores duais correspondentes a (5.136), o problema (F5) relaxado tem a seguinte formulação.

$$\begin{aligned}
 V(R6\alpha) = & \text{Mínimo} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} (c_{nij} - M_{nij} \alpha_{nij}) x_{nij} + \\
 & + \sum_{n \in N} \sum_{j \in K} (c_{nj} + 2 \sum_{i \in K} M_{nij} \alpha_{nij}) y_{nj} + \\
 & + \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} \left[(t_{ni}^0 - t_{nj}^0) + (Q_{nij} - M_{nij}) \right] \alpha_{nij}
 \end{aligned}$$

S.A. (5.134) - (5.135) e (5.137), (5.138)

Definindo:

$$U = \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} (Q_{nij} - M_{nij}) \alpha_{nij}$$

$$\hat{c}_{nij}(\alpha_{nij}) = c_{nij} - M_{nij} \alpha_{nij}$$

$$\hat{c}_{nj}(\alpha_{nij}) = c_{nj} + 2 \sum_{i \in K} M_{nij} \alpha_{nij}$$

Podemos reescrever (R6 α) como:

$$\begin{aligned}
 V(R6\alpha) = & \text{Mínimo} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} \hat{c}_{nij}(\alpha_{nij}) x_{nij} + \\
 & + \sum_{n \in N} \sum_{j \in K} \hat{c}_{nj}(\alpha_{nij}) y_{nj} + \\
 & + \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} (t_{ni}^0 - t_{nj}^0) \alpha_{nij} + U \quad (5.139)
 \end{aligned}$$

$$\text{S.A.} \quad \sum_{n \in N} \sum_{i \in K} x_{nij} = 1, \quad \forall j \in K \quad (5.140)$$

$$x_{nij} \in (0,1), y_{nj} \in (0,1), \quad \forall n \in N, \quad \forall i \in K, \quad \forall j \in K \quad (5.141)$$

$$U_j \geq t_{nj}^0 + t_{nj} - (1 - y_{nj}) N_{nj}, \text{ onde} \quad (5.142)$$

$$N_{nj} \geq b_j + t_{nj}, \quad \forall n \in N, \quad \forall j \in K$$

$$a_i \leq t_{ni}^0 \leq b_i, \quad \forall n \in N, \quad \forall i \in K \quad (5.143)$$

O problema relaxado (R6 α) por sua vez pode ser decomposto em dois subproblemas independentes (R6X α) e (R6Y α) como mostrados a seguir:

$$V(R6X\alpha) = \text{Mínimo} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} \hat{c}_{nij} (\alpha_{nij}) X_{nij} \quad (5.144)$$

$$\text{S.A.} \quad \sum_{n \in N} \sum_{i \in K} x_{nij} = 1, \quad \forall j \in K \quad (5.145)$$

$$x_{nij} \in (0,1), \quad \forall n \in N, \quad \forall i \in K, \quad \forall j \in K \quad (5.146)$$

$$V(R6Y\alpha) = \text{Mínimo} \sum_{n \in N} \sum_{j \in K} \hat{c}_{nj} (\alpha_{nij}) y_{nj} + \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} (t_{ni}^0 - t_{nj}^0) \alpha_{nij} \quad (5.147)$$

S.A.

$$U_j \geq t_{nj}^0 + t_{nj} - (1 - y_{nj}) N_{nj}, \text{ onde} \quad (5.148)$$

$$N_{nj} \geq b_j + t_{nj}, \quad \forall n \in N, \quad \forall j \in K$$

$$a_i \leq t_{ni}^0 \leq b_i, \quad \forall n \in N, \quad \forall i \in K \quad (5.149)$$

E para cada vetor multiplicador $\alpha = (\alpha_{nij}) \geq 0$ fixado:

$$V(R6\alpha) = V(R6X\alpha) + V(R6Y\alpha) + U \quad (5.150)$$

Finalmente como numa solução viável de (F5) a expressão

$$(t_{ni}^0 + Q_{nij} - t_{nj}^0 - (1 + x_{nij} - 2y_{nj}) M_{nij}) z_{nij}$$

é não positiva, neste caso o valor de $V(R6\alpha)$ com $\alpha \geq 0$ que mais se aproxima de $V(F5)$ é dado por:

$$(R6) \quad (V(R6) = \text{Máx. } (V(R6\alpha), \alpha_{nij} \geq 0, \forall n \in N, \forall i \in K, \forall j \in K)$$

O problema (R6) é do tipo maxmin e pode ser resolvido através dos métodos do tipo subgradiente.

ANÁLISE DA RELAXAÇÃO (R6)

Ao relaxarmos as restrições de acoplamento (5.136) do problema (F5) geramos dois subproblemas independentes denotados por (R6X α) e (R6Y α).

O primeiro é um problema de roteamento conhecido como o "problema de seleção múltipla" que pode ser resolvido por uma simples inspeção nos coeficientes da função objetivo para cada linha $j \in K$.

O segundo é um problema de scheduling com restrições do

tipo "time-windows".

A grande vantagem deste procedimento é a redução no número de restrições e variáveis de cada problema primal como mostra a tabela (5.4.1) dada a seguir):

PROBLEMA PRIMAL	NÚMERO RESTRIÇÕES	NÚMERO DE VARIÁVEIS ORIGINAIS	NÚMERO DE VARIÁVEIS C/ AS DE FOLGA
(F5)	79.550	78.000	175.500
(R6X α)	50	75.000	75.000
(R6Y α)	4.500	3.000	7.500

Exemplo com $|K| = 50$ trips, $|N| = 30$ navios

Tabela (5.4.1)

A tabela (5.4.1) nos mostra que apesar das reduções obtidas em (R6 α), ainda temos que trabalhar com um problema com cerca de 75.000 variáveis que na prática limita sua implementação.

Com o objetivo de reduzir as dimensões dos problemas a serem resolvidos, propomos a seguir novas técnicas para gerar uma aproximação de $V(R6)$.

Considere o dual do problema (R6X α) com relaxação linear, denotado por (Dx α).

$$V(DX\alpha) = \text{Máximo} \sum_{p \in K} z_p \quad (5.152)$$

$$\text{S.A.} \sum_{p \in K} e_{ni p} z_p \leq \hat{c}_{ni j} (\alpha_{ni j}) \quad \forall n \in N, \forall i \in K, \forall j \in K \quad (5.153)$$

$$z_p \text{ irrestrito em sinal e onde} \quad (5.154)$$

$$e_{ni p} = \begin{cases} 1, & \text{se } p = j \\ 0, & \text{c/c} \end{cases}$$

Muitas das inequações (5.153) são redundantes e podem ser detectadas através do seguinte procedimento:

$$\text{Seja } \hat{c}_j (\alpha_j) = \text{Mínimo}_{(n, i)} \{ \hat{c}_{ni j} (\alpha_{ni j}) \}, \quad \forall j \in K \quad (5.155)$$

Portanto, em (5.153) conservamos apenas uma restrição para cada $j \in K$, obtendo o seguinte problema:

$$V(DX\alpha) = \text{Máximo} \sum_{j \in K} z_j \quad (5.156)$$

$$(DX\alpha) \quad \text{S.A.} \sum_{j \in K} z_j \leq \hat{c}_k (\alpha_k), \quad \forall i \in K \quad (5.157)$$

$$z_j \text{ irrestrito em sinal} \quad (5.158)$$

Note que à primeira vista o problema (DX α) não pode ser resolvido por Métodos de Programação Linear, já que todas as variáveis z_j são todas irrestritas em sinal.

Mas isso pode ser facilmente superado efetuando a seguinte transformação:

$$z_j = x_j - y_j, \text{ onde } x_j, y_j \geq 0, \text{ p/ todo } j \in K \quad (5.159)$$

Substituindo (5.159) em (DX α) definido por (5.156)-(5.158), obtemos o seguinte problema de Programação Linear.

$$V(DX\alpha) = \text{Minimizar } \sum_{j \in K} (y_j - x_j) \quad (5.160)$$

$$(DX\alpha) \text{ S.A. } \sum_{j \in K} x_j - \sum_{j \in K} y_j \leq \hat{c}_j(\alpha_j), \quad \forall i \in K \quad (5.161)$$

$$x_j \geq 0, y_j \geq 0 \quad \forall j \in K \quad (5.162)$$

A dimensão do problema (DX α) descrito por (5.160) - (5.162) é $|K|$ restrições por 2 $|K|$ variáveis ao invés de $|K|$ restrições por $|N| \times |K|^2$ variáveis do problema (R δ X α).

Portanto para obter uma aproximação de $V(R\delta\alpha)$ com $\alpha = \alpha^k$ fixado, basta resolvermos dois subproblemas (DX α^k) e (R δ Y α^k) com dimensões (50 x 2500) e (4500 x 300) respectivamente (veja tabela (5.4.2)).

Isto é, para cada vetor multiplicador $\alpha = (\alpha_{ni_j}) \geq 0$ fixado:

$$V(R\delta\alpha) = V(DX\alpha) + V(R\delta Y\alpha) + U \quad (5.163)$$

PROBLEMA PRIMAL	NÚMERO RESTRIÇÕES	NÚMERO VARIÁVEIS ORIGINAIS	NÚMERO VARIÁVEIS C/ AS DE FOLGA
(F5)	79.550	78.000	157.500
(DX α)	50	2.500	2.550
(R6Y α)	4.500	3.000	7.500

Exemplo com $|K| = 50$ trips e $|N| = 30$ navios

Tabela (5.4.2)

O procedimento aqui desenvolvido resolve o problema das dimensões de cada problema primal (vetor multiplicador fixado), como mostra a tabela 5.4.2.

A dificuldade passa a ser unicamente o número elevado de multiplicadores duais $\alpha_{ntj} \geq 0$, do problema dual lagrangeano que no exemplo dado na tabela (5.4.2) se situa em torno de 75.000.

Para superar este problema, desenvolvemos um outro esquema de resolver (F5) relaxando um outro conjunto de restrições.

RVID) RELAXAÇÃO LAGRANGEANA CONVENCIONAL 7:

O procedimento aqui adotado é semelhante ao usado na formulação (F3), onde transformamos um problema de roteamento com frota heterogênea em um conjunto de problemas

independentes com frota homogênea.

Considere o problema (F5) relaxando as restrições (5.134) com multiplicadores duais β_k , $k \in K$ irrestritos em sinal.

O problema resultante denotado por (R7 β) será da forma:

$$V(R7\beta) = \text{Mínimo} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} (c_{nij} - \beta_j) x_{nij} + \\ + \sum_{n \in N} \sum_{j \in K} c_{nj} y_{nj} + \sum_{j \in K} \beta_j \quad (5.164)$$

S.A.

$$x_{nij} \in (0,1), y_{nj} \in \{0,1\}, \forall n \in N, \forall i \in K, \forall j \in K \quad (5.165)$$

$$t_{ni}^0 + Q_{nij} - t_{nj}^0 \leq (1 + X_{nij} - 2y_{nj}) M_{nij}, \text{ onde} \quad (5.166)$$

$$M_{nij} \geq b_i + Q_{nij} - a_j, \forall n \in N, \forall i \in K, \forall j \in K$$

$$U_j \geq t_{nj}^0 + t_{nj} - (1 - y_{nj}) N_{nj}, \text{ onde}$$

$$N_{nj} \geq b_j + t_{nj}, \forall j \in K, \forall n \in N \quad (5.167)$$

$$a_i \leq t_{ni}^0 \leq b_j, \forall i \in K, \forall n \in N \quad (5.168)$$

com $\beta = (\beta_k)$, onde β_k é irrestrito em sinal $\forall k \in K$.

$$\text{Considere} \quad \bar{c}_{nij}(\beta_j) = c_{nij} - \beta_j \\ e \quad (5.169)$$

$$W = \sum_{j \in K} \beta_j$$

Substituindo (5.169) em (R7 β) e levando-se em conta que as restrições (5.165) - (5.168) podem ser representadas em separado para cada navio, concluímos que resolver (R7 β) equivale a resolver $|N|$ subproblemas independentes com frota homogênea como mostrado a seguir:

Subproblema para o navio $n = 1$ (R7 $\beta(1)$)

$$V(R7\beta(1)) = \text{Min.} \sum_{i \in K} \sum_{j \in K} \bar{c}_{1ij} (\beta_j) x_{1ij} + \sum_{j \in K} c_{1j} y_{1j} \quad (5.170)$$

$$\text{S.A. } x_{1ij} \in \{0,1\} \quad y_{1j} \in \{0,1\}, \quad \forall i \in K, \quad \forall j \in K \quad (5.171)$$

$$t_{1i}^0 + Q_{1ij} - t_{1j}^0 \leq (1 + x_{1ij} - 2y_{1j}) M_{1ij}, \quad (5.172)$$

$$\text{onde } M_{1ij} \geq b_i + Q_{1ij} - a_j, \quad \forall i \in K, \quad \forall j \in K$$

$$U_j \geq t_{ij}^0 + t_j - (1 - y_{1j}) N_{1j}, \quad \text{onde} \\ N_{1j} \geq b_j + t_j, \quad \forall j \in K \quad (5.173)$$

$$a_i \leq t_{1i}^0 \leq b_i, \quad \forall i \in K \quad (5.174)$$

Observe que as restrições relaxadas (5.134)

$$\sum_{n \in N} \sum_{i \in K} x_{n1j} = 1, \quad \forall j \in K, \quad \text{garantiam que todo trip } j \in K \text{ fosse}$$

atendido. Ao relaxar estas restrições o sinal dos

coeficientes $\bar{c}_{nij}(\beta_j)$ são fundamentais já que se todos os $\bar{c}_{nij}(\beta_j) \geq 0$, então teremos uma solução inviável nula: $(x_{nij} = x_{nj} = 0)$. Obviamente tal situação não ocorre quando escolhermos multiplicadores adequados, em particular $(\beta_k > 0)$.

De modo análogo se define os $|N| - i$ subproblemas restantes.

Assim para cada vetor multiplicador $\beta = (\beta_j)$ fixado obtemos um valor de $V(R7\beta)$ dado por:

$$V(R7\beta) = \sum_{n \in N} V(R7\beta(n)) + W \quad (5.175)$$

E o melhor limite inferior de $V(F5)$ dentre os $V(R7\beta)$ será dado por:

$$(R7) \left[V(R7) = \text{Máx.} \left[\begin{array}{l} V(R7\beta), \text{ com } \beta = \beta(j) \text{ irrestrito em} \\ \text{sinal} \qquad \qquad \qquad \forall j \in K \end{array} \right] \right]$$

Observe que a razão de relaxarmos exatamente as restrições (5.134) de (F5) é a de retirarmos as únicas restrições que impedem um estudo em separado para cada navio $n \in N$, relaxando (5.134) transformamos um problema de roteamento com frota heterogênea em $|N|$ subproblemas de roteamento com frota homogênea com dimensões extremamente reduzidas como mostra a tabela (5.4.3) dado a seguir:

PROBLEMA	NÚMERO DE RESTRIÇÕES DO PRIMAL	NÚMERO DE VARIÁVEIS ORIGINAIS PRIMAS	NÚMERO DE VARIÁVEIS C/FOLGA PRIMAS	NÚMERO DE MULTIPLICADORES DUAS
(F5)	79.550	78.000	157.500	—
(R7)	2.650	2.600	5.200	50

Exemplo com $|K| = 50$ trips, $|N| = 30$ navios

Tabela (5.4.3)

RELAXAÇÕES LAGRANGEANAS CONVENCIONAIS: CONCLUSÕES

As relaxações lagrangeanas convencionais aqui adotadas conseguem reduzir drasticamente as dimensões do problema primal, além disso (R7) reduz igualmente o número de multiplicadores lagrangeanos (veja tabela (5.4.3)), possibilitando daí o cálculo de uma cota inferior $V(R7)$ de $V(F5)$ resolvendo um problema de dimensões bem aceitáveis (tabela (5.4.3)).

Para concluir a análise da formulação (F5), apresentamos a seguir, a possibilidade de obter soluções viáveis ou "quase viáveis" para (F5) ao efetuar uma Decomposição Lagrangeana.

TÉCNICAS DE DECOMPOSIÇÃO LAGRANGEANA APLICADAS AO PROBLEMA (F5)

Sabemos que as decomposições lagrangeanas clássicas, além de gerar cotas inferiores para o valor ótimo do problema original (F5), tem maiores possibilidades que as relaxações lagrangeanas convencionais de produzir uma solução viável para (F5).

Propomos aqui, duas decomposições lagrangeanas sendo que na primeira, a preocupação é apenas a de obter limites inferiores para $V(F5)$, para isso decomponemos cada um dos n -subproblemas $(R7\beta(n))$, $n=1, \dots, |N|$.

A segunda decomposição é feita em cima do problema original (F5) com o propósito de também tentar gerar soluções viáveis para (F5).

DV) DECOMPOSIÇÃO LAGRANGEANA 5:

O subproblema $(R7\beta(n=1))$ descrito anteriormente pode ser reescrito na seguinte forma equivalente.

$$V(\overline{R7\beta(1)}) = \text{Mín.} \sum_{i \in K} \sum_{j \in K} \bar{c}_{1ij}(\beta) x_{1ij} + \sum_{j \in K} c_{1j} y_{1j} \quad (5.176)$$

$$\text{S.A. } x_{1ij} \in (0, 1), y_{1j} \in (0, 1), z_{1j} \in (0, 1), \forall i \in K, \forall j \in K \quad (5.177)$$

$$t_{1i}^0 + Q_{1ij} - t_{1j}^0 \leq (1 + X_{1ij} - Z_{1j}) M_{1ij} \quad (5.178)$$

$$\text{onde } M_{1ij} \geq b_i + Q_{1ij} - a_j, \quad \forall i \in K, \quad \forall j \in K$$

$$U_j \geq \bar{t}_{1j}^0 + t_j - (1 - y_{1j}) N_{1j}, \quad \text{onde} \quad (5.179)$$

$$N_{1j} \geq b_j + t_j, \quad \forall j \in K$$

$$a_i \leq t_{1i}^0, \quad \forall i \in K \quad (5.180)$$

$$b_i \geq \bar{t}_{1i}^0, \quad \forall i \in K \quad (5.181)$$

$$y_{1j} = z_{1j}, \quad \forall j \in K \quad (5.182)$$

$$t_{1i}^0 = \bar{t}_{1i}^0, \quad \forall i \in K \quad (5.183)$$

Relaxando as restrições de acoplamento (5.182) e (5.183) com multiplicadores irrestritos em sinal γ_i e η_i , $\forall i \in K$, obtemos a seguinte relaxação (D5 δ (1)) de (R7 β (1)), onde $\delta = (\beta, \gamma, \eta) \in \mathbb{R}^a |K|$ com $\beta_i \geq 0$, γ_i e η_i irrestritos em sinal para todo $i \in K$.

$$\begin{aligned} V(D5\delta(1)) = & \text{Mínimo} \sum_{i \in K} \sum_{j \in K} \bar{c}_{1ij} (\beta_j) x_{1ij} + \sum_{j \in K} c_{1j} y_{1j} + \\ & + \sum_{j \in K} \gamma_j (y_{1j} - z_{1j}) + \sum_{j \in K} \eta_j (t_{1j}^0 - \bar{t}_{1j}^0) \end{aligned}$$

$$\text{S.A. (5.177) - (5.181)}$$

O problema (D5 δ (1)) por sua vez pode ser decomposto em

dois subproblemas menores e independentes denotados por (D5X δ (1)) e (D5Y δ (1)), onde

$$V(D5X\delta(1)) = \text{Mínimo} \sum_{i \in K} \sum_{j \in K} \bar{c}_{1ij} (\beta_j) x_{1ij} - \sum_{j \in K} \gamma_j z_{1j} + \sum_{j \in K} \eta_j t_{1j}^0 \quad (5.184)$$

$$\text{S.A. } x_{1ij} \in (0,1), z_{1j} \in (0,1), \forall i \in K, \forall j \in K \quad (5.185)$$

$$t_{1i}^0 + Q_{1ij} - t_{1j}^0 \leq (1 + x_{1ij} - z_{1j}) M_{1ij}, \text{ onde } (5.186)$$

$$M_{1ij} \geq b_i + Q_{1ij} - a_j, \forall i \in K, \forall j \in K$$

$$t_{1i}^0 \geq a_i, \forall i \in K \quad (5.187)$$

onde $\delta = (\beta, \gamma, \eta)$.

$$V(D5Y\delta(1)) = \text{Mín.} \sum_{j \in K} (c_{1j} + \gamma_j) y_{1j} - \sum_{j \in K} \eta_j \bar{t}_{1j}^0 \quad (5.188)$$

$$\text{S.A. } y_{1j} \in (0,1), \forall i \in K \quad (5.189)$$

$$U_j \geq \bar{t}_{1j}^0 + t_j - (1 - y_{1j}) N_{1j}, \text{ onde } (5.190)$$

$$N_{1j} \geq b_j + t_j, \forall i \in K$$

$$\bar{t}_{1j}^0 \leq b_j, \forall j \in K \quad (5.191)$$

Assim, para cada vetor multiplicador $\delta = (\beta, \gamma, \eta)$ fixado e $n = 1$ teremos:

$$\boxed{V(D5\delta(1)) = V(D5X\delta(1)) + V(D5Y\delta(1))} \quad (5.192)$$

Aplicando este procedimento para os $|N|$ navios da frota, teremos portanto:

$$V(D5\delta) = \sum_{n \in N} V(D5\delta(n)) + W \quad (5.193)$$

onde $\delta = (\beta, \gamma, \eta)$, com $\beta = (\beta_i)$, $\gamma = (\gamma_i)$, $\eta = (\eta_i)$ e $\beta_i \geq 0$, γ_i , η_i irrestritos em sinal para todos $i \in K$.

Com isso, o valor ótimo dentre os limites inferiores $V(D5\delta)$ será dado por:

$$(D5) \quad \left(V(D5) = \text{Máximo} \{ V(D5\delta), \right. \quad \text{onde } \delta = (\beta, \gamma, \eta) \text{ e}$$

$$\left. \beta = (\beta_i), \gamma = (\gamma_i), \eta = (\eta_i) \text{ com } \beta_i \geq 0, \gamma_i \text{ e } \eta_i \right.$$

$$\left. \text{irrestritos em sinal para todo } i \in K \right) \quad (5.194)$$

CONCLUSÕES DA DECOMPOSIÇÃO (D5)

A decomposição lagrangeana (D5) comparada à relaxação lagrangeana (R7) nos mostra os seguintes resultados:

PROBLEMA	NÚMERO DE RESTRIÇÕES	NÚMERO VARIÁVEIS ORIGINAIS PRIMAIS	NÚMERO DE VARIÁVEIS COM AS DE FOLGA	NÚMERO DE MULTIPLICADORES DUAIS
(F5)	79.550	78.000	78.000	—
(R7)	2.650	2.600	5.200	50
(D5)*	1.550	2.600	4.150	50

Comparações entre as dimensões do maior problema a ser resolvido com $|K| = 50$, $|N| = 30$.

(*) subproblema $(D5X\varphi(n))$ de $(D5\varphi)$

TABELA (5.4.4)

Os resultados da tabela (5.4.4) mostram uma significativa redução no número de restrições e um aumento mínimo no número de multiplicadores duais ao passar da relaxação lagrangeana convencional (R7) para a decomposição lagrangeana (D5).

DVI) DECOMPOSIÇÃO LAGRANGEANA 6:

A decomposição lagrangeana aqui proposta se assemelha à relaxação (R7) visto anteriormente.

De fato, a idéia básica é converter o problema (F5)

relaxado em $|N|$ subproblemas homogêneos menores e independentes.

O problema (F5) pode ser reescrito na seguinte forma equivalente:

$$V(\bar{F5}) = V(F5) = \text{Mínimo} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} c_{nij} x_{nij} + \sum_{n \in N} \sum_{j \in K} c_{nj} y_{nj} \quad (5.195)$$

$$\text{S.A.} \sum_{n \in N} \sum_{i \in K} z_{nij} = 1, \quad \forall j \in K \quad (5.196)$$

$$x_{nij} \in (0, 1), z_{nij} \in (0, 1), y_{nj} \in (0, 1), \forall n \in N, \forall i \in K, \forall j \in K \quad (5.197)$$

$$t_{ni}^0 + Q_{nij} - t_{nj}^0 \leq (1 + x_{nij} - 2y_{nj}) M_{nij}, \text{ onde}$$

$$M_{nij} \geq b_i + Q_{nij} - a_j, \quad (5.198)$$

para todo $n \in N, i \in K, j \in K$

$$U_j \geq t_{nj}^0 + t_{nj} - (1 - y_{nj}) N_{nj}, \text{ onde } N_{nj} \geq b_j + t_{nj},$$

$$\forall n \in N, \forall j \in K \quad (5.199)$$

$$a_i \leq t_{ni}^0 \leq b_i, \quad \forall n \in N, \forall i \in K \quad (5.200)$$

$$\sum_{n \in N} \sum_{i \in K} z_{nij} = \sum_{n \in N} \sum_{i \in K} x_{nij}, \quad \forall j \in K \quad (5.201)$$

Relaxando as restrições de acoplamento (5.201) de $(\bar{F5})$ vamos obter o seguinte problema:

$$\begin{aligned}
 V(D6\alpha) = \text{Mínimo} & \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} (c_{nij} - \alpha_j) x_{nij} + \\
 & + \sum_{n \in N} \sum_{j \in K} c_{nj} y_{nj} + \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} \alpha_j z_{nij} \quad (5.202)
 \end{aligned}$$

S.A. (5.196) - (5.200)

onde $\alpha = (\alpha_j)$ e α_j irrestritos em sinal $\forall j \in K$.

O problema (D6 α) pode por sua vez ser decomposto em dois subproblemas independentes denotados por (D6z α) e (D6w α).

$$V(D6z\alpha) = \text{Mínimo} \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} \alpha_j z_{nij} \quad (5.203)$$

$$\text{S.A.} \sum_{n \in N} \sum_{i \in K} z_{nij} = 1, \forall j \in K \quad (5.204)$$

$$z_{nij} \in \{0, 1\}, \forall n \in N, \forall i \in K, \forall j \in K \quad (5.205)$$

$$\begin{aligned}
 V(D6w\alpha) = \text{Mínimo} & \sum_{n \in N} \sum_{i \in K} \sum_{j \in K} \hat{c}_{nij} (\alpha_j) x_{nij} + \\
 & + \sum_{n \in N} \sum_{j \in K} c_{nj} y_{nj} \quad (5.206)
 \end{aligned}$$

$$\text{S.A.} x_{nij} \in \{0, 1\}, y_{nj} \in \{0, 1\}, \forall n \in N, \forall i \in K, \forall j \in K \quad (5.207)$$

$$t_{ni}^0 + q_{nij} - t_{nj}^0 \leq (1 + x_{nij} - 2y_{nj}) M_{nij}, \text{ onde} \quad (5.208)$$

$$M_{nij} \geq b_i + Q_{nij} - a_j, \quad \forall n \in N, \quad \forall i \in K, \quad \forall j \in K$$

$$U_j \geq t_{nj}^0 + t_{nj} - (1 - y_{nj}) N_{nj}, \quad \text{onde } N_{nj} \geq b_j + t_{nj} \text{ para} \\ \forall n \in N, \quad \forall j \in K \quad (5.209)$$

$$a_i \leq t_{ni}^0 \leq b_i, \quad \forall n \in N, \quad \forall i \in K \quad (5.210)$$

Mas observe que o problema (D6W α) descrito por (5.206) - (5.210) é exatamente o problema (R7 β) visto anteriormente.

Desta forma, podemos aplicar aqui os mesmos procedimentos de transformar o problema (D6W α) em $|N|$ subproblemas homogêneos como desenvolvidos em (R7 β).

Isto é,

$$V(D6W\alpha) = \sum_{n \in N} V(D6W\alpha(n)) \quad (5.211)$$

onde $V(D6W\alpha(n))$ é o n -ésimo subproblema com frota homogênea, gerado a partir de (D6W α).

O melhor limite inferior dentre os $V(D6W\alpha)$, com $\alpha = (\alpha_j)$ irrestritos em sinal, é dado por:

$$(D6W) \quad \{ V(D6W) = \underset{\alpha}{\text{Máximo}} \{ V(D6W\alpha) \} \} \quad (5.212)$$

Analogamente, o melhor limite inferior dentre os

$V(D6z\alpha)$ será dado por:

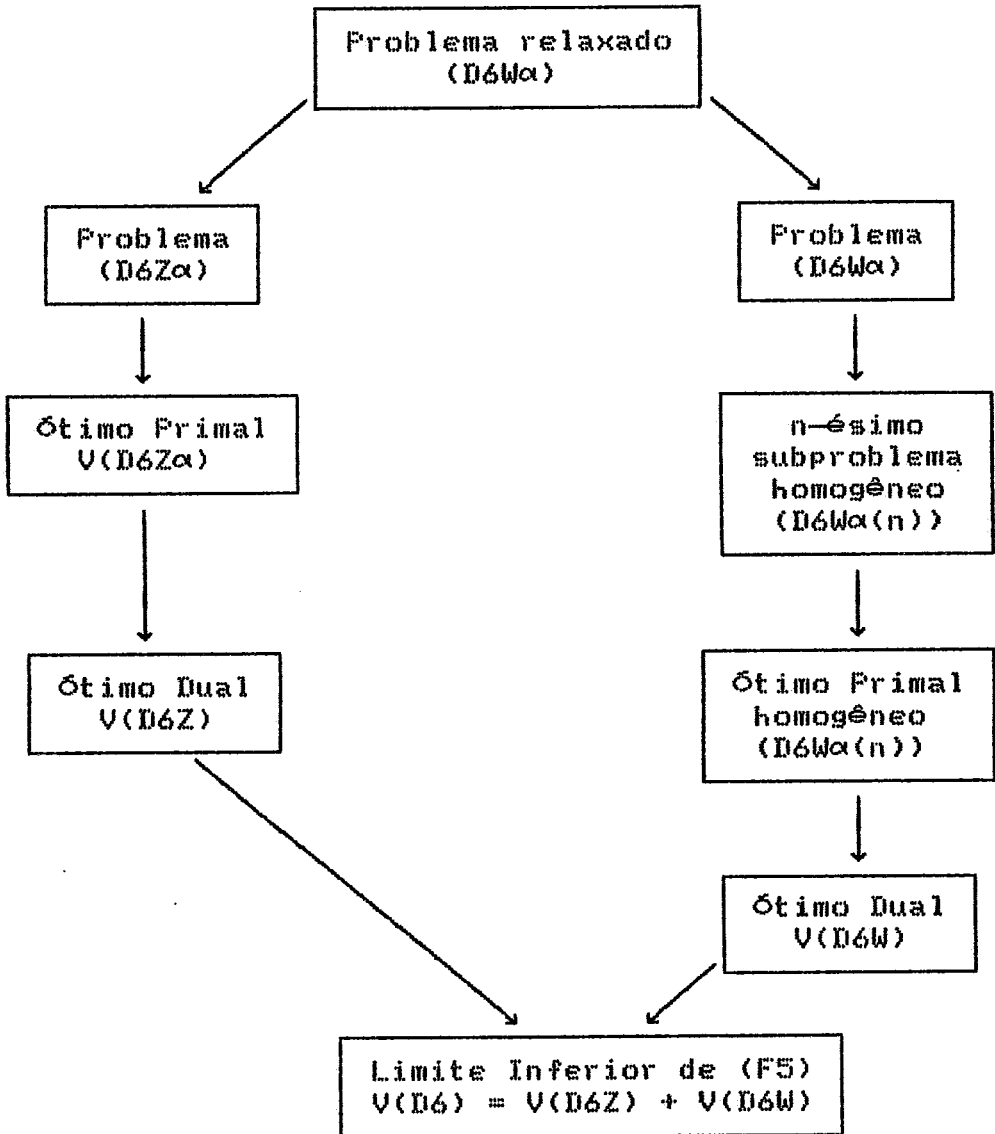
$$(D6z) \{ V(D6z) = \underset{\alpha}{\text{Máximo}} \{ V(D6z\alpha) \} \} \quad (5.213)$$

E finalmente, o melhor limite inferior de $V(F5)$ obtido nesta decomposição lagrangeana será :

$$(D6) \{ V(D6) = V(D6Z) + V(D6W) \} \quad (5.214)$$

Mostramos, a seguir, o desenvolvimento completo da decomposição lagrangeana $(D6)$ através de um fluxograma.

FLUXOGRAMA DA DECOMPOSIÇÃO LAGRANGEANA (D6)



CONCLUSÕES DA DECOMPOSIÇÃO (D6)

Em termos de complexidade, os procedimentos (D6) e (R7) são equivalentes pois ambos necessitam resolver basicamente o mesmo subproblema $(R7\beta) \equiv (D6W\alpha)$.

No problema (D6) existe o cálculo adicional de (D6z)

mas este é dominado por (D6W) em termos de complexidade.

De fato, $V(D6Z\alpha)$ pode ser obtido por uma simples inspeção nos valores dos coeficientes da sua função objetivo.

As dimensões de (D6W α) são exatamente as de (R7 β) como mostra a tabela (5.4.3), inclusive com o mesmo número de multiplicadores duais.

A vantagem de se trabalhar com (D6) ao invés de (R7) é que no primeiro não descartamos nenhuma restrição original ao contrário de (R7). Desta forma, a princípio aumenta a possibilidade de gerarmos uma solução viável para (F5) em (D6).

Finalmente, por um resultado mostrado por *M. GUINARD* e *KIM* (1987a) e (1987b), podemos garantir que o limite inferior $V(D6)$ é sempre igual ou superior ao melhor limite inferior obtido via relaxação lagrangeana convencional. Assim valem as desigualdades:

$$V(D7) \leq V(D6) \leq V(F5)$$

CAPÍTULO VI

GERAÇÃO DE LIMITES SUPERIORES:

UMA NOVA GENERALIZAÇÃO DO ALGORITMO DE CLARKE AND WRIGHT

VI.1) INTRODUÇÃO

Apresentamos um novo procedimento heurístico para a solução de problemas de roteamento e *scheduling* de navios petroleiros com restrições do tipo *time-windows*.

O algoritmo proposto pode ser visto como uma generalização do método de *CLARKE and WRIGHT (1964)* para o problema de despacho de veículos.

De fato, enquanto no algoritmo de *CLARKE e WRIGHT (1964)*, são geradas rotas sem restrições de tempo e para um conjunto homogêneo de veículos com todos saindo e retornando de um único nó origem (depósito), o nosso algoritmo gera um conjunto de *schedules* com restrições do tipo *time-windows*, múltiplos depósitos e frota heterogênea de navios.

A única simplificação que fazemos em relação ao modelo real é supor velocidade única para os navios. O modelo com múltiplas velocidades pode ser adaptado sem problemas ao nosso algoritmo introduzindo uma nova variável ao problema ou então tomando cada combinação possível de velocidades num *schedule* de um navio como um novo navio da frota, conservando neste último caso, o modelo com velocidade única.

Numa rota gerada pelo algoritmo de *CLARKE e WRIGHT (1964)*, o veículo parte do depósito carregado, para atender

um conjunto ordenado (no espaço) de nós destinos, retornando ao depósito no final da jornada, sempre respeitando as restrições de capacidade do veículo.

Enquanto isso, um *schedule* no nosso algoritmo consiste numa sequência ordenada (no espaço e no tempo) de trips i , onde cada trip i por sua vez é descrito por um local de origem, um local de destino, um tempo e um custo para percorrer o trip i .

Além disso, um trip i contém um tempo de janela *time-window*, dentro do qual os navios a ele alocados, devem aportar na sua origem, prontos para operarem.

Portanto, num *schedule* s , o navio se desloca do local onde se encontra atualmente (início do atual horizonte de planejamento), que pode ser um nó não físico (trip) ou um arco (distância direta entre dois trips) para atender uma sequência ordenada de trips i , respeitando as restrições de espaços (*routing*) e de tempo (*scheduling*).

A nossa solução inicial é semelhante à de CLARKE E WRIGHT (1964), alocando um navio n para cada trip i , criando com isso, na maioria das aplicações, um conjunto de navios artificiais para viabilizar esta solução.

Isto ocorre pelo fato do número de navios disponíveis ser geralmente menor que o número de trips do problema.

Os navios artificiais podem ser interpretados na prática como navios afretados emergencialmente no mercado para atender a um determinado trip.

O procedimento neste algoritmo é o de efetuar uma sequência de sintetizações entre dois schedules distintos.

Devido a heterogeneidade da frota, as sintetizações entre dois schedules devem ser referidas a um determinado navio. Assim dizemos que uma síntetização é viável para um navio n se as restrições de espaço e de tempo a ele envolvidas, são satisfeitas.

Nas sintetizações entre dois schedules, descarta-se sempre os schedules de navios artificiais, ou seja, somente permitimos aqui sintetizações entre um schedule do navio da frota com um schedule de um navio afretado, não considerando sintetizações entre dois schedules de navios da frota ou entre dois navios artificiais (afretados).

Para o primeiro caso, a justificativa é a de que não vale a pena descartar navios da frota disponíveis da solução final do problema, devido à escassez do mesmo.

O veto, no segundo caso, é devido a prováveis inviabilidades econômicas e/ou físicas de se afretar emergencialmente um navio no mercado para efetuar vários

trips com características às vezes totalmente distintas.

Quando a demanda de um trip i não puder ser totalmente atendida por nenhum navio da frota numa única visita, a estratégia é a de particionar a demanda em um conjunto de demandas menores e disjuntas.

Neste caso, cada parcela é vista como um novo trip com origem, destino, *time-window* idênticos ao trip original i .

As distâncias entre dois trips parcelados de um mesmo trip original é dado pela distância entre a origem e o destino (ou vice-verso) do trip i .

Finalmente o objetivo do nosso algoritmo pode ser colocado como o de minimizar o custo global de atender todos os trips, minimizando o número de navios artificiais na solução final e paralelamente gerando schedules de custo total mínimo.

VI-2) NOTACÕES DO ALGORITMO

Considere os seguintes conjuntos:

N = frota heterogênea de navios

K = conjunto de trips do atual horizonte de planejamento

R = conjunto das restrições locais do problema (por exemplo: compatibilidade entre navio-carga, navio-porto, navio-bandeira, navio-percurso, navio-demanda, etc)

$(N(k) \subset N)$ = conjunto de navios que atendem às restrições R do trip $k \in K$.

$(K(n) \subset K)$ = conjunto de trips, cujas restrições R são atendidas pelo navio $n \in N$.

$TRIP(k) = \begin{cases} k, & \text{se o trip } k \text{ não foi descartado até o momento} \\ \emptyset, & \text{se o trip } k \text{ foi descartado das análises} \\ & \text{futuras} \end{cases}$

Q_{ni} = capacidade do navio n , referente à carga do trip $i \in K$.

D_i = demanda do trip $i \in K$

$C_{ni j}$ = custo do navio n , percorrer em lastro o trecho entre o final do trip i e a origem do trip j .

C_{nk} = custo do navio n , efetuar o trip $k \in K$.

$C_{\phi ni}$ = custo do navio n aportar na origem do trip i , no início do schedule do navio n .

e_{ni}^0 = tempo estimado para o navio n chegar a origem do trip i pronto para operar, no início do schedule do navio n .

t_{ni}^0 = tempo para o navio n chegar a origem do trip i pronto para operar. Em particular, se i for o trip inicial do schedule, então $t_{ni}^0 = e_{ni}^0$.

t_{nk} = tempo para o navio n efetuar o trip $k \in K$.

$t_{ni j}$ = tempo para o navio n , se deslocar em lastro do final do trip i ao início do trip j .

U_i = tempo limite para concluir o trip $i \in K$.

$[a_i, b_i]$ = time-window do trip $i \in K$

$NTW(k) = \{ n \in N(k) / a_k \leq t_{nk}^o \leq b_k \}$

$KTW(n) = \{ k \in K(n) / a_k \leq t_{nk}^o \leq b_k \}$

$NU(l, K) = \{ n \in NTW(k) / t_{nk}^o + t_{nk} \leq U_k \}$ onde

l = número do estágio atual.

$KU(l, n) = \{ k \in KTW(n) / t_{nk}^o + t_{nk} \leq U_k \}$

$KA = \{ k \in K / N(k) \cup NTW(k) \cup NU(k) = \phi \} = \{ \text{conjunto de trips que não pode ser alocado a nenhum navio da frota} \}$.

$SEQ(n, l)$ = sequência de trips do schedule do navio n , no estágio l .

$NA1 = \{ \text{conjunto de navios artificiais alocados para atender trips } k \in K \}$.

$NA2 = \{ \text{conjunto de navios artificiais alocados para atender trips de } KA \}$

$IA(n)$ = trip alocado mais recentemente ao schedule do navio $n \in N$.

$CUSTO(n)$ = custo parcial do schedule do navio $n \in N$

VI-3) DESCRIÇÃO DO ALGORITMO

PASSO 0 : INICIALIZAÇÃO:

São dados: $N = \{ 1, 2, 3, \dots, N_{navio} \}$

$K = \{ 1, 2, 3, \dots, N_{TRIP} \}$

$l = 1$

Faça $TRIP(k) = k, \forall k \in K$.

PASSO 0.1: Gerar os conjuntos:

$N(k)$ para todo $k \in K$

$K(n)$ para todo $n \in N$

PASSO 0.1.1: Para $k = 1, 2, 3, \dots, NTRIP$

Se $N(k) = \emptyset$, faça

$K = K - \{k\}$

$KA = KA \cup \{k\}$

$TRIP(k) = \emptyset$

Ou seja, se nenhum navio $n \in N$ consegue atender às restrições locais R do trip k , então excluímos este trip de futuras considerações. Na prática a alternativa para este caso é afretar um navio no mercado para efetuar o trip k . O navio afretado será designado *navio artificial*.

PASSO 0.1.2 : Para $n = 1, 2, 3, \dots, NNAVID$

Se $K(n) = \emptyset$, faça

$N = N - \{n\}$

A condição $K(n) = \emptyset$ significa que o navio n não atende às restrições locais R de nenhum de trip $k \in K$ do atual horizonte de planejamento. Neste caso, devemos excluir o navio n de futuras considerações. Na prática, esta situação implica em deixar o navio n ocioso no atual horizonte de

planejamento.

PASSO 0.2 : Para todo trip remanescente $k \in K$ e para todo navio remanescente $n \in N$, geramos os seguintes conjuntos:

$$NTW(k), \forall k \in K$$

$$KTW(n), \forall n \in N$$

PASSO 0.2.1: $\forall k \in K$

Se $NTW(k) = \emptyset$ e $a_k > t_{nk}^0, \forall n \in N$, faça:

$$K = K - \{k\}$$

$$KA = KA \cup \{k\}$$

$$Trip(k) = \emptyset$$

A condição $NTW(k) = \emptyset$ significa que ou $a_k > t_{nk}^0$ ou $t_{nk}^0 \Rightarrow b_k$ para todo navio $n \in N$. No primeiro caso, descartamos o trip k das futuras considerações, já que neste caso, nenhum navio $n \in N$ poderá atender à sua restrição time-window em qualquer estágio $l \geq 1$.

PASSO 0.2.2: $\forall n \in N$

Se $KTW(n) = \emptyset$ e se $a_k > t_{nk}^0, \forall k \in K$, faça:

$$N = N - \{n\}$$

A condição $KTW(n) = \emptyset$ significa que o navio $n \in N$ não

atende o time-window de nenhum trip $k \in K$, no início do schedule de n , e se além disso, $a_k > t_{nk}^0, \forall k \in K$, neste caso o navio n é descartado de futuras considerações.

PASSO 0.3: Para todo navio remanescente $n \in N$ e para todo trip remanescente $k \in K$, gerar os conjuntos $NU(1, k)$ e $KU(1, n)$ para $l = 1$.

$\forall k \in K, \forall n \in N$ gerar:

$NU(1, k)$

$KU(1, n)$

PASSO 0.3.1: Para cada $k \in K$, verificar:

Se $NU(1, k) = \emptyset$ faça:

$K = K - \{k\}$

$KA = KA \cup \{k\}$

$Trip(k) = \emptyset$

PASSO 0.3.2: Para cada $n \in N$, verificar:

Se $KU(1, n) = \emptyset$, faça:

$N = N - \{n\}$

PASSO 1: Primeiro Estágio ($l = 1$)

No primeiro estágio, designamos inicialmente um navio n para cada trip $k \in K$. Como o número de trips é normalmente maior que o número de navios da frota, para que a designação seja completa, acrescentamos ao conjunto N , se necessário, um conjunto de navios artificiais

$NA_1 = \{\text{conjunto de navios artificiais para atender os trips } k \in K\}$.

Na prática, NA_1 pode ser interpretado como um conjunto de navios afretados emergencialmente no mercado para atender os trips $k \in KU$ (i, \bar{n}) $\neq \emptyset$ que não foram ainda alocados para nenhum navio

$\bar{n} \in N$.

Assim, NA_1 é gerado de forma a satisfazer à seguinte desigualdade:

$$|N \cup NA_1| = |K|$$

Cada navio artificial $n \in NA_1$, é gerado de forma a atender as restrições de espaço e de tempo de um trip específico $k \in K$ e que ainda não esteja alocado a nenhum schedule de navios da frota.

Assim, a priori, nada garante que um navio $n \in NA_1$ tenha condições de atender a outros trips $k \in K$.

Uma segunda designação será feita para os trips $k \in KA$ (se $KA \neq \emptyset$).

Para isso, geramos um novo conjunto de navios artificiais denotado NA_2 , tal que: $|NA_2| = |KA|$; isto é, para cada navio $n \in NA_2$, será alocado um trip $k \in KA$.

A necessidade de gerar dois conjuntos de navios artificiais NA_1 e NA_2 é justificada da seguinte forma:

Os navios artificiais $n \in NA_1$ são alocados aos trips $k \in K$, que por sua vez podem ser alocados aos schedules de algum navio $n \in N$ em sintetização posteriores.

Enquanto isso, os navios $n \in NA_2$ são alocados a trips $k \in KA$, que pela sua construção, não podem ser designados a nenhum schedule de nenhum navio da frota $n \in N$.

Desta forma, os pares $(n \in NA_2, k \in KA)$ geram schedules simples (um único trip) para cada navio artificial $n \in NA_2$.

Isto significa que na solução final do problema existirão pelo menos $|NA_2|$ navios artificiais, fato este que não chega a nos preocupar pois na prática (veja Becker (1987)) dificilmente existirão trips $k \in K$ que não possa ser atendido por nenhum navio da frota no início de um planejamento, ou seja, o número $|KA|$ normalmente é nulo ou muito pouco significativo.

PASSO 11: Determinar para cada navio remanescente $n \in N$, o custo de alocar cada trip $k \in K \cup \{1, n\}$, no início do schedule do navio n . Os custos para os navios afretados são conhecidos antecipadamente. Assim, $\forall n \in N$ calcule:

$$\text{CUSTO}_{nk} = CO_{nk} + C_{nk}, \text{ com } k \in KU(1, n)$$

PASSO 12: Para cada navio remanescente $n \in N$ verificar a cardinalidade do conjunto $KU^*(1, n)$ onde $|KU^*(1, n)|$ é definido como $|KU^*(1, n)| = |KU(1, n)| -$ (o número de trips $k \in KU(1, n)$ com trip $(k) = \emptyset$).

Se $|KU^*(1, n)| = 1$, faça:

$$\text{SEQ}(n, 1) = \bar{k} \in KU(1, n)$$

$$\text{TA}(n) = \text{SEQ}(n, 1)$$

$$\text{TRIP}(\bar{k}) = \emptyset$$

$$\text{CMIN}_1(n) = \text{CUSTO}_{nk}$$

Se, $|KU^*(1, n)| > 1$, significa que existem vários trips $k \in KU(1, n)$ candidatos ao schedule do navio $n \in N$, neste caso escolha dentre os trips candidatos, aquele trip \bar{k} tal que:

$$t_{nk}^0 = \text{mínimo} \{ t_{nk}^0, k \in KU(1, n) \}$$

Isto equivale a calcular:

$KU^{(1)}(1, n) = \{ \bar{k} \in KU(1, n) \mid t_{nk}^0 = \text{mínimo } t_{nk}^0, \text{ com } k \in KU(1, n) \}$

Se $|KU^{(1)}(1, n)| = 1$, faça:

$CMIN1(n) = CUSTO_{n\bar{k}}$

$SEQ(n, 1) = \bar{k}$

$TA(n) = SEQ(n, 1)$

$TRIP(\bar{k}) = \emptyset$

retornar PASSO 1.2) com $n = n + 1$.

Se $|KU^{(1)}(1, n)| > 1$, escolhemos dentre os trips remanescentes $k \in KU^{(1)}(1, n)$ com $TRIP(k) \neq \emptyset$, aquele trip \bar{k} tal que:

$a\bar{k} = \text{mínimo } \{ak \mid k \in KU^{(1)}(1, n) \text{ e } TRIP(k) = k\}$

Isto é, gerar os conjuntos:

$KU^{(2)}(1, n) = \{ \bar{n} \in KU^{(1)}(1, n) \mid TRIP(\bar{k}) = \bar{k} \text{ e}$

$a\bar{k} = \text{mínimo } \{ak\} \}$.

Se $|KU^{(2)}(1, n)| = 1$, faça:

$CMIN1(n) = CUSTO_{n\bar{k}}$

$SEQ(n, 1) = \bar{k}$

$TA(n) = SEQ(n, 1)$

$TRIP(\bar{k}) = \emptyset$

retorne Passo 1.2) com $n = n + 1$.

Se $|KU^{(2)}(1, n)| > 1$, gerar o conjunto:

$KU^{(3)}(1, n) = \{\bar{k} \mid b_{\bar{k}} = \text{mínimo} \{b_j \mid j \in KU^{(2)}(1, n) \text{ e } \text{trip}(j) = j\}\}$.

Se $|KU^{(3)}(1, n)| = 1$, faça:

$$CMIN\ i(n) = CUSTO_{nk}^-$$

$$SEQ(n, 1) = \bar{k}$$

$$TA(n) = \bar{k}$$

$$TRIP(\bar{k}) = \emptyset$$

retorne Passo 1.2) com $n = n + 1$.

Se $|KU^{(3)}(1, n)| > 1$, gerar o conjunto:

$KU^{(4)}(1, n) = \{\bar{k} \mid U_{\bar{k}} = \text{mínimo} \{U_j \mid j \in KU^{(3)}(1, n) \text{ e } \text{trip}(j) = j\}\}$

Se $|KU^{(4)}(1, n)| = 1$, faça:

$$CMIN\ i(n) = CUSTO_{nk}^-$$

$$SEQ(n, 1) = \bar{k}$$

$$TA(n) = \bar{k}$$

$$TRIP(\bar{k}) = \emptyset$$

retorne PASSO 1.2) com $n = n + 1$.

Se $|KU^{(4)}(1, n)| > 1$, gerar o conjunto:

$KU^{(5)}(1, n) = \{k \mid CUSTO_{nk}^- = \min \{CUSTO_{nk}^- \mid k \in KU^{(4)}(1, n) \text{ e } TRIP(k) = k\}\}$

no caso de $|KU^{(5)}(1, n)| > 1$, escolha o primeiro \bar{k} da lista e faça:

$$CMIN\ i(n) = CUSTO_{nk}^-$$

$$SEQ(n, 1) = \bar{k}$$

$$TA(n) = \bar{k}$$

$$TRIP(\bar{k}) = \emptyset$$

retorne PASSO 1.2) com $n = n + 1$

PASSO 13: Se $|\bar{K}| > |N|$ (\bar{K} = conjunto K inicial)

Gerar o conjunto

$NA_1 = \{NNAVIO + 1, \dots, NNAVIO + p\}$ onde

$$p = |\bar{k}| - |N| - |KA|$$

Para cada navio $n \in NA_1$, alocar um trip $k \in K$ tal que

$$k \in \bigcup_{n \in N} SEQ(n, 1)$$

Isto é, faça para cada par:

$(n \in NA_1, k \in K - \bigcup_{n \in N} SEQ(n, 1))$ a alocação:

$$SEQ(n, 1) = k$$

$$TA(n) = k$$

$$CMINI(n) = CUSTO_{nk}$$

PASSO 14: Se $|KA| = \emptyset \longrightarrow$ ESTÁGIO (L+1)

Se $|KA| > \emptyset$, gerar um novo conjunto de navios artificiais.

$$NA_2 \text{ tal que } |NA_2| = |KA|$$

Alocar para cada trip $k \in KA$ um navio $n \in NA_2$

$$\text{SEQ}(n, 1) = k$$

$$\text{TRIP}(k) = \emptyset$$

$$\text{CMIN}_1(n) = \text{CUSTO}_{nk}$$

Obs: Nos passos 1.3) e 1.4) supomos conhecido os custos de alocar um navio $n \in NA_1$ a um trip $k \in K$ e $n \in NA_2$ a um trip $k \in KA$.

CONCLUSÕES DO PRIMEIRO ESTÁGIO ($l = 1$)

Neste estágio inicial ($l = 1$) resolvemos um problema de alocação linear. Veremos adiante que este tipo de problema é resolvido de forma completa somente uma única vez neste algoritmo.

Além da alocação linear de cada navio $n \in N \cup NA_1 \cup NA_2$ determinamos:

Os custos de cada alocação inicial ($CUSTO_{kn}$) ou $CMIN_1(n)$

O TRIP alocado ao schedule do navio n , neste estágio 1 $SEQ(n, 1)$

O TRIP incorporado mais recentemente ao schedule do navio n ($TA(n)$).

PASSO 2: ESTÁGIO ($l \geq 2$)

Nos estágios $l \geq 2$ são efetuadas sintetizações entre dois schedules.

Como a frota de navios é heterogênea, as sintetizações

devem sempre ser referidas a um determinado navio.

Assim, se K_i e K_j são dois trips de K e se

$$\text{SEQ}(n_i, 1) = K_i$$

$$\text{SEQ}(n_j, 1) = K_j \text{ com } i \neq j$$

Existem neste caso, as seguintes possibilidades entre os schedules de n_i e n_j :

i) Sintetização entre k_i e k_j é viável para o schedule de n_i .

ii) Sintetização entre k_i e k_j é viável para o schedule de n_j .

iii) Sintetização entre k_i e k_j é viável a ambos os schedules.

iv) Sintetização entre k_i e k_j não é viável a nenhum dos dois schedules.

Nos estágios $l \geq 2$, as sintetizações serão entre um schedule de navio $n \in N$ com um schedule de $n \in NA_1$.

Não serão consideradas sintetizações entre dois schedules de navios de N e nem sintetizações entre dois schedules de navios artificiais cujas justificativas foram colocadas na introdução deste capítulo.

Uma sintetização é dita viável para o navio $n \in N$ se ela atende às restrições de espaço e de tempo do navio n e dos trips envolvidos neste schedule.

ATUALIZAÇÃO DOS CONJUNTOS $KU(L, N)$

No início de cada estágio $l \geq 2$, devemos atualizar os conjuntos $KU(l, n) = \{ \text{conjunto de trips candidatos a serem alocados para os schedule do navio } n, \text{ no estágio } l \}$.

PASSO 2.1: Para cada navio remanescente $n \in N$, determinar

$$t_{nk}^0 = t_{nTA(n)}^0 + t_{nTA(n)} + t_{nTA(n)k}, \quad \forall k \in KU(l-1, n) \cup \bigcup_{n \in NA_1} SEQ(n, l-1)$$

tal que $TRIP(k) = k$

PASSO 2.2: Gerar para todo navio remanescente $n \in N$

$$KU(l, n) = \{ k \in KU(l-1, n) \cup \bigcup_{n \in NA_1} SEQ(n, l) \mid a_k \leq t_{nk}^0 \leq b_k, 1 \leq l \leq L \}$$

e, além disso, $t_{nk}^0 + t_{nk} \leq u_k$ e $TRIP(k) = K$.

Se $|KU(l, n)| = \emptyset$ e se $t_{nk}^0 > b_k$ para todo k

remanescente, faça: $N = N - \{n\}$.

Se $|KU(1, n)| > 0 \rightarrow$ PASSO 2.3

Obs: A condição $TRIF(k) = k$ ou $k \in \bigcup_{\substack{n \in N \\ 1 \leq i \leq l}} SEQ(n, i)$ evita

que ocorra uma sintetização entre schedules de navios $n \in N$. Neste caso, um dos navios de N seria obrigatoriamente descartado da solução final, o que na prática significa deixá-lo ocioso durante o atual horizonte de planejamento ainda que ele seja viável para atender alguns trips $k \in K$.

EXEMPLO: $KU(1, n_1) = \{k_1, k_3\}$ e $SEQ(n_1, 1) = \{k_1\}$

$KU(1, n_2) = \{k_3, k_4\}$ e $SEQ(n_2, 1) = \{k_3\}$

Uma sintetização entre os schedules $\{k_1\}$ e $\{k_3\}$ para o navio n_1 , elimina o navio n_2 de futuras considerações, descartando-os da solução final.

PASSO 2.3: As sintetizações nos estágios $l \geq 2$, serão somente para os schedules de navios $n \in N$ tais que $|KU(1, n)| > 0$.

TESTE DE PARADA:

Se $|KU(1, n)| = 0$ para todo $n \in N \Rightarrow$ (FIM)

Caso contrário, as sintetizações obedecem aos seguintes critérios:

Se $|KU(1, n)| = 1$, neste caso faça:

$$CMIN_1(n) = CMIN_1(n) + C_{nTA(n)\bar{k}} + C_{nk}^-, \quad \bar{k} \in KU(1, n)$$

$$SEQ(n, 1) = \bar{k} \in KU(1, n)$$

$$TRIP(\bar{k}) = \emptyset$$

$$TA(n) = \begin{cases} SEQ(n, 1) \text{ se } \sum_{i=1}^{l-1} SEQ(n, i) = \emptyset \\ \text{ou} \\ SEQ(n, \bar{j}) \text{ com } \bar{j} = \max\{1 \leq j < l \mid SEQ(n, j) \neq \emptyset\} \end{cases}$$

$$SEQ(n', 1) = \emptyset, \text{ onde } n' \in NA_1 \text{ com } SEQ(n', 1) = \bar{k}$$

Se $|KU(1, n)| > 1$, gerar a sequência de conjuntos $KU^{(j)}(1, n)$ $j = 1, 2, 3, 4, 5$ necessários como no PASSO 1.2) gerando os valores:

$$CMIN_1(n) = CMIN_1(n) + C_{nTA(n)\bar{k}} + C_{nk}^-$$

$$SEQ(n, 1) = \bar{k}$$

$$TA(n) = \begin{cases} SEQ(n, 1) \text{ se } \sum_{i=1}^{l-1} SEQ(n, i) = \emptyset \\ SEQ(n, \bar{j}) \text{ com } \bar{j} = \max\{1 \leq j < l \mid SEQ(n, j) \neq \emptyset\} \end{cases}$$

$$TRIP(\bar{k}) = \emptyset$$

$$SEQ(n', 1) = \emptyset, \text{ onde } n' \in NA_1 \text{ com } SEQ(n', 1) = \bar{k}$$

onde \bar{k} é o trip agregado ao schedule do navio k

neste estágio.

CONCLUSÕES DO ESTÁGIO $L \geq 2$

A solução dos estágios $l \geq 2$ é dado por:

$$\left[\begin{array}{l} \left[\begin{array}{l} \bigcup_{\substack{n \in N \\ 1 \leq i \leq l}} \text{SEQ}(n, i) \end{array} \right], \text{ para cada } n \in N \text{ e} \\ \text{SEQ}(n', 1), \text{ para cada } n' \in NA_1 \text{ remanescente} \\ \text{SEQ}(n'', 1), \text{ para cada } n'' \in NA_2 \end{array} \right]$$

O custo total até o momento é dado por:

$$\left[\begin{array}{l} \text{CUSTOT}(l) = \sum_{n \in N} \text{CMNI}_1(n) + \sum_{n \in NA_1} \sum_{k \in K} C_{nk} \text{SEQ}(n, 1) + \\ + \sum_{n \in NA_2} \sum_{k \in KA} C_{nk} \end{array} \right]$$

CRITÉRIO DE PARADA:

Se $\text{TRIP}(k) = 0 \quad \forall k \in K$ o procedimento chega ao seu final, caso contrário, faça $l = l + 1$ e retorne ao PASSO 2.1) do algoritmo.

EXEMPLO NUMÉRICO: Resolvemos, a seguir, um pequeno exemplo mostrando passo a passo o desenvolvimento do nosso algoritmo, com os valores iniciais gerados aleatoriamente.

Considere $k = \{1, 2, 3, 4\}$ um conjunto de quatro trips

e $N = \{1, 2\}$, um conjunto de dois navios com velocidade única.

PASSO 0: INICIALIZAÇÃO:

Faço $TRIP(k=1) = 1$

$TRIP(k=2) = 2$

$TRIP(k=3) = 3$

$TRIP(k=4) = 4$

PASSO 0.1: Gerar os conjuntos:

$N(k=1) = \{1, 2\}$

$K(n=1) = \{1, 2, 3, 4\}$

$N(k=2) = \{1, 2\}$

$K(n=2) = \{1, 2, 4\}$

$N(k=3) = \{1\}$

$N(k=4) = \{1, 2\}$

PASSO 0.1.1:

Temos $N(k) \neq \emptyset \forall k = 1, 2, 3, 4$

PASSO 0.1.2:

Temos $K(n) \neq \emptyset, \forall n = 1, 2$

PASSO 0.2: Gerar os conjuntos:

$NTW(k = 1) = \{1, 2\}$

$KTW(n = 1) = \{1, 2\}$

$NTW(k = 2) = \{1, 2\}$

$KTW(n = 2) = \{1, 2, 4\}$

$NTW(k = 3) = \emptyset$

$$NTW(k = 4) = \{ 2 \}$$

PASSO 0.2.1: Como $NTW(k = 3) = \emptyset$ e supondo que $a_n > t_{na}^0$, $\forall n = 1, 2$, neste caso faço:

$$K = K - \{3\} = \{1, 2, 4\}$$

$$KA = \{3\}$$

PASSO 0.2.2: Como $KTW(n) \neq \emptyset$, $\forall n = 1, 2$ prossigo,

PASSO 0.3: Gerar os conjuntos:

$$NU(1, K=1) = \{2\}$$

$$NU(1, K=2) = \{1, 2\}$$

$NU(1, K=3)$ não é analisado pois $K = \{1, 2, 4\}$

$$NU(1, K=4) = \{2\}$$

$$KU(1, n=1) = \{2\}$$

$$KU(1, n=2) = \{1, 2, 4\}$$

PASSO 0.3.1: Temos que $NU(1, k) \neq \emptyset$ para todo $K = 1, 2, 4$.

PASSO 0.3.2: Temos que $KU(1, n) \neq \emptyset$ para todo n remanescente isto é, $n = 1, 2$.

PASSO 1: PRIMEIRO ESTÁGIO: $1 = 1$:

PASSO 11: Devemos determinar para cada navio remanescente ($n=1, n=2$), o custo de alocar cada trip $k \in KU(1, n)$, no início de um schedule.

$$CUSTO_{n=1, k=2} = CO_{n=1, k=2} + C_{n=1, k=2}$$

$$CUSTO_{n=2, k=1} = CO_{n=2, k=1} + C_{n=2, k=1}$$

$$CUSTO_{n=2, k=2} = CO_{n=2, k=2} + C_{n=2, k=2}$$

$$CUSTO_{n=2, k=4} = CO_{n=2, k=4} + C_{n=2, k=4}$$

PASSO 12: Verificar a cardinalidade de cada conjunto $KU(1, n)$ atualmente e alocar para cada navio, seu primeiro trip.

Para $n = 1$, temos $KU(1 = 1, n = 1) = 2$ e $trip\ 2 = 2$, logo:

$|KU(1 = 1, n = 1)| = 1$, neste caso faço:

$SEQ(n = 1, l = 1) = trip\ k = 2$

$TA(n = 1) = trip\ k = 2$

$TRIP(k = 2) = \emptyset$

$CMINI(n=1) = CUSTO_{n=1, k=2}$

Para $n = 2$, temos que $KU(1 = 1, n = 2) = \{1, 2, 4\}$

e $TRIP(k = 1) = 1$

$TRIP(k = 2) = \emptyset$

$TRIP(k = 4) = 4$

logo $|KU(1 = 1, n = 2)| = 2$

Gerar o conjunto $KU(1 = 1, n = 2)$

Suponha que neste exemplo $KU^{(1)}(1 = 1, n = 2) = \{4\} \subset KU(1 = 1, n = 2)$

Como $|KU^{(1)}(1 = 1, n = 2)| = 1$ faço:

SEQ (n = 2, l = 1) = trip k = 4

TA(n = 2) = trip k = 4

TRIP(k = 4) = \emptyset

C MIN1(n = 2) = CUSTO_{n=2, k=4}

PASSO 13: Temos que $|\bar{K}| = 4$ e $|\bar{N}| = 2$ onde \bar{K} e \bar{N} são os conjuntos K e N originais.

Neste caso, geramos o conjunto:

$NA_1 = \{n = 3\}$, onde $|NA_1| = |\bar{K}| - |\bar{N}| - |KA| = 1$.

Efetuamos a seguinte alocação:

SEQ(n=3, l=1) = trip k = 1

TA(n=3) = trip k = 1

CMIN1(n=3) = (valor pré-determinado)

PASSO 14: Temos que $|KA| = 1 > \emptyset$

Geramos então o segundo conjunto de navios artificiais

NA_2 com $|NA_2| = |KA|$

$NA_2 = \{n = 4\}$

Efetuar a alocação:

SEQ(n=4, l=1) = trip k=3

TA(n=4) = trip k=3

CMIN1(n=4) = (custo pré-determinado)

$$\text{TRIP}(k=3) = \emptyset$$

PASSO 2 : ESTÁGIO (1 = 2)

PASSO 2.1 : Para $n=1$ como

$K \in K \cup (1=1, n=1) \cup \text{SEQ}(n=3, l=1)$ com trip $k=k$ é dado por (1), neste caso só calculamos o valor de:

$$t_{n=1, k=1}^0 = t_{n=1, \text{TA}(n=1)}^0 + t_{n=1, \text{TA}(n=1)} + t_{n=1, \text{TA}(n=1), k=1}$$

Analogamente, como $k \in KU(1=1, n=2) \cup \text{SEQ}(n=3, l)$ com trip $k=k$ é igual a $(1, 2, 4) - (2, 4) = 1$, só calculamos o valor de:

$$t_{n=2, k=1}^0 = t_{n=2, \text{TA}(n=2)}^0 + t_{n=2, \text{TA}(n=2)} + t_{n=2, \text{TA}(n=2), k=1}$$

PASSO 2.2 :

$$\text{Considere } KU(n=1, l=2) = \emptyset$$

$$KU(n=2, l=2) = \{1\}$$

PASSO 2.3: como $|KU(n=1, l=2)| = \emptyset$ só consideramos $n=2$ a partir desse momento.

$$\text{Temos que } |KU(n=2, l=2)| = \{1\}.$$

Neste caso fazemos:

$$\text{SEQ}(n=2, l=2) = \text{trip } 1$$

$$\text{TA}(n=2) = \text{SEQ}(n=2, l=1) = \text{trip } 4$$

$$\text{TRIP}(k=1) = \emptyset$$

$$\text{SEQ}(n=3, 1) = \emptyset$$

CRITÉRIO DE PARADA

Como $\text{TRIP}(k) = \emptyset \forall k = 1, 2, 3, 4$, o procedimento chega ao seu final, com a seguinte solução:

SCHEDULES FINAIS DE CADA NAVIO

NAVIO	ESTAGIO 1	ESTAGIO 2
n=1	TRIP k=2	\emptyset
n=2	TRIP k=4	TRIP k=1
n=3 (*)	\emptyset	\emptyset
n=4 (*)	TRIP k=3	\emptyset

(*) navios afretados no mercado.

TABELA 5-1

CUSTO GLOBAL FINAL

$$\begin{aligned} \text{CUSTO T} &= \sum_{n=1}^2 \text{CMIN}_1(n) + \sum_{n=3}^4 \text{CMIN}_1(n) \text{SEQ}(n,1) = \\ &= \text{CMIN}_1(1) + \text{CMIN}_1(2) + \text{CMIN}_1(4). \end{aligned}$$

CAPÍTULO VII

RELAÇÕES ENTRE DUAS FORMULAÇÕES DISTINTAS DO PROBLEMA (FP)

INTRODUÇÃO

Apresentamos aqui, relações entre duas formulações para o problema de roteamento e scheduling de navios petroleiros em longo curso, na presença de restrições do tipo *time-windows*.

A primeira formulação é do tipo *set-covering* com restrições adicionais (SCA), e a outra é uma formulação que engloba todo o conjunto de restrições numa única fase, denotada por (F2M).

O objetivo aqui é o de mostrar que o limite inferior obtido ao utilizar uma relaxação lagrangeana convencional a um conjunto de restrições de (F2M), relaxação esta denotada por (RLF2M), é sempre igual ou superior ao limite obtido pelo problema (SCA) com relaxação linear (SCAL).

Isto é, mostramos que valem as seguintes desigualdades:

$$V(\text{SCAL}) \leq V(\text{RLF2M}) \leq V(\text{SCA}) = V(\text{F2M})$$

VII.D FORMULAÇÕES MATEMÁTICAS DE (SCA) E (F2M)

Ambas as formulações (SCA) e (F2M) foram desenvolvidas no capítulo III, onde podem ser encontradas maiores explicações e suas respectivas notações, que são suprimidas neste capítulo.

FORMULAÇÃO (SCA)

O problema (SCA) é descrito da seguinte forma:

$$V(\text{SCA}) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (7.1)$$

$$\text{S.A.} \quad \sum_{s \in S} e_{ks} x_s \geq 1, \quad \forall k \in K \quad (7.2)$$

$$\sum_{s \in S(n)} x_s \leq 1, \quad \forall n \in N \quad (7.3)$$

$$x_s \in (0, 1), \quad \forall s \in S \quad (7.4)$$

e onde $(e_{ks}) = (0, 1), \quad \forall k \in K, \quad \forall s \in S$

Vamos chamar de (SCAL), o problema (SCA) com relaxação linear.

FORMULAÇÃO (F2M)

A formulação (F2M) é uma versão modificada da formulação (F2) apresentada no capítulo III.

Em (F2M) supomos que cada demanda possa ser atendida por um único navio, numa única visita, e sua formulação é dada por:

$$V(F2M) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in PK(i)} c_{nvi} x_{nvi} + \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} y_{nvi} \quad (7.5)$$

$$S.A. \sum_{n \in N_i} \sum_{v \in V} y_{nvi} \geq 1, \forall i \in K \quad (7.6)$$

$$\sum_{n \in N} \sum_{v \in V} y_{nvi} \leq 1, \forall i \in K \quad (7.7)$$

$$\sum_{i \in AK(j)} \sum_{v \in V} x_{nvi} = \sum_{v \in V} y_{nvj}, \forall j \in K, \forall n \in N \quad (7.8)$$

$$t_{ni}^0 + t_{nvi} + t_{nv'ij} - t_{nj}^0 \leq (2 - \sum_{v \in V} (y_{nvj} - x_{nvi})) M_{nvv'ij} \quad (7.9)$$

$$\text{onde } M_{nvv'ij} \geq b_i + t_{nvi} + t_{nv'ij} - a_j, \\ \forall n \in N, \forall v, v' \in V, \forall i \in K, \forall j \in PK(i)$$

$$U_j \geq t_{nj}^0 + t_{nvj} - (1 - \sum_{v \in V} y_{nvj}) N_{nvj}, \quad (7.10)$$

$$\text{onde } N_{nvj} \geq b_j + t_{nvj}, \forall n \in N, \forall v \in V, \forall j \in K$$

$$a_j \leq t_{nj}^0 \leq b_j, \quad \forall n \in N, \quad \forall j \in K \quad (7.11)$$

$$x_{nvi j} \in (0,1), y_{nvj} \in (0,1), \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K, \quad \forall j \in K \quad (7.12)$$

Na formulação (F2M) temos que:

i) as restrições (7.6) e (7.7) juntas resultam na igualdade

$$\sum_{n \in N_i} \sum_{v \in V} y_{nvi} = 1, \quad \forall i \in K, \quad \text{mas separadas propositalmente}$$

para usarmos uma relaxação lagrangeana em (7.6).

ii) nas restrições (7.8) fica implícito (por (7.7)) que

$$\sum_{v \in V} y_{nvj} \leq 1, \quad \forall j \in K, \quad \forall n \in N.$$

O PROBLEMA (F2M) RELAXADO

Se relaxarmos as restrições (7.6) de (F2M), obtemos o seguinte problema lagrangeano, denotado por $(RLF2M)_\mu$ com multiplicadores $\mu_i \geq 0, \quad \forall i \in K$.

$$\begin{aligned} V(RLF2M)_\mu = \text{Mínimo} & \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in K} c_{nvi j} x_{nvi j} + \\ & + \sum_{n \in N} \sum_{v \in V} \sum_{j \in K} c_{nvj} y_{nvj} - \sum_{i \in K} \mu_i \\ & \left(1 - \sum_{n \in N_i} \sum_{v \in V} y_{nvi} \right) \end{aligned}$$

S.A. (7.7) - (7.12)

com $\mu = (\mu_i) \geq 0, \forall i \in K$

Estamos interessados em obter multiplicadores $\mu_i^* \geq 0, i \in K$, tais que:

$$V(\text{RLF2M}) = V(\text{RLF2M}\mu^*) = \underset{\mu \geq 0}{\text{Máximo}} \{V(\text{RLF2M}\mu)\} \quad (7.13)$$

O problema (RLF2M) acima não é linear, mas pode ser facilmente resolvido através de métodos do tipo subgradiente.

TEOREMA 7.1: Se $V(\text{SCAL})$ é o valor ótimo do problema (SCAL) e $V(\text{RLF2M})$ o do problema (RLF2M), então valem as seguintes desigualdades a seguir:

$$V(\text{SCAL}) \leq V(\text{RLF2M}) \leq V(\text{SCA}) = V(\text{F2M})$$

Prova: Naturalmente se trabalharmos com uma única formulação (SCAL) ou (RLF2M), o resultado acima vale por um resultado obtido por *Geoffrion* (1974) que mostra que as relaxações lagrangeanas fornecem limites iguais ou melhores que a obtida via relaxação linear.

No nosso caso a comparação é feita entre duas formulações distintas (SCAL) e (RLF2M).

Podemos supor sem perda de generalidade, que em (SCA) são analisadas todas as colunas viáveis do problema a menor das redundantes e/ou dominadas (veja formulação 1, cap. III).

Além disso, como $V(\text{SCA})$ e $V(\text{F2M})$ são valores ótimos do problema (FP), as relações $V(\text{RLF2M}) \leq V(\text{SCA}) = V(\text{F2M})$ valem naturalmente.

Portanto basta mostrarmos que $V(\text{SCAL}) \leq V(\text{RLF2M})$.

Para cada navio $n \in N$ e para cada velocidade $v \in V$, vamos definir $V(n, v) = \{ y_{nvi}^s / s \in S \}$ que representa uma família de soluções viáveis de (RLF2M).

Em particular, tomemos como sendo igual a $|S|$, a cardinalidade de cada família $V(n, s)$ onde S é o conjunto dos schedules viáveis gerados em (SCA).

Vamos chamar as soluções y_{nvi}^s de (7.6) como combinações lineares convexas de y_{nvi}^s , para cada $n \in N$, $v \in V$ e $i \in K$, isto é:

$$Y_{nvi} = \sum_{s \in S} \lambda_s y_{nvi}^s, \quad \forall i \in K, \quad \forall n \in N, \quad \forall v \in V \quad (7.14)$$

onde $y_{nvi}^s = \begin{cases} 1, & \text{se o navio } n \in N, \text{ efetua o trip } i, \text{ na} \\ & \text{velocidade } v, \text{ na solução } s \\ \emptyset, & \text{c/c} \end{cases}$

Substituindo a igualdade (7.14) em (7.5) e (7.6),
obtemos:

$$V(\text{RLF2M}) = \text{Mínimo} \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in PK(i)} c_{nvi} X_{nvi} + \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} \left(\sum_{s \in S} \lambda_s y_{nvi}^s \right) \quad (7.15)$$

$$\text{S.A.} \sum_{n \in N_i} \sum_{v \in V} \left(\sum_{s \in S} \lambda_s y_{nvi}^s \right) \geq 1, \quad \forall i \in K \quad (7.16)$$

$$\sum_{s \in S} \lambda_s = 1, \text{ com } \lambda_s \geq 0, \quad (7.17)$$

Definindo $\sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in PK(i)} c_{nvi} X_{nvi} = F$ (const. em λ)

$$\sum_{n \in N} \sum_{v \in V} \sum_{i \in K} c_{nvi} y_{nvi}^s = c_s \quad e$$

$$\sum_{n \in N_i} \sum_{v \in V} y_{nvi}^s = d_{i_s}, \quad \forall i \in K, \quad \forall s \in S$$

Podemos reescrever (7.15) - (7.17) na forma:

$$V(\text{RLF2M}) = \text{Mínimo} \sum_{s \in S} c_s \lambda_s \quad (7.18)$$

$$\text{S.A.} \sum_{s \in S} d_{i_s} \lambda_s \geq 1, \quad \forall i \in K \quad (7.19)$$

$$\sum_{s \in S} \lambda_s = 1 \quad (7.20)$$

$$0 \leq \lambda_s \leq 1, \quad \forall s \in S \quad (7.21)$$

OBSERVAÇÕES:

i) No conjunto de restrições (7.19) temos que

$$d_{is} = \sum_{n \in N_i} \sum_{v \in V} y_{nvi}^s, \quad \forall i \in K, \quad \forall s \in S, \quad \text{onde cada } y_{nvi}^s \text{ satisfaz às condições (7.7) e (7.12).}$$

Portanto, podemos concluir que: $\forall i \in K, \forall s \in S$

$$d_{is} = \begin{cases} 1 \\ \text{ou} \\ 0 \end{cases}$$

Além disso, pela definição dos y_{nvi}^s

$$d_{is} = \begin{cases} 1, & \text{se o trip } i \text{ está contido no schedule } s \in S \\ 0, & \text{c/c} \end{cases}$$

de onde se conclui que $d_{is} = e_{is}, \forall i \in K, \forall s \in S$, onde (e_{is}) é a matriz das restrições set covering de (SCA).

ii) Em (7.20), $S = \bigcup_{n \in N} S(n)$ onde cada $S(n)$ representa o conjunto de schedules (colunas) viáveis do navio $n \in N$.

$$\text{Assim, } S(n) \subset S, \quad \forall n \in N \text{ w } \sum_{s \in S(n)} \lambda_s \leq 1, \quad \forall n \in N \quad (7.20')$$

Portanto, substituindo as restrições (7.20) por (7.20') estamos relaxando o problema (RLF2M) descrito por (7.18) - (7.20), permitindo que numa solução viável do problema, possam existir mais de um schedule, o que é mais que razoável.

Definindo de (R) o problema (7.18), (7.19), (7.20') e (7.21), teremos que:

$$V(R) = \text{Mínimo} \cdot \sum_{s \in S} c_s \lambda_s \quad (7.22)$$

$$\text{S.A.} \sum_{s \in S} d_{is} \lambda_s \geq 1, \forall i \in K \quad (7.23)$$

$$\sum_{s \in S(n)} \lambda_s \leq 1, \forall n \in N \quad (7.24)$$

$$0 \leq \lambda_s \leq 1, \forall s \in S \quad (7.25)$$

Observe que uma solução ótima do problema descrito por (7.18) - (7.21) é composto de um único schedule (coluna) s^* que naturalmente deve conter todos os trips $s \in S$. Ou seja, a solução ótima é do tipo $\lambda_{s^*} = 1, \lambda_s = 0, \forall s \in S, s \neq s^*$ que claramente satisfaz às restrições (7.23) - (7.25) de R.

Portanto, uma solução ótima de (7.18) - (7.21) é uma solução viável de (R), donde se conclui que R é uma relaxação do problema (RLF2M) descrito por (7.18) - (7.21) ou que $V(\text{RLF2M}) \geq V(R)$.

Mas note que (R) é exatamente o problema (SCAL), logo $V(\text{RLF2M}) \geq V(\text{SCAL})$ ou

$$\boxed{V(\text{SCAL}) \leq V(\text{RLF2M}) \leq V(\text{SCA}) = V(\text{F2M})} \quad (7.26)$$

A igualdade $V(\text{SCAL}) = V(\text{RLF2M})$ ocorre na situação em que um único schedule atende a todos os trips do problema (solução de (7.18) - (7.21)).

Apresentamos, a seguir, um exemplo numérico mostrando a desigualdade $V(\text{SCAL}) \leq V(\text{RLF2M})$.

EXEMPLO: Supomos $|V| = 1$, $|N| = 2$, $|K| = 4$, $|S| = 4$

Considere: as $|S| = 4$ soluções de (7.7) a (7.12) como sendo:

SOLUÇÃO $\tilde{S} = 1$

$$\begin{array}{ll} y_{111}^1 = 1 & y_{211}^1 = 0 \\ y_{112}^1 = 0 & y_{212}^1 = 1 \\ y_{113}^1 = 1 & y_{213}^1 = 0 \\ y_{114}^1 = 0 & y_{214}^1 = 1 \end{array}$$

SOLUÇÃO $\tilde{S} = 2$

$$\begin{array}{ll} y_{111}^2 = 1 & y_{211}^2 = 0 \\ y_{112}^2 = 1 & y_{212}^2 = 0 \end{array}$$

$$y_{113}^2 = 0$$

$$y_{114}^2 = 0$$

$$y_{213}^2 = 1$$

$$y_{214}^2 = 1$$

SOLUÇÃO $s = 3$

$$y_{111}^3 = 0$$

$$y_{112}^3 = 1$$

$$y_{113}^3 = 0$$

$$y_{114}^3 = 0$$

$$y_{211}^3 = 1$$

$$y_{212}^3 = 0$$

$$y_{213}^3 = 1$$

$$y_{214}^3 = 1$$

SOLUÇÃO $s = 4$

$$y_{111}^4 = 0$$

$$y_{112}^4 = 1$$

$$y_{113}^4 = 1$$

$$y_{114}^4 = 0$$

$$y_{211}^4 = 1$$

$$y_{212}^4 = 0$$

$$y_{213}^4 = 0$$

$$y_{214}^4 = 0$$

esta solução não atende (7.6)

As soluções de (7.6) são dadas por:

$$y_{nvi} = \sum_{s \in S} \lambda_s y_{nvi}^s, \quad \forall n \in N, \quad \forall v \in V, \quad \forall i \in K$$

Logo teremos o seguinte sistema de equações:

$$y_{111} = \lambda_1 + \lambda_2$$

$$y_{112} = \lambda_2 + \lambda_3 + \lambda_4$$

$$y_{113} = \lambda_1 + \lambda_4$$

$$y_{114} = 0$$

$$y_{211} = \lambda_3 + \lambda_4$$

$$y_{212} = \lambda_1$$

$$y_{213} = \lambda_2 + \lambda_3$$

$$y_{214} = \lambda_1 + \lambda_2 + \lambda_3$$

onde $\lambda_i \geq 0$, $i = 1, 2, 3, 4$ e $\sum_{i=1}^4 \lambda_i = 1$ (**)

Por outro lado, as restrições (7.6) exigem que:

$$y_{111} + y_{211} \geq 1 \Rightarrow \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \geq 1 \text{ (redundante por (**))}$$

$$y_{112} + y_{212} \geq 1 \Rightarrow \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \geq 1 \text{ (redundante por (**))}$$

$$y_{113} + y_{213} \geq 1 \Rightarrow \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \geq 1 \text{ (redundante por (**))}$$

$$y_{114} + y_{214} \geq 1 \Rightarrow \lambda_1 + \lambda_2 + \lambda_3 \geq 1 \Rightarrow$$

$$\lambda_4 = 0 \text{ e } \lambda_1 + \lambda_2 + \lambda_3 = 1$$

Logo por (*) $y_{214} = 1$

Como o sistema (*) possui 7 equações e 4 incógnitas (λ_i), existem 3 variáveis livres aos quais arbitramos os valores viáveis: $\lambda_1 = \lambda_3 = 0$ e $\lambda_2 = 1$

Note que com esta escolha, estamos supondo que o schedule * contém todos os trips do problema.

Para estes valores particulares dos λ_i teremos:

$$y_{111} = 1$$

$$y_{211} = 0$$

$$y_{112} = 1$$

$$y_{212} = 0$$

$$\begin{array}{ll}
 y_{113} = 0 & y_{213} = 1 \\
 y_{114} = 0 & y_{214} = 1
 \end{array}$$

Tomando os custos $c_{nvi} = 1$ e $c_{nvi j} = 1$, $\forall n \in N$, $\forall i \in K$, $\forall j \in K$ e $\forall v \in V$ teremos:

$$\begin{aligned}
 V(\text{RLF2M}) &= P + \sum_n \sum_v \sum_i c_{nvi} y_{nvi} = P + 4 \text{ onde} \\
 P &= \sum_n \sum_v \sum_i \sum_j x_{nvi j} \text{ (constante em } y)
 \end{aligned}$$

Mas de igualdade (7.8) de (F2M) podemos determinar o valor de P , a partir dos valores dos y_{nvi} , usando o seguinte sistema linear.

$$(7.8) \quad \sum_{i \in K} \sum_{j \in K} x_{nvi j} = \sum_{v \in V} y_{nvj}, \quad \forall n \in N, \quad \forall j \in K \text{ ou}$$

$$\left. \begin{array}{l}
 1 = y_{111} = x_{1111} + x_{1121} + x_{1131} + x_{1141} \\
 1 = y_{112} = x_{1112} + x_{1122} + x_{1132} + x_{1142} \\
 0 = y_{113} = x_{1113} + x_{1123} + x_{1133} + x_{1143} \\
 0 = y_{114} = x_{1114} + x_{1124} + x_{1134} + x_{1144} \\
 0 + y_{211} = x_{2111} + x_{2121} + x_{2131} + x_{2141} \\
 0 = y_{212} = x_{2112} + x_{2122} + x_{2132} + x_{2142} \\
 1 = y_{213} = x_{2113} + x_{2123} + x_{2133} + x_{2143} \\
 1 = y_{214} = x_{2114} + x_{2124} + x_{2134} + x_{2144}
 \end{array} \right\} D$$

Observe que não é necessário resolver o sistema de equações lineares (D) para obter o valor de P , basta somarmos as equações de (D), ou seja:

$$P = \sum_{n \in N} \sum_{v \in V} \sum_{i \in K} \sum_{j \in K} x_{nvi j} = 4$$

Portanto $V(\text{RLF2M}) = P + 4 = 8$

Determinamos agora o valor de $V(\text{SCAL}) =$
 $= c_1 \lambda_1 + c_2 \lambda_2 + c_3 \lambda_3 + c_4 \lambda_4$

Na nossa escolha particular dos λ , teremos $V(\text{SCAL}) =$
 $= c_2 \lambda_2 = c_2$

DETERMINAÇÃO DO VALOR DE (C2) À PARTIR DOS VALORES DE C_{nvi} E $C_{nvi j}$

Na matriz *set covering de CR* = (SCAL)

Temos que $d_{i_s} = \sum_{n \in N_i} \sum_{v \in V} y_{nvi}^s = e_{i_s} \quad \forall i \in K, \forall s \in S$

$s = 1$

$$d_{11} = y_{111}^1 + y_{211}^1 = 1$$

$$d_{21} = y_{112}^1 + y_{212}^1 = 1$$

$$d_{31} = y_{113}^1 + y_{213}^1 = 1$$

$$d_{41} = y_{114}^1 + y_{214}^1 = 1$$

$s = 2$	$s = 3$	$s = 4$
$d_{12} = 1$	$d_{13} = 1$	$d_{14} = 1$
$d_{22} = 1$	$d_{23} = 1$	$d_{24} = 1$
$d_{32} = 1$	$d_{33} = 1$	$d_{34} = 1$
$d_{42} = 1$	$d_{43} = 1$	$d_{44} = 0$

	λ_1	λ_2	λ_3	λ_4
trip 1	1	1	1	1
trip 2	1	1	1	1
trip 3	1	1	1	1
trip 4	1	1	1	0

matriz das restrições

set covering de $(t_{i_s}) = (e_{i_s})$

(P) = (SCA)

Pela matriz de *set covering* acima, podemos determinar os custos c_s , a partir dos custos c_{nvi} e $c_{nvi j}$ relacionados com o schedule $s \in S$.

De fato, para $s = 2$, teremos o caso em que o schedule contém todos os trips do problema, assim:

$$c_2 \approx \sum c_{nvi} + (\text{custos das viagens ociosas}) = 4 + VD$$

Logo $c_2 \leq 4 + 4 = 8$, pois $VD =$ (soma de três $c_{nvi j}$ que por hipótese são todos iguais a 1) mais o custo inicial para o navio aportar na origem do trip inicial a 1 pois também é uma viagem em lastro ($c_{nvi j}$).

Portanto $c = 8$ e $V(SCA) = 8$ se o navio (único pois

existe um único schedule na solução) no início do planejamento não está na origem do trip inicial.

No caso particular onde não ocorre o caso acima, teremos $V(SCAL) = 7 < V(RLF2M)$.

Assim, no caso geral

$$V(SCAL) \leq 8 = V(RLF2M)$$

CAPITULO VIII

SOLUÇÃO VIA TÉCNICAS EXATAS:

UM NOVO ESQUEMA UTILIZANDO GERAÇÃO DE COLUNAS

E PROGRAMAÇÃO DINÂMICA

VIII.1 - INTRODUÇÃO

Apresentamos uma generalização de uma técnica desenvolvida por *DESROSIERS, SOUMIS e DESROLHERS (1983)* para a solução de um problema de roteamento de veículos com restrições do tipo "*time-windows*", frota homogênea, uma origem, único produto e sem restrições de capacidade.

A nossa proposta é dirigida ao problema aplicativo (FP) que é uma generalização dos problemas clássicos de roteamento de veículos encontrados na literatura pelo fato de ser caracterizado por uma frota heterogênea de navios, restrições de capacidade dos navios, restrições de tempo limite para efetuar cada trip múltiplas origens, múltiplos produtos para cada navio, além das restrições do tipo "*time-windows*".

O esquema de solução que propomos se baseia em reescrever uma formulação inicial (F7) como um problema set-covering com restrições adicionais (SCA) (ambos desenvolvidos no capítulo III) e resolvê-lo via simplex com um novo esquema de geração de colunas.

O subproblema de geração de colunas é descrito como uma generalização do problema de caminho mínimo com restrições de *scheduling* (CMS) que pode ser decomposto em $|N|$ subproblemas menores e independentes, de caminho mínimo, um

para cada navio n , isto é, $(CMS(n))$, onde $n = 1, 2, \dots, |N|$.

Recorde que num problema de geração de colunas, devemos buscar apenas um schedule não básico de um navio da frota heterogênea.

Assim, se $V(CMS)$ é o valor ótimo de (CMS) e $V(CMS(n))$ o de $(CMS(n))$, então $V(CMS)$ é dado pelo menor valor dos $V(CMS(n))$, com $n \in N$. Isto é,

$$V(CMS) = \underset{n \in N}{\text{Mínimo}} \{V(CMS(n))\}$$

Finalmente, a nossa proposta é a de resolver cada subproblema $(CMS(n))$ com as restrições de integralidade via programação dinâmica, adaptando o algoritmo de *Ford e Bellman* ao nosso problema, generalizando uma proposta anterior de *Desrosiers, Soumis e Desrosiers (1983)*.

VIII.2 - FORMULAÇÃO MATEMÁTICA

Considere inicialmente a formulação (F7) desenvolvida no capítulo III.

Vimos que em (F7) são efetuadas algumas simplificações em relação a formulações mais gerais (por exemplo a formulação (F2)), como a de supor velocidade única para os navios e que cada trip do atual horizonte de planejamento

possa ser atendido por um único navio da frota, numa única visita.

Devido a pequenas modificações aqui sugeridas, vamos reescrever de uma forma compacta, as notações de (F7) e (SCA) desenvolvidas no capítulo III deste trabalho.

NOTAÇÃO DE (F7)

N = conjunto de navios heterogêneos da frota

K = conjunto de trips do atual horizonte de planejamento

KAP = subconjunto de trips de horizontes de planejamento anteriores formado pelos trips mais recentemente efetuados por cada navio $n \in N$ (origem de cada schedule) adicionado com um trip artificial t (destino comum de todos os schedules) (veja figura 7.1).

$\bar{K} = K \cup KAP$

N_i = conjunto de navios $n \in N$ que atendem às restrições locais R (veja notação de F7, capítulo III) do problema onde $n \in N_i$ implica que n atende à demanda do trip i , numa única visita.

$C_{ni j}$ = custo operacional do navio $n \in N$, efetuar o trecho $(i,$

j) em

lastro mais o custo do navio n efetuar o trip $j \in K$.

$$C_{njt} = 1, \forall n \in N, \forall j \in K$$

U_j = tempo limite para efetuar o trip $j \in K$

t_{ni} = tempo para o navio $n \in N$, efetuar o trip $i \in K$

t_{nij} = tempo para o navio $n \in N$, percorrer em lastro, o trecho (i, j) com $i \in \bar{K}$, $j \in K$

t_{njt} = (constante comum), $\forall n \in N, \forall j \in K$.

t_{ni}^o = variável contínua de tempo necessário para o navio n aportar na origem do trip $i \in K$, pronto para operar.

X_{nij} = variável binária zero-um =
$$\begin{cases} 1, & \text{se } n \text{ percorre } (i, j) \\ 0, & \text{c/c} \end{cases}$$

$[a_i, b_i]$ = *time-window* do trip $i \in K$.

TRIPS ENVOLVIDOS NO ATUAL PLANEJAMENTO

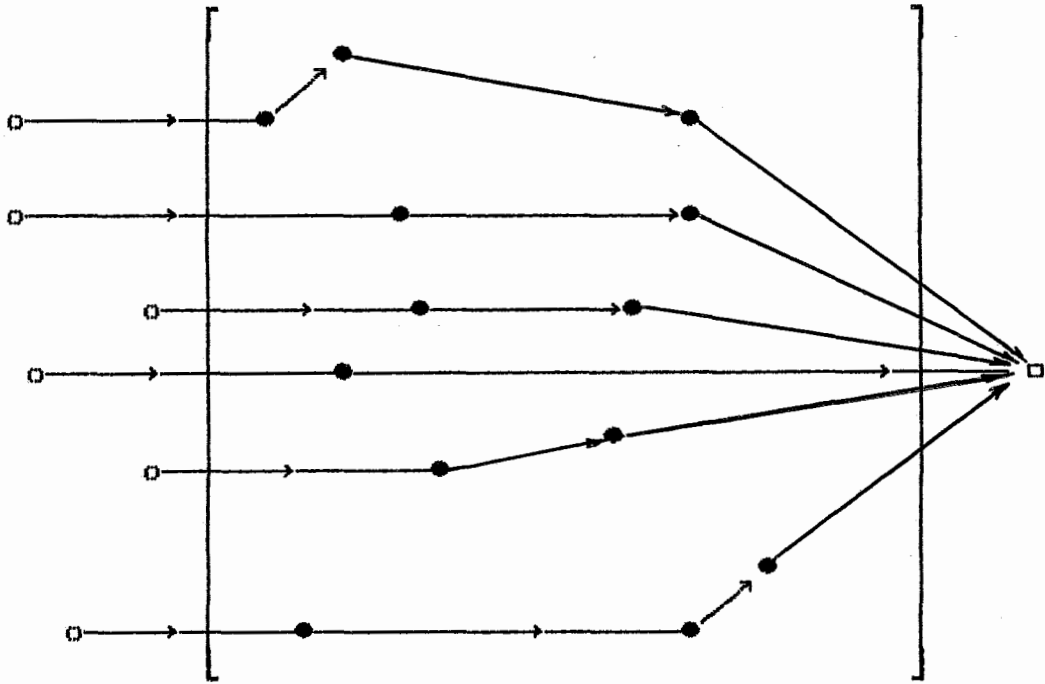


FIGURA 8.1

● = trips do atual horizonte de planejamento (K)

o = trips de horizontes anteriores (KAP)

□ = trip artificial, comum a todos os trips (KAP)

(→) = schedule do navio n_i

Exemplo de um conjunto de trips $\bar{K} = (K) \cup (KAP)$

com $|N| = 6$ navios

FORMULAÇÃO MATEMÁTICA (F7)

$$V(F7) = \text{Mínimo} \sum_{n \in N} \sum_{i \in \bar{K}} \sum_{j \in \bar{K}} c_{nij} x_{nij} \quad (8.1)$$

$$S.A. \sum_{i \in \bar{K}} x_{nij} = 1, \quad \forall n \in N, \quad \forall j \in \bar{K} \quad (8.2)$$

$$\sum_{i \in \bar{K}} x_{nji} = 1, \quad \forall n \in N, \quad \forall j \in \bar{K} \quad (8.3)$$

$$x_{nij} \geq 0, \quad \forall n \in N, \quad \forall i \in \bar{K}, \quad \forall j \in \bar{K} \quad (8.4)$$

$$t_{ni}^0 + t_{ni} + t_{nij} - t_{nj}^0 \leq (1 - x_{nij}) M_{nij} \quad (8.5)$$

$$\text{onde } M_{nij} \geq b_i + t_{ni} + t_{nij} - a_j, \quad \forall n \in N, \quad \forall i \in \bar{K}, \quad \forall j \in \bar{K}$$

$$U_j \geq t_{nj} + t_{nj} (1 - \sum_{i \in \bar{K}} x_{nij}) N_{nj}, \quad (8.6)$$

$$\text{onde } N_{nj} \geq b_j + t_{nj}, \quad \forall n \in N, \quad \forall j \in \bar{K}$$

$$a_i \leq t_{ni} \leq b_i, \quad \forall n \in N, \quad \forall i \in \bar{K} \quad (8.7)$$

$$x_{nij} \in \{0, 1\}, \quad \forall n \in N, \quad \forall i \in \bar{K}, \quad \forall j \in \bar{K} \quad (8.8)$$

As restrições (8.1) a (8.4) descrevem um problema de roteamento puro, conhecido como $|N|$ - *problemas de fluxo de custo mínimo*, com a propriedade de integralidade.

As restrições de alocação linear (8.2) e (8.3) nos dizem que cada trip $j \in \bar{K}$ deve ser atendido numa única visita

e por um único navio.

O conjunto de restrições (8.5) a (8.7) formam o problema de *scheduling*, sendo que (8.5) requer que se $X_{nij} = 1$, então todas as restrições de tempo correspondentes ao arco (i, j) devem ser atendidos.

Já (8.6) exige que se n chega a origem do trip $j \in K$, então n deve atender também o limite de tempo existente para concluir o trip.

As restrições (8.7) são os *time-windows* dos trips e finalmente (8.8) são variáveis binárias do problema.

Acrescentando as restrições de *scheduling* ao problema de roteamento puro (8.1) a (8.4), perde-se a propriedade de integralidade do problema, mas é possível mostrar que (F7) sem as restrições (8.8) admite ao menos uma solução ótima inteira.

De fato, seja (x^*, t^*) uma solução ótima de (F7) excluindo (8.8). Se x^* possuir as componentes todas inteiras, a demonstração está completa, caso contrário, fixamos o vetor t^* e considere a sub-rede formado somente pelos valores positivos da solução x^* .

A otimização nesta sub-rede com t^* fixo, equivale a resolver um problema de fluxo de custo mínimo $(F7t^*)$ que

possui a propriedade de integralidade.

Considere (x^{t*}, T^*) a solução ótima de $(F7t^*)$.

Vamos mostrar que (X^{t*}, T^*) é também uma solução ótima de $(F7)$.

Seja $V(F7t^*)$ é o valor ótimo de $(F7t^*)$, como X^* é viável em $(F7t^*)$, então $V(F7) \geq V(F7t^*)$.

Por outro lado, $(F7t^*)$ é o problema $(F7)$ com restrições adicionais, ou seja, o conjunto viável de $(F7t^*)$ está contido na região viável de $(F7)$, logo $V(F7t^*) \geq V(F7)$.

Portanto $V(F7t^*) = V(F7)$, ou seja, (X^{t*}, T^*) é também uma solução ótima inteira de $(F7)$.

VIII.3 - LITERATURA

Dentre os vários trabalhos existentes na literatura para resolver um problema do tipo $(F7)$ (ou casos particularers), podemos citar os seguintes:

LEVIN (1966) discretiza as variáveis contínuas de tempo

num problema de planejamento de vôos de aviões.

ORLOFF (1974) resolve um problema do tipo *set partitioning*, gerando explicitamente numa etapa inicial, todas as rotas viáveis. Os resultados não foram satisfatórios devido ao elevado número de colunas (rotas).

FISCHER e JAIKUMAR (1981) Propõe uma relaxação lagrangeana para obter um *bound* para o método *branch-and-bound*, num problema de alocação linear generalizado.

SWERSEY e BALLARD (1982) usam o algoritmo simplex para resolver um problema de percurso de ônibus escolares.

DESROSIERS, SOUMIS e DESROCHERS (1983) propõe uma formulação *set-partitioning* resolvido via geração de colunas. O sub-problema é um problema de caminho mínimo com restrição *time-windows* resolvido via algoritmo de *Ford e Bellman* para caminho mínimo adaptado.

A técnica de *Desrosiers, Soumis e Desrochers (1983)* se mostrou muito eficiente, principalmente quando no subproblema de geração de colunas, cada veículo se ocupa de poucos trips, ou ainda quando os *time-windows* conseguem reduzir bastante o número de rotas viáveis (Por exemplo, *time-windows* pequenos).

O nosso objetivo foi o de generalizar o procedimento usado por *Desrosiers, Soumis e Desrochers (1983)*, adaptando a técnica de geração de colunas e Programação Dinâmica a um problema de roteamento e *scheduling* com frota heterogênea de navios, múltiplos depósitos, múltiplos produtos e restrições do tipo *time-windows*.

Aqui, a formulação inicial (F7) é reescrita como um problema *set covering com restrições adicionais (SCA)* e este, por sua vez, resolvido via simplex, relaxando as restrições de integralidade.

Propomos em (SCA), um esquema de geração de colunas onde o subproblema é descrito como no máximo $|N|$ problemas de caminho mínimo com restrições de *scheduling*.

Estes subproblemas são resolvidos com as restrições de integralidade, usando generalizações de algoritmos de caminho mínimo desenvolvidos por *Ford e Bellman e Desrosiers, Soumis e Desrochers (1983)*.

O PROBLEMA SET COVERING COM RESTRICÇÕES ADICIONAIS

A formulação (F7) pode ser reformulada particionando o conjunto de restrições em duas etapas.

Na primeira, analisamos as restrições locais do problema e com o objetivo de gerar todos os *schedules* viáveis do problema, a menor das redundantes e/ou dominadas.

Na etapa seguinte, busca-se dentre os *schedules* gerados, um subconjunto, que juntos cubram todos os trips do problema a um menor custo global possível, respeitando as restrições do problema.

O problema desta última etapa pode ser descrito como um problema *set covering com restrições adicionais (SCA)* como desenvolvido no capítulo III, e reescrito resumidamente a seguir:

$$V(SCA) = \text{Mínimo} \sum_{s \in S} c_s x_s \quad (8.9)$$

$$\text{S.A.} \quad \sum e_{ks} x_s \geq 1, \quad \forall k \in K \quad (8.10)$$

$$\sum_{s \in S(n)} x_s \leq 1, \quad \forall n \in N \quad (8.11)$$

$$x_s \in \{0, 1\}, \quad \forall s \in S \quad (8.12)$$

O conjunto de restrições (8.10) nos diz que cada trip $i \in K$ deve estar contido em pelo menos um *schedule* $s \in S$.

As restrições adicionais (8.11) requerem que cada navio $n \in N$ deve efetuar no máximo um *schedule* no atual horizonte de planejamento e finalmente (8.12) são as variáveis binárias do problema relacionada com a decisão de usar ou não o *schedule* $s \in S$.

Uma solução ótima de (SCA) é formada por um conjunto de *schedules* de custo mínimo atendendo às restrições de roteamento e de *scheduling* do problema.

Resolvemos inicialmente o problema (SCA) com relaxação linear via algoritmo simplex e posteriormente, se necessário, um método do tipo *branch-and-bound* para obter uma solução ótima inteira.

Devido ao elevado número de *schedules* (colunas) existente ($|S| \gg 0$), utilizamos um procedimento de geração de colunas à cada iteração do simplex como mostramos a seguir:

VIII.4 - O SUBPROBLEMA DE GERAÇÃO DE COLUNAS

O problema aplicativo (FP), consiste de uma frota heterogênea de navios.

Na literatura, um *schedule* de um veículo consiste de uma sequência ordenada no tempo e espaço de trips. Além disso, na maioria dos casos, os *schedules* iniciam e terminam num único nó (depósito), comum a todos os veículos.

Contudo no nosso problema (FP), existem vários depósitos iniciais e os navios não retornam necessariamente ao ponto inicial após cumprir a última tarefa.

Assim, para enquadrar o problema (FP) no caso tradicional, adicionamos um trip artificial t que representa o nó final, comum a todos os *schedules*.

Para isso, definimos custos $C_{njt} = 0$, e tempos de chegada em t , $t_{njt} = 0$, $\forall n \in N$, $\forall j \in K$. Além disso, em (FP), existem múltiplas origens, mais especificamente uma origem $s(n)$ para cada navio $n \in N$.

Cada $s(n)$ na prática representa o trip efetuado mais recentemente pelo navio $n \in N$, em horizontes de planejamentos, ou a atual localização do navio $n \in N$.

Para colocar o nosso subproblema de geração de colunas

como um problema clássico de caminho mínimo, geramos um segundo trip artificial s , que representa a origem comum de todos os navios da frota com custos $C_{nsj} = 0$ e tempos $t_{nsj} = 0$, $\forall n \in N$, $\forall j \in (KAP)$.

Com isso, o problema de geração de colunas passa a ser um problema de caminho mínimo de s até t , respeitando as restrições de roteamento e de *scheduling* do problema.

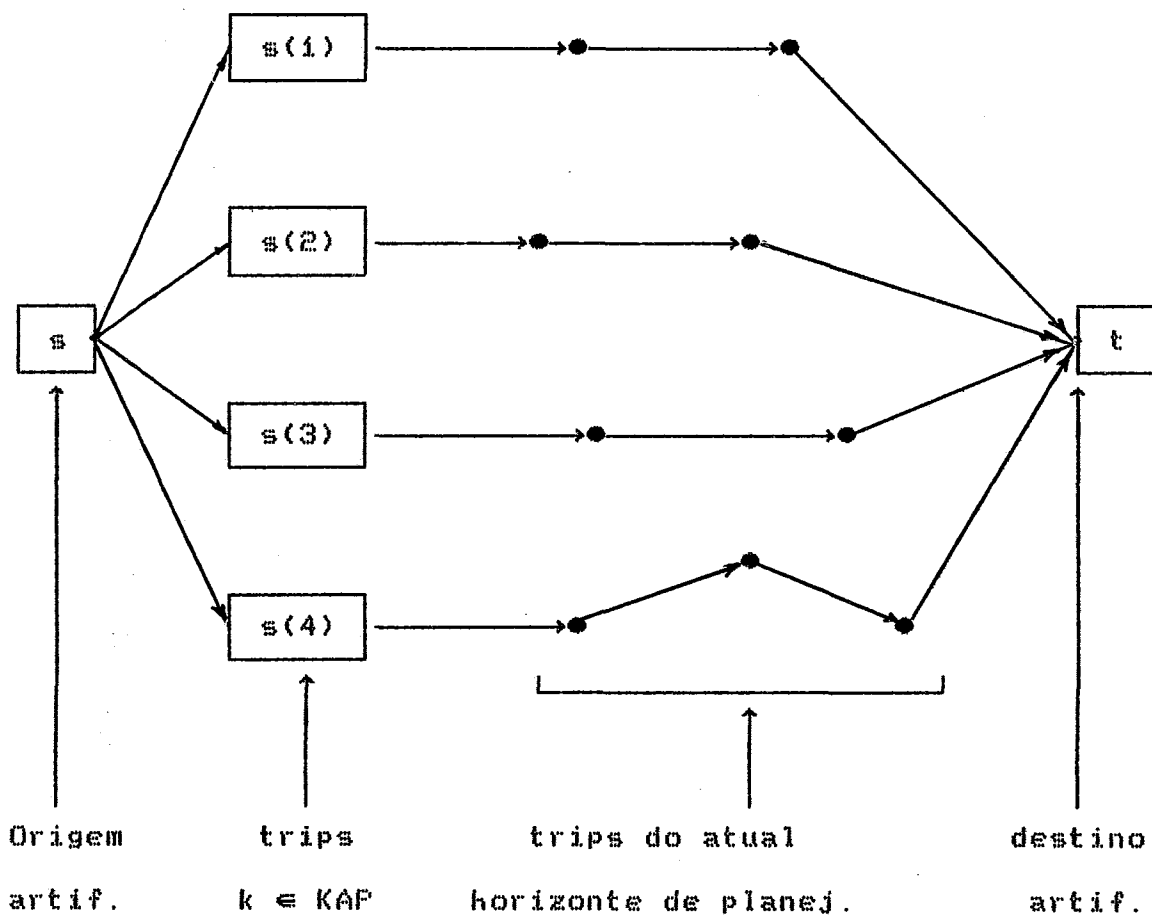


FIGURA 8.2

FORMULAÇÃO MATEMÁTICA DO SUBPROBLEMA DE GERAÇÃO DE COLUNAS DE (SCAL)

O objetivo do subproblema de geração de colunas de (SCAL), denotado por (CMS) "*caminho mínimo com restrições de scheduling*", é a de gerar um *schedule* de menor custo relativo, iniciando em s e terminando em t , satisfazendo as restrições de roteamento e *scheduling* do problema.

Sejam Z_{ni} as variáveis duais das restrições de (SCAL) desta forma, o problema (CMS) pode ser representado do seguinte modo:

$$\text{Minimizar } \sum_{n \in N} \sum_{i \in \bar{K}} \sum_{j \in \bar{K}} (c_{nij} - z_{ni}) x_{nij} \quad (8.13)$$

$$\text{S.A. } \sum_{i \in \bar{K}} x_{nij} = \sum_{i \in \bar{K}} x_{ji}, \quad \forall j \in \bar{K}^{(*)}, \quad \forall n \in N \quad (8.14)$$

$$\sum_{n \in N_i} \sum_{j \in \bar{K}} x_{nsj} = \sum_{n \in N_i} \sum_{j \in \bar{K}} x_{njt} = 1 \quad (8.15)$$

$$x_{nij} \geq 0, \quad \forall n \in N, \quad \forall i \in \bar{K}, \quad \forall j \in \bar{K} \quad (8.16)$$

$$t_{ni}^0 + t_{ni} + t_{nij} - t_{nj}^0 \leq (1 - x_{nij}) M_{nij}, \quad \text{onde}$$

$$M_{nij} \geq b_i + t_{ni} + t_{nij} - a_j, \quad \forall n \in N, \quad \forall i \in \bar{K}, \quad \forall j \in \bar{K} \quad (8.17)$$

$$U_j \geq t_{nj}^0 + t_{nj} (1 - \sum_{i \in \bar{K}} x_{nij}) N_{nj}, \quad \text{onde } N_{nj} \geq b_j + t_{nj} \quad (8.18)$$

para todo $n \in N$, e todo $j \in K$

$$a_i \leq t_{ni}^0 \leq b_i, \quad \forall n \in N, \quad \forall i \in K \quad (8.19)$$

$$X_{nij} \in (0, 1), \quad \forall n \in N, \quad \forall i \in K, \quad \forall j \in K \quad (8.20)$$

ANÁLISE DA FORMULAÇÃO (CMS)

As restrições (8.13) a (8.16) de (CMS) formam um problema de caminho mínimo de s até t para uma frota heterogênea de navios, que possui a propriedade de integração.

Por outro lado, ao acrescentarmos as restrições de *scheduling* (8.17) a (8.19), a propriedade de integralidade é perdida, mas como visto em (F7), podemos ainda *garantir a existência de uma solução ótima inteira para (CMSL)*, onde (CMSL) é o problema (CMS) com relaxação linear.

Propomos resolver o problema de geração de coluna resolvendo até $|N|$ subproblemas menores e independentes de caminho mínimo de s até t , mas com uma *frota homogênea de navios*. Com isso, eliminamos um índice das variáveis de fluxo x_{nij} , passando a trabalhar com variáveis do tipo x_{ij} , reduzindo consideravelmente as dimensões dos subproblemas.

A técnica de particionar (CMS) em $|N|$ subproblemas independentes e homogêneos (CMS (n)) é justificada da seguinte forma: cada navio $n \in N$ possui um subconjunto $S(n)$ de colunas da matriz de (SCA). A cada iteração do simplex em (SCA), escolhemos uma coluna dentre $S = \bigcup_{n \in N} S(n)$ para entrar na nova base.

Portanto, ao invés de escolher a coluna de menor custo relativo no conjunto global S , podemos escolher inicialmente a coluna de menor custo relativo $s(n)$ no subconjunto $S(n) \subset S$.

De fato, o problema (SCA) requer somente uma coluna (*schedule*) que melhore o valor da sua função objetivo, isto é, basta encontrar uma *coluna de custo relativo negativo*.

Este objetivo pode ser alcançado resolvendo apenas um subproblema (CMS (n=1)) desde que tenhamos $V(\text{CMS}(n=1)) < 0$, neste caso, a solução ótima deste subproblema dado por um *schedule* $s(n=1)$ entra na nova base do simplex em (SCA).

Caso contrário, se $V(\text{CMS}(n=1)) \geq 0$, partimos para resolver o segundo subproblema (CMS (n=2)) e assim sucessivamente. Isto equivale a resolver no pior caso, $|N|$ subproblemas independentes de caminho mínimo de s até t com frota homogênea de navios.

Caso desejemos determinar a cada iteração do simplex, o

menor custo relativo dos schedules não-básicos, resolvemos os $|N|$ subproblemas (CMS (n)) e determinamos $V(\text{CMS } (N^*)) = \text{Mínimo } \{V(\text{CMS } (n)), n \in N\}$.

Se $V(\text{CMS}(n^*)) \geq 0$, a atual solução do simplex é a ótima, caso contrário, escolhemos a solução ótima $s(n^*)$ de CMS (n*) para entrar na nova base de (SCA).

Finalmente uma nota sobre a escolha de \bar{K} ao invés de K , nas restrições (8.14) de (CMS), provocando com isso, redundâncias em (8.15).

Tal escolha foi feita para que implicitamente (CMS) atendesse às desigualdades ($\sum_{i \in \bar{K}} x_{nij} \leq 1$) usadas para melhor relacionar o procedimento de geração de colunas com uma decomposição do tipo *dantzig-wolfe* feito adiante.

FORMULAÇÃO MATEMÁTICA DOS SUBPROBLEMAS HOMOGÊNEOS DE GERAÇÃO DE COLUNAS CMS(N)

Cada subproblema com frota homogênea CMS(n), $n \in N$ é descrito de forma semelhante ao problema (CMS) definido por (8.13) a (8.20). As alterações estão no número de índices das variáveis, que são reduzidas para dois e

consequentemente, reduzindo o número de variáveis e de restrições do problema.

$$V(\text{CMS}(n)) = \text{mínimo} \sum_{i \in K} \sum_{j \in K} (c_{ij} - z_i) X_{ij} \quad (8.21)$$

$$\text{S.A.} \quad \sum_{i \in K} x_{ij} = \sum_{i \in K} x_{ji}, \quad \forall j \in K \quad (8.22)$$

$$\sum_{j \in K} x_{=j} = \sum_{j \in K} x_{jt} = 1 \quad (8.23)$$

$$x_{ij} \geq 0, \quad \forall i \in K, \quad \forall j \in K \quad (8.24)$$

$$t_i^0 + t_i + t_{ij} - t_j^0 \leq (1 - X_{ij}) M_{ij}, \quad \text{com } M_{ij} \geq b_i + t_i + t_{ij} - a_j \quad (8.25)$$

$$\forall i \in K, \quad \forall j \in K$$

$$U_j \geq t_j^0 + t_j (1 - \sum_{i \in K} x_{ij}) N_j, \quad \text{com } N_j \geq b_j + t_j, \quad \forall j \in K \quad (8.26)$$

$$a_j \leq t_j^0 \leq b_j, \quad \forall j \in K \quad (8.27)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in K, \quad \forall j \in K$$

Na realidade no subproblema CMS(n) descrito por (8.21) a (8.28), cada variável X depende de n, isto é, $X_{ij} = X_{ij}(n) \in N$, analogamente para as variáveis $t_i^0 = t_i^0(n)$, bem como as constantes $c_{ij} = c_{ij}(n)$, $t_i = t_i(n)$ e $t_{ij} = t_{ij}(n)$, $U_j = U_j(n)$, $a_j = a_j(n)$, $b_j = b_j(n)$.

Ao substituírmos (CMS) pelos subproblemas (CMS(n)), $n \in N$, obtemos uma significativa redução nas dimensões do problema de geração de colunas, como mostra a tabela a seguir:

	número de restrições de roteamento	número de restrições de scheduling	número de variáveis X	número de variáveis Y
CMS	$ N * K + 1$	$2 * N * (K ^2 + K)$	$ N * K ^2$	$ N * K $
CMS(n)	$ K + 1$	$2(K ^2 + K)$	$ K ^2$	$ K $

Exemplo com $|N|$ navios, $|K|$ trips

Tabela (7.1)

VIII.5 - A SOLUÇÃO DOS SUBPROBLEMAS DE GERAÇÃO CMS(N)

A técnica de solução proposta para os subproblemas de geração de colunas (CMS(n)), $n \in N$ é uma generalização da proposta feita por *Desrochers, Soumis e Desrochers (1983)* para um problema de roteamento com frota homogênea e única origem.

Este procedimento, adaptado ao problema com frota heterogênea, múltiplas origens e restrições de tempo limite gera ao invés de um subproblema de caminho mínimo, até $|N|$

subproblemas de caminho mínimo de s até t, onde cada subproblema está relacionado com um determinado navio de frota.

Através de informações sobre a atual solução básica viável do simplex aplicada ao problema (SCA), podemos ordenar, a cada iteração do simplex, os $|N|$ sub-problemas (CMS(n)) do melhor ao pior navio candidato a mudar de *schedule* de uma iteração para outra.

Feita esta ordenação, o primeiro subproblema (CMS(n₁)) da lista é resolvido e caso ela forneça um valor ótimo negativo, isto significa que *já foi encontrado um schedule não básico candidato a ingressar na nova base do simplex*, neste caso, retorna-se ao problema principal com esta coluna.

Caso contrário, se $V(\text{CMS}(n_1)) \geq 0$, resolvemos o segundo subproblema da lista e assim sucessivamente até chegarmos a uma iteração onde necessariamente teremos que resolver os $|N|$ subproblemas, obtendo $V(\text{CMS}(n_m)) \geq 0$, neste caso, a solução atual do simplex é a solução ótima de (SCA).

Observe que a técnica de geração de colunas aqui proposta é exatamente uma decomposição do tipo *Dantzig-wolfe* do problema (F7), onde as restrições originais de *scheduling* e as restrições de conservação de fluxo são alocados ao subproblema (CMS) e a restrição de que todo trip deve ser

atendido ($\sum_{i \in K} x_{nij} = 1, \forall n \in N, \forall j \in K$) alocado ao problema mestre.

Mas recorde que na análise do subproblema (CMS), salientamos que ela satisfaz implicitamente restrições do tipo $\sum_{i \in K} x_{nij} \leq 1$ (*). Assim, no problema mestre de Dantzig-Wolfe, do problema (F7), podemos substituir, sem perda de generalidade, as restrições (8.2) dadas por $\sum_{i \in K} x_{nij} = 1$ pelas desigualdades $\sum_{i \in K} x_{nij} \geq 1$ (**), já que (*) e (**) reunidas, equivalem à igualdade (8.2).

Portanto, através de um procedimento usado no capítulo VI, podemos mostrar que o problema (SCA) é uma relaxação do problema mestre da decomposição de Dantzig-Wolfe para o problema (F7) no caso geral, e equivalente no caso particular onde em (SCA) existe apenas um schedule na solução ótima.

Ou seja, no caso geral, os problemas mestre (SCA) e o subproblema (CMS) são equivalentes a uma decomposição Dantzig-Wolfe de uma relaxação do problema (F7).

Os problemas de geração de colunas (CMS(n)) são resolvidos aqui, usando uma generalização do algoritmo proposto por Desrosiers, Soumis e Desrochers (1983) que por sua vez, adaptaram o algoritmo de caminho mínimo de Ford e Bellman.

O algoritmo adaptado proposto por *Desrosiers, Soumis, Desrochers (1983)* foi definido como o "problema de caminho mínimo com restrições 'time-windows'", e pode ser assim resumido:

No problema clássico de caminho mínimo, o algoritmo de *Ford e Bellman* armazena para cada nó j da rede um valor π_j que representa o caminho mínimo da origem s até o nó j .

Neste algoritmo, o princípio de otimalidade é descrito como "se r é um caminho de custo mínimo de s até j e se (i, j) é o seu arco terminal, então o caminho parcial de s até i é também um caminho de custo mínimo."

Desrosiers, Soumis e Desrochers (1983) desenvolveram um generalização deste algoritmo para um problema de roteamento com restrições "time-windows". Neste caso, cada *schedule* é comparado em termos de seus custos e também dos seus tempos de chegada a um nó j .

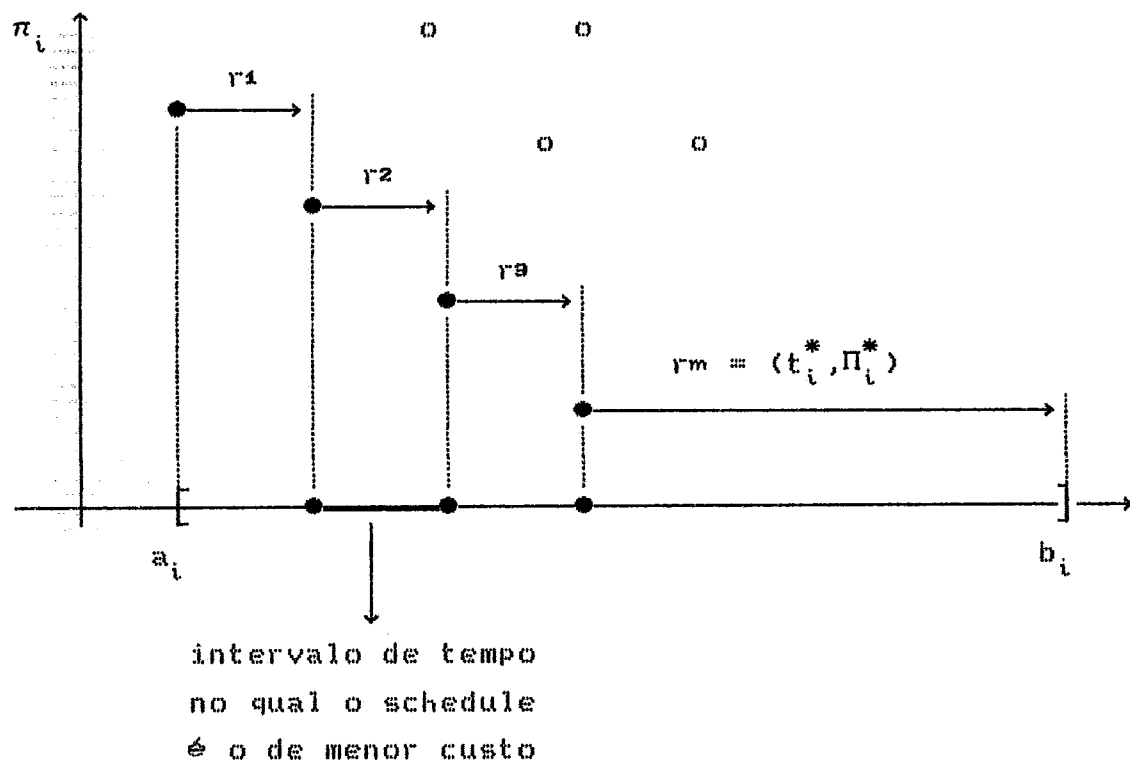
O princípio de otimalidade passa a ser então descrito na forma:

Se r é um *schedule* de custo mínimo de s até j , chegando em j no tempo t_j ou antes e se (i, j) é seu arco terminal, então o *schedule* parcial de s até i também um *schedule* de custo mínimo, chegando em i no tempo $t_j - t_{i,j}$ ou antes".

O algoritmo armazena para cada nó, pares de valores (t_i, π_i) que são otimizados até que seja encontrado o par (t_i^*, π_i^*) de menor custo, satisfazendo às restrições de *scheduling*.

Se $S_i = \{(t_i^\lambda, \pi_i^\lambda) \mid i \leq \lambda \leq m\}$ representa o conjunto de pares (t_i, π_i) candidatos a $(t_i^{(*)}, \pi_i^{(*)})$, excluindo os pares dominados, podemos então ordenar os componentes de S_i numa ordem decrescente de custos e crescente nos tempos t_i , ou vice-versa, na Figura (8.3) mostrada a seguir.

Os valores de $(t_i^\lambda, \pi_i^\lambda)$ são enquadrados no plano representado pelo intervalo do *time-window* do trip i , $[a_i, b_i]$ o eixo π_i de custos.



onde o = schedules dominados

FIGURA 8.3

As listas subsequentes L_j , formado por *schedules* que contém o nó i , são constituídos à partir das informações dos L_i e dos valores correspondentes ao arco (i, j) .

Finalmente, a nossa proposta é generalizar o algoritmo proposto por *Desrosiers, Soumis, Desrochers (1983)*, acrescentando aos valores armazenados em cada trip, os *tempos limites para concluir o trip*.

Portanto, para cada trip i , armazenamos três valores $\pi = (c_i, t_{oi}, t_{ei})$ onde c_i = custo mínimo de se locomover de s

até i , t_{oi} é o tempo para percorrer o trecho entre s até a origem de i , e t_{oe} o tempo para ir de s até i , incluindo o tempo para efetuar o trip i .

Assim, para cada trip i , são gerados uma sequência de aproximação sucessivas parciais $\pi_i^{(m)} = (c_i^{(m)}, t_{oi}^{(m)})$ e aproximação finais $\pi_i^{(m)} = (c_i^{(m)}, t_{oi}^{(m)}, t_{ei}^{(m)})$ que convergem para $\pi_i = (c_i, t_{oi}, t_{ei})$ (veja figura 8.3) adiante.

Cada $\pi_i = (c_i^{(m)}, t_{oi}^{(m)}, t_{ei}^{(m)})$ representa um *schedule* de s até i , passando por no máximo m arcos *intertrips*.

O princípio de otimalidade, adaptado ao nosso algoritmo de caminho mínimo (CMS) passa a ser da seguinte forma:

"se r é um *schedule* de custo mínimo, da origem s até o trip j , chegando na origem de j , no tempo t_j^0 ou antes e concluindo (*) o trip j , no tempo global $t_e j$ ou antes, e se (i, j) é o seu arco *intertrip* terminal, então o *schedule* parcial de r da origem s até o trip i é também um *schedule* de custo mínimo, dentre todos os *schedules* de custo mínimo, dentre todos os *schedules* que chegam à origem de i no tempo $t_j^0 - t_{ij}$ ou antes e concluem (*) e no tempo $t_{ej} - t_{ij}$ ou antes".

(*) o tempo para concluir um trip é medido a partir da origem s , isto é, $t_{ei} = t_{oi} + (\text{tempo para efetuar trip } i)$.

ALGORITMO DE CAMINHO MÍNIMO PARA O PROBLEMA (CMS(N))

PASSO 0: INICIALIZAÇÃO

Faça $\pi_{s_0}^{(1)} = (0, 0)$

$\pi_{s_0}^{(1)} = (0, 0, 0)$

$K = K - \{s_0\}$

$K^{(m)} = \emptyset, \forall m = 1, 2, \dots, |N| - 1$

PASSO 1: APROXIMAÇÃO $M = 1$

Selecionar todos os arcos *intertrips* candidatos a serem o primeiro trip do schedule de custo mínimo.

P1.1) Para cada $j \in K$ faça:

$$\pi_{j \leftarrow s_0}^{(m=1)} = (C_j^{(1)}; T\theta_j^{(1)}) = \begin{cases} (cso_j, tso_j) & \text{se } tso_j \in [a_j, b_j] \\ \text{neste caso faça } K^{(1)} = K^{(1)} \cup \{j\} \\ \text{ou} \\ (\emptyset, \emptyset) & \text{caso contrário} \end{cases}$$

P1.2) Para cada $j \in K^{(1)}$ faça:

$$TE_j^{(1)} = T\theta_j^{(1)} + t_j$$

$$C_j^{(1)} = \bar{C}_j^{(1)} + C$$

$$\pi^{(1)} = \begin{cases} (C_j^{(1)}, T\theta_j^{(1)}, TE_j^{(1)}) & \text{se } TE_j^{(1)} \leq U_j \text{ ou} \\ (\theta\theta, \theta\theta, \theta\theta) \text{ c/c, neste \u00faltimo caso fa\u00e7a: } K = K - \{j\} \end{cases}$$

Para cada $j \in K - K^{(1)}$ fa\u00e7a:

$$\pi_j^{(1)} = (C_j^{(1)}, T\theta_j^{(1)}, TE_j^{(1)}) = (\theta\theta, \theta\theta, \theta\theta)$$

PASSO 2: (M + 1) - \u00c9SIMA APROXIMA\u00c7\u00c3O SUCESSIVA DE Π_j

P2.1) Para cada $j \in K$, fa\u00e7a:

$$\pi_j^{(m+1)} = \begin{cases} (\bar{C}_j^{(m+1)}, T\theta_j^{(m+1)}) = \{(\min \{C_j^{(m)}, \min_{k \neq j} \{c_k^{(m)} + c_{kj}\}\}, \\ \min \{TE_j^{(m)}, \min_{k \neq j} \{TE_k^{(m)} + t_{kj}\}\}) \\ \text{se } TE_k^{(m)} + t_{kj} \in [a_j, b_j], \text{ neste caso} \\ \text{fa\u00e7a: } K^{(m+1)} = K^{(m+1)} \cup \{j\} \\ \text{ou fa\u00e7a} \\ (\theta\theta, \theta\theta) \text{ c/c} \end{cases}$$

P2.2) Para cada $j \in K^{m+1}$, fa\u00e7a:

$$\begin{cases} (C_j^{(m+1)}, T\theta_j^{(m+1)}, TE_j^{(m+1)}) = (\bar{C}_j^{(m+1)} + c_j, \bar{T}\theta_j^{(m+1)}, \\ T\theta_j^{(m+1)} + t_j) \text{ se } TE_j^{(m+1)} \leq U_j \end{cases}$$

$$\pi_j^{(m+1)} = \begin{cases} \text{ou} \\ (\emptyset\emptyset, \emptyset\emptyset, \emptyset\emptyset) \end{cases}, \text{ caso contrário, neste último caso faça:}$$

$K = K - \{j\}$, neste caso, retornar ao passo P2.1) com o mesmo m e com o k atualizado até um arco intertrip terminal viável ligando o trip j ou até que $m = |N| - 2$

Para cada $j \in K - K^{(m+1)}$ faça:

$$\pi_j^{(m+1)} = (\emptyset\emptyset, \emptyset\emptyset, \emptyset\emptyset)$$

PASSO 3: Se $m = |N| - 1$ (FIM)

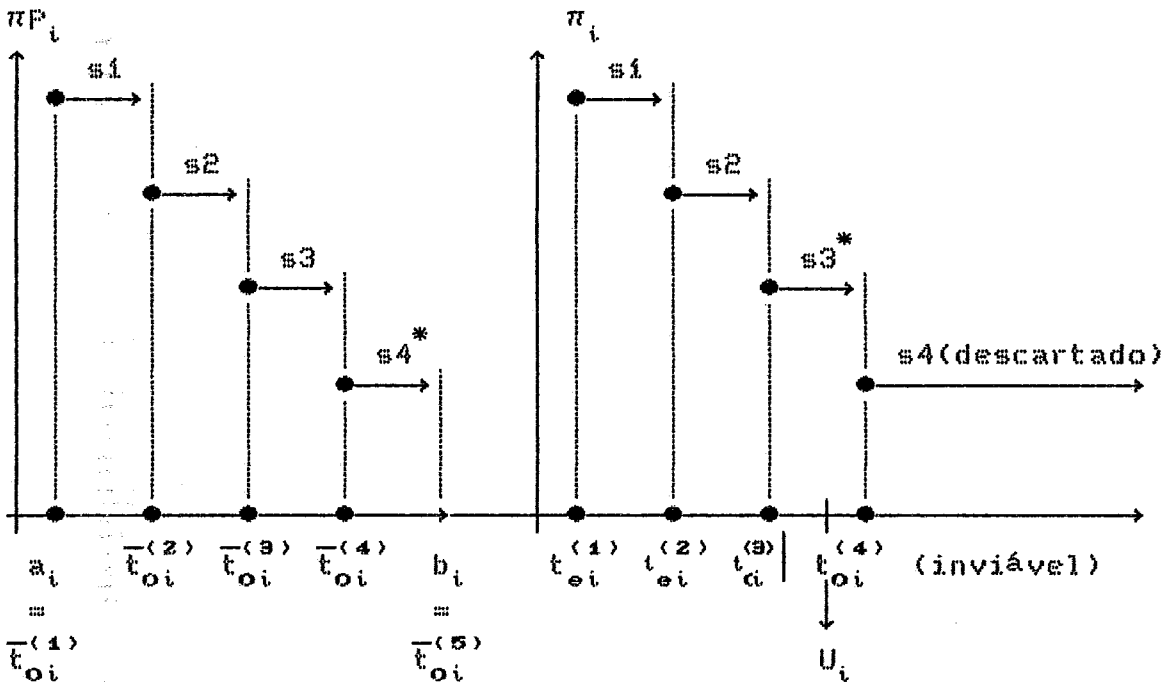
Caso contrário, retorne ao **PASSO 2**, com $m = m + 1$

ANÁLISE DO ALGORITMO

As aproximações sucessivas $\pi P_j^{(m)}$ e $\pi_j^{(m)}$, mostrados na figura (8.3) a seguir indicam sequências (πP_j^m) e (π_j^m) decrescentes em termos de custos e crescentes nos tempos; onde $\pi P_j^{(m)}$ determina o arco terminal ligado ao trip j na m -ésima aproximação e $\pi_j^{(m)}$ determina se o arco obtido em $\pi P_j^{(m)}$ é ou não viável em termos do tempo limite U_j .

Além disso, observamos que o custo ótimo \bar{C}_i de $\pi P_i^{(m)}$ não é o custo ótimo C_i de π_i , e geralmente o tempo de

chegada $\bar{t}_{oi} < t_{oi}$.



Representação de uma relação entre custos (π_i) e tempos de chegada e de execução de um trip i .

(FIGURA 8.3)

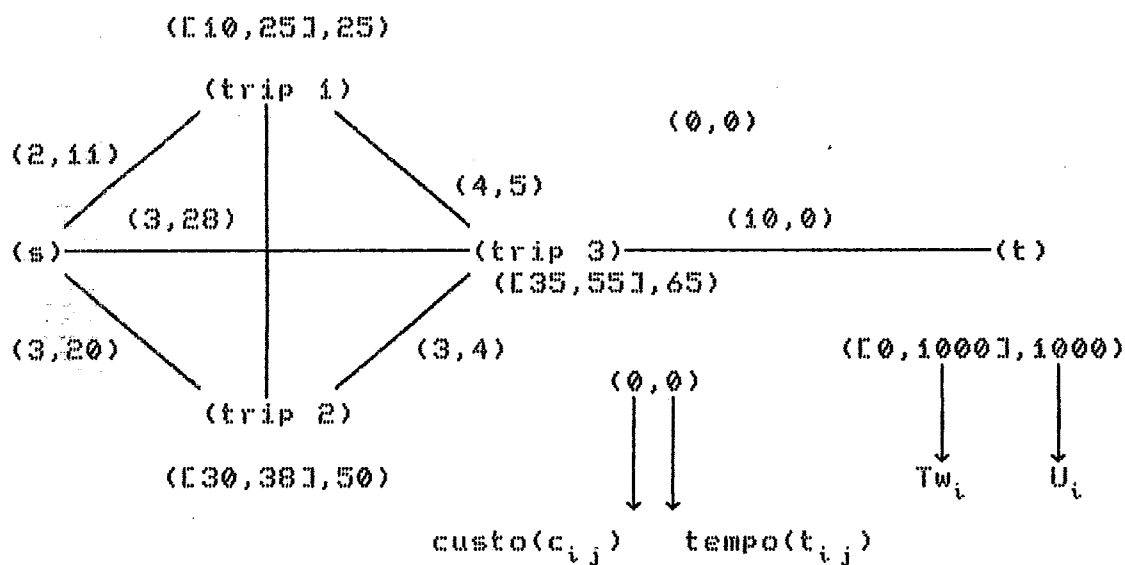
Pela figura 8.3) notamos que alguns *schedules* de πP_i são descartados por inviabilidade final.

Podemos então dizer que na etapa inicial determinamos soluções "otimistas" πP_i e na etapa final (π_i) procuramos as soluções realistas, (viáveis).

Resolvemos, a seguir, um pequeno exemplo numérico aplicando passo a passo o algoritmo aqui proposto.

EXEMPLO NUMÉRICO: Considere:

$c_1 = 5$	$t_1 = 5$	$U_1 = 25$
$c_2 = 6$	$t_2 = 10$	$U_2 = 50$
$c_3 = 10$	$t_3 = 10$	$U_3 = 65$
$c_t = 0$	$t_t = 0$	$U_t = 1000$



(FIGURA 8.4)

PASSO 0: INICIALIZAÇÃO:

$$\pi_{s0}^{(1)} = (0,0)$$

$$\pi_{s0}^{(1)} = (0,0,0)$$

$$K = K - \{s_0\} = \{s_0, 1, 2, 3, t\} - \{s_0\} = \{1, 2, 3, t\}$$

$$K^{(1)} = K^{(2)} = \dots = K \setminus |N|^{-1}$$

PASSO 1: APROXIMAÇÃO $m = 1$:

P1.1)

$$j = 1 \Rightarrow t_{s_01} = 11 \in [10, 25] \Rightarrow \pi P_1^{(1)} = (\bar{c}_1^{(1)}, T\theta_1^{(1)}) = \\ = (2, 11) \Rightarrow K^{(1)} = \{1\}$$

$$j = 2 \Rightarrow t_{s_02} = 20 \notin [30, 38] \Rightarrow \pi P_2^{(1)} = (\bar{c}_2^{(1)}, T\theta_2^{(1)}) = (00, 00)$$

$$j = 3 \Rightarrow t_{s_03} = 28 \notin [40, 55] \Rightarrow \pi P_3^{(1)} = (\bar{c}_3^{(1)}, T\theta_3^{(1)}) = (00, 00)$$

$$j = 4 \Rightarrow t_{s_0t} = 00 \notin [0, 100] \Rightarrow \pi P_t^{(1)} = (c_t^{(1)}, T\theta_t^{(1)}) = (00, 00)$$

P1.2) $K^{(1)} = \{1\}$

$$j = 1 \Rightarrow TE_1^{(1)} = T\theta_1^{(1)} + t_1 = 11 + 5 = 16 < 25 = U_1 \Rightarrow c_1^{(1)} =$$

$$\bar{c}_1^{(1)} + c_1 = 7 \Rightarrow \pi_1^{(1)} = (c_1^{(1)}, T\theta_1^{(1)}, TE_1^{(1)}) = (7, 11, 16)$$

$$\pi_2^{(1)} = (c_2^{(1)}, T\theta_2^{(1)}, TE_2^{(1)}) = (00, 00, 00)$$

$$\pi_3^{(1)} = (c_3^{(1)}, T\theta_3^{(1)}, TE_3^{(1)}) = (00, 00, 00)$$

$$\pi_t^{(1)} = (c_t^{(1)}, T\theta_t^{(1)}, TE_t^{(1)}) = (00, 00, 00)$$

PASSO 2: APROXIMAÇÃO SUCESSIVA ($m = 2$) dos Π_j

P2.1) Para cada $j \in K = \{1, 2, 3, t\}$

$$j = 1$$

$$TE_2^{(1)} + t_{21} = 00 + 7 = 00 \notin [10, 35] \Rightarrow c_2^{(1)} + c_{21} = 00$$

$$TE_3^{(1)} + t_{31} = 00 + 5 = 00 \notin [10, 25] \Rightarrow c_3^{(1)} + c_{31} = 00$$

$$TE_t^{(1)} + t_{t1} = 00 + 0 = 00 \notin [10, 25] \Rightarrow c_t^{(1)} + c_{t1} = 00$$

$$\log_0 \pi F_1^{(2)} = (c_1^{(2)}, T\theta_1^{(2)}) = (00, 00)$$

$$j = 2$$

$$TE_1^{(1)} + t_{12} = 16 + 7 = 25 \notin [30, 38] \Rightarrow c_1^{(1)} + c_{12} = 00$$

$$TE_3^{(1)} + t_{32} = 00 + 44 = 00 \notin [30, 38] \Rightarrow c_3^{(1)} + c_{32} = 00$$

$$TE_t^{(1)} + t_{t2} = 00 + 0 = 00 \notin [30, 38] \Rightarrow c_t^{(1)} + c_{t2} = 00$$

$$\log_0 \pi F_2^{(2)} = (c_2^{(2)}, T\theta_2^{(2)}) = (00, 00)$$

$$j = 3$$

$$TE_1^{(1)} + t_{13} = 16 + 5 = 21 \notin [40, 55] \Rightarrow c_1^{(1)} + c_{13} = 00$$

$$TE_2^{(1)} + t_{23} = 00 + 4 = 00 \notin [40, 55] \Rightarrow c_2^{(1)} + c_{23} = 00$$

$$TE_t^{(1)} + t_{t3} = 00 + 0 = 00 \notin [40, 55] \Rightarrow c_t^{(1)} + c_{t3} = 00$$

$$\log_0 \pi F_3^{(2)} = (c_3^{(2)}, T\theta_3^{(2)}) = (00, 00)$$

$$j = t$$

$$TE_1^{(1)} + t_{1t} = 16 + 0 = 16 \in [0, 1000] \Rightarrow c_1^{(1)} + c_{1t} = 7 + 0 = 7$$

$$TE_2^{(1)} + t_{2t} = 00 + 0 = 00 \in [0, 1000] \Rightarrow c_2^{(1)} + c_{2t} = 00$$

$$TE_3^{(1)} + t_{3t} = 00 + 0 = 00 \in [0, 1000] \Rightarrow c_3^{(1)} + c_{3t} = 00$$

$$\log_0 \pi P_t^{(2)} = (c_t^{(2)}, T\theta_t^{(2)}) =$$

$$= (\min\{c_t^{(1)}, \min\{7, 00, 00\}\}, \min\{TE_t^{(1)}, \min\{16, 00, 00\}\}) =$$

$$= (\min\{00, 7\}, \min\{00, 16\}) = (7, 16) = (c_t^{(2)}, T\theta_t^{(2)})$$

$$\text{Faça: } K^{(2)} = \{t\}$$

P2.2) Para $j \in K^{(2)} = \{t\}$ faça:

$$j = t:$$

$$TE_t^{(2)} = T\theta_t^{(2)} + t_t = 16 + 0 = 116 < 10000 = U_t \Rightarrow$$

$$\Pi_t^{(2)} = (c_t^{(2)}, T\theta_t^{(2)}, TE_t^{(2)}) = (7 + 0, 16, 16) = (7, 16, 16)^*$$

Para $j \in K - K^{(2)} = \{1, 2, 3\}$ faça:

$$\Pi_1^{(2)} = (00, 00, 00)$$

$$\Pi_2^{(2)} = (00, 00, 00)$$

$$\Pi_3^{(2)} = (00, 00, 00)$$

PASSO 3: Como $M = 2 < 3 = |N| - 1$, retorne ao Passo 2) com $m = 3$

PASSO 2: $M = 3$

P2.1) $\forall j \in \{1, 2, 3\}$ faça

$j = 1$

$$TE_2^{(2)} + t_{21} = 00 + t_{21} \notin [10, 25] \Rightarrow c_2^{(2)} + c_{21} = 00$$

$$TE_3^{(2)} + t_{31} = 00 + t_{31} \notin [10, 25] \Rightarrow c_3^{(2)} + c_{31} = 00$$

$$TE_t^{(2)} + t_{t1} = 16 + 0 = 16 \in [10, 25] \Rightarrow c_t^{(2)} + c_{t1} = 7 + 0 = 7$$

$$\text{Logo } \pi P_1^{(3)} = (C_1^{(3)}, TO_1^{(3)}) =$$

$$= [\min(c_1^{(2)}, \min(00, 00, 7)), \min(TE_1^{(2)}, \min(00, 00, 16))] =$$

$$= [\min(00, 7), \min(00, 16)] = (7, 16)$$

$$j = 2: TE_1^{(2)} + t_{12} = 00 + t_{12} = 00 \notin [30, 38] \Rightarrow c_1^{(2)} + c_{12} = 00$$

$$TE_3^{(2)} + t_{32} = 00 + t_{32} = 00 \notin [30, 38] \Rightarrow c_3^{(2)} + c_{32} = 00$$

$$TE_t^{(2)} + t_{t2} = 16 + 0 = 16 \notin [30, 38] \Rightarrow c_t^{(2)} + c_{t2} = 00$$

$$\text{Logo } \pi P_2^{(3)} = (C_2^{(3)}, TO_2^{(3)}) = (00, 00)$$

$$j = 3$$

$$TE_1^{(2)} + t_{13} = 00 + t_{13} = 00 \in [35, 55] \Rightarrow c_1^{(2)} + c_{13} = 00$$

$$TE_2^{(2)} + t_{23} = 00 + t_{23} = 00 \in [35, 55] \Rightarrow c_2^{(2)} + c_{23} = 00$$

$$TE_t^{(2)} + t_{t3} = 16 + 0 = 16 \in [35, 55] \Rightarrow c_t^{(2)} + c_{t3} = 00$$

$$\text{logo } \pi P_3^{(3)} = (c_3^{(3)}, T\theta_3^{(3)}) = (00, 00)$$

$$j = t$$

$$TE_1^{(2)} + t_{1t} = 00 + t_{1t} = 00 \in [0, 1000] \Rightarrow c_1^{(2)} + c_{1t} = 00$$

$$TE_2^{(2)} + t_{2t} = 00 + t_{2t} = 00 \in [0, 1000] \Rightarrow c_2^{(2)} + c_{2t} = 00$$

$$TE_3^{(2)} + t_{3t} = 00 + t_{3t} = 00 \in [0, 1000] \Rightarrow c_3^{(2)} + c_{3t} = 00$$

$$\text{logo } \pi P_t^{(3)} = (c_t^{(3)}, T\theta_t^{(3)}) = (00, 00) \in K^{(3)} = \emptyset$$

P2.2) $K^{(3)} = \emptyset$ então

$$\Pi_1^{(3)} = (00, 00, 00)$$

$$\Pi_2^{(3)} = (00, 00, 00)$$

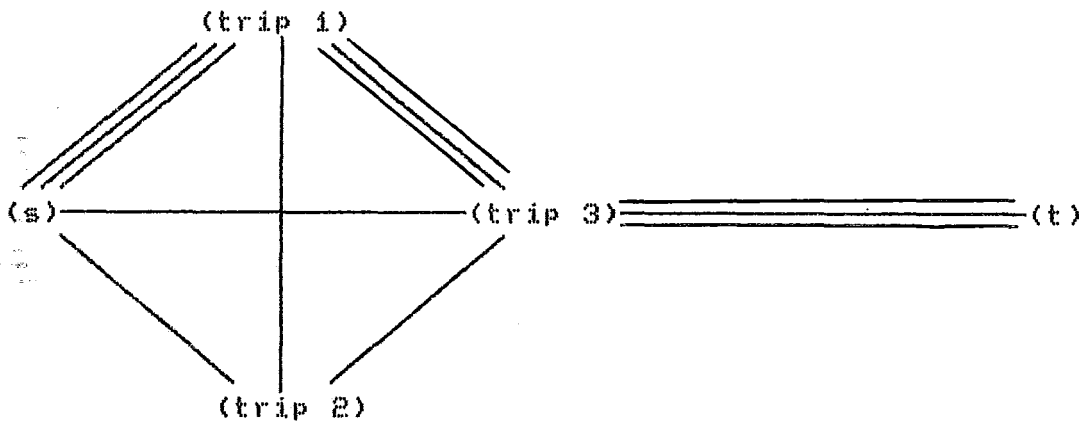
$$\Pi_3^{(3)} = (00, 00, 00)$$

$$\Pi_t^{(3)} = (00, 00, 00)$$

PASSO 3) Como $m = 3 = |N| - 1$ (FIM)

Schedule Ótimo será dado por

$\pi_t^* = (c^*, T\theta^*, TE^*) = \pi_t^{(2)} = (7, 16, 16)$, de custo global 7, efetuando $2-1 = 1$ trip e um tempo global de $t = 16$



(≡) = schedule Ótimo : (s) → (trip 1) → (t)

(Figura 8.5)

VIII.6 - SOLUÇÃO DO PROBLEMA MESTRE (SCA)

O problema de recobrimento com restrições adicionais (SCA) é resolvido com relaxação linear pelo método simplex gerando, a cada iteração, uma nova coluna para ingressar na base, como mostramos em VII.5).

Caso a solução ótima linear seja também uma solução inteira, neste caso o procedimento chega ao seu final; por outro lado, se o algoritmo simplex gerar uma solução ótima não inteira, utiliza-se um algoritmo do tipo *branch-and-bound*, para obter solução ótima inteira.

No nosso procedimento de resolver o problema (SCA), é fácil de se verificar que o simplex dispense maior esforço computacional, se comparado com a resolução do subproblema de geração de colunas.

Assim, podemos pensar em gerar à cada iteração do método simplex, ao invés de uma coluna, várias colunas, procurando reduzir com isso o número de iterações do simplex.

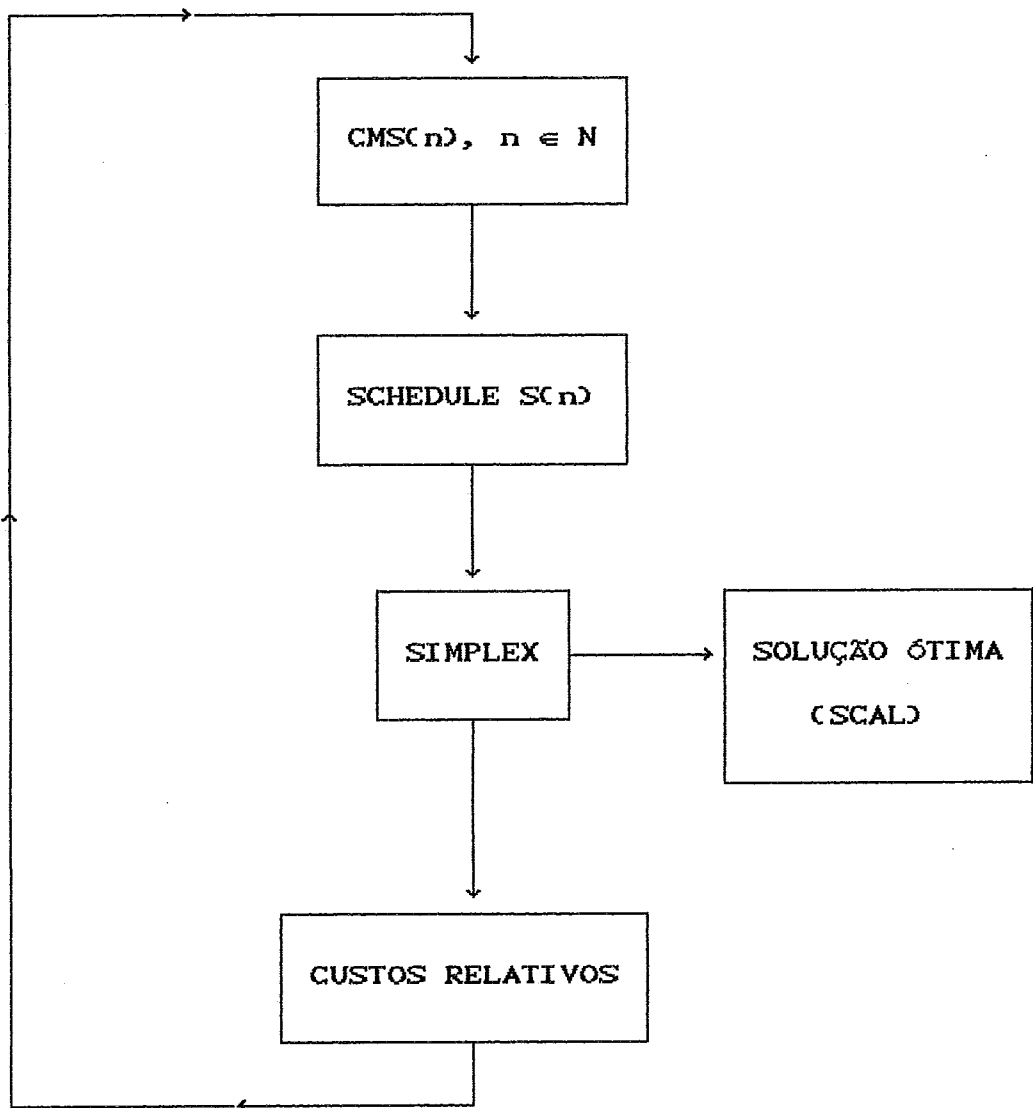
Esta proposta é reforçada se observarmos que no nosso procedimento de geração de colunas, podemos gerar a cada iteração do simplex, no máximo, uma coluna de cada subproblema $CMS(n)$, $n \in N$.

Assim, à cada iteração do simplex, o número de colunas geradas seria o número de subproblemas $CMS(n)$ com valor ótimo negativo ($V(CMS(n)) < 0$).

Um caminho alternativo para resolver o problema (SCA), é pelos algoritmos polinomiais do tipo Karmarkar (1983), onde algumas versões, tem complexidade $O(Ln^3)$.

Finalmente, uma iteração do problema (SCA) com relaxação linear, seguindo os procedimentos desenvolvidos neste trabalho, tem a seguinte configuração.

ITERAÇÕES DO PROBLEMA (SCAL)



(FIGURA 8.6)

CAPÍTULO IX

CONCLUSÕES E PROPOSTAS PARA TRABALHOS FUTUROS

CONCLUSÕES

O problema de roteamento e *scheduling* de veículos é atualmente uma das áreas de maior sucesso dentro da programação matemática, evidenciado pelo elevado número de aplicações e desenvolvimentos teóricos encontrados na literatura.

Contudo, como citamos no desenvolvimento deste trabalho, as aplicações de problemas de roteamento de navios têm sido pouco exploradas na literatura.

Diante disto, passamos a acreditar na necessidade de se desenvolver uma pesquisa mais específica para esta classe de aplicações, através de abordagens mais profundas e atualizadas sobre os problemas de roteamento de navios, com o intuito de também divulgar o alcance das técnicas de otimização nesta área, esperando com isso o desenvolvimento de novas técnicas direcionadas para este tipo de problemas.

Os problemas de roteamento de veículos são, na sua maioria, problemas NP-árduos (*NP-hard*) e em alguns casos, o problema de encontrar uma solução viável já é um problema NP-completo, o que acaba influenciando na escolha de

algoritmos heurísticos na maioria das aplicações encontradas na literatura.

Contudo, acreditamos ser vital o desenvolvimento de novas técnicas exatas para serem utilizadas ou isoladamente ou conjugadas a métodos heurísticos na solução destes problemas.

Nos últimos anos tem surgido na literatura novas técnicas de relaxações e decomposições lagrangeanas bem como novos procedimentos exatos para resolver problemas considerados até então insolúveis via otimização.

O nosso trabalho foi iniciado com a geração de várias formulações para o problema (FP), como problemas de programação linear misto.

As relaxações e decomposições lagrangeanas desenvolvidas no capítulo 4 se mostraram eficientes quando aplicadas à maioria das formulações anteriormente geradas.

Na formulação (F1), uma relaxação lagrangeana convencional (veja *GEOFFRION (1974)*) transforma o problema num modelo clássico de recobrimento que embora também seja NP-completo, possui bons algoritmos para a sua solução.

Ao dualizarmos o outro conjunto de restrições de (F1),

geramos um novo problema relaxado que pode ser resolvido através de uma simples inspeção nos coeficientes da sua função objetivo.

Resultados semelhantes foram obtidos usando técnicas de decomposição lagrangeana ao problema (F1), com a vantagem de que neste caso, com determinados critérios de "copiar" variáveis, reduzimos sensivelmente o número de multiplicadores duais além de obter limites inferiores melhores ou iguais aos obtidos via relaxações lagrangeanas convencionais.

Outro resultado promissor foi obtido na formulação (F3), ao utilizarmos uma técnica que converte problemas com frota heterogênea em um conjunto de problemas com frota homogênea através de relaxações lagrangeanas adequadas.

Este procedimento reduziu drasticamente as dimensões do problema a ser resolvido como mostrado na tabela 5.2 do capítulo 5, onde um problema original com aproximadamente 154.000 restrições e 159.000 variáveis foi relaxado para um conjunto de problemas com no máximo 5.000 restrições e 5.300 variáveis.

A eficiência das técnicas de relaxação lagrangeana pode ainda ser sentida na formulação (F5) quando um problema de aproximadamente 80.000 restrições e 175.000 variáveis foi

reduzido para problemas com no máximo 4.500 restrições e 7.500 variáveis.

Estes números foram ainda mais reduzidos ao utilizarmos técnicas de converter modelos de frota heterogênea em um conjunto de modelos com frota homogênea, resultando em problemas com no máximo 2.600 restrições de igualdade e 5.200 variáveis primais com 50 multiplicadores duais.

Ainda na formulação (F5), ao utilizarmos técnicas de decomposição lagrangeana, geramos problemas relaxados que fornecem limites estritamente melhores que aos obtidos via relaxações lagrangeanas convencionais, necessitando apenas resolver problemas com até 1.500 restrições de igualdade, 4.000 variáveis e 50 multiplicadores duais.

No capítulo 6, desenvolvemos um algoritmo heurístico que pode ser visto como uma extensão da heurística de CLARKE e WRITE (1964).

Como no nosso problema a frota de navios é heterogênea, as sintetizações entre dois *schedules* foram sempre referidas a um determinado navio, em termos da sua viabilidade.

Além disso, como no problema aplicativo (FP) existem restrições do tipo *time-windows* e conseqüentemente o surgimento de variáveis de tempo, existe a necessidade de se

optar pela economia ou do tempo ou dos custos que normalmente são objetivos conflitantes (já que velocidades maiores acarretam em custos maiores de combustíveis mas possibilitam o atendimento de um número maior de tarefas por cada navio da frota).

No nosso algoritmo, fizemos a opção de minimizar os tempos gastos pelos navios da frota para efetuar cada trip e somente nos casos de empate, escolhemos sintetizações de menor custo.

Isto pode ser justificado pelas características do problema (FP), cuja frota atual de navios é insuficiente para atender a todos os trips (contratos), exigindo normalmente o afretamento de navios no mercado que ao final pode acabar incorrendo num custo global maior.

O algoritmo é bastante flexível, comportando facilmente modificações na sua estrutura como, por exemplo, podemos pensar em alterar o critério do cálculo das economias usado no algoritmo por um critério baseado numa combinação linear dos custos e dos tempos envolvidos, de forma semelhante ao proposto por *SOLOMON (1987)*.

A heurística foi testada para exemplos de dimensões reduzidas gerados a partir de dados previamente conhecidos do problema (FP), encontrando soluções ótimas ou muito

próximas do ótimo.

O procedimento exato proposto no capítulo 8 apresenta um novo esquema de geração de colunas para o problema de recobrimento com restrições adicionais (formulação (F1)), relaxando as restrições de integralidade.

O subproblema de geração de colunas foi descrito como um problema de caminho mínimo com restrições adicionais de tempo (CMS) onde a sua solução ótima equivale ao schedule viável de menor custo relativo que deve ser incorporado à nova base do Simplex do problema mestre (F1) relaxado, se o valor ótimo $V(\text{CMS})$ for negativo.

O algoritmo global é uma generalização do método anteriormente proposto por *DESROSIERS, SOUMIS e DESROCHERS (1983)*, incorporando restrições de: tempo limite para efetuar um trip, frota heterogênea de veículos e múltiplas origens.

Para a busca de uma solução ótima inteira, foi sugerido o uso de um procedimento de *Branch-and-Bound*.

PROPOSTAS PARA TRABALHOS FUTUROS

Dentre as futuras pesquisas a serem realizadas estão:

i) A implementação dos algoritmos aqui desenvolvidos para exemplos de grande porte.

ii) O uso de técnicas de Programação Linear Multiobjetivo mais especificamente com dois objetivos a saber: a minimização de tempos e a minimização dos custos que normalmente são itens conflitantes.

iii) Uma nova formulação matemática do problema (FP) generalizando a formulação proposta por *MILLER, TUCKER e ZEMLIN (1960)* (evitar formação de sub-rotas no (FCV)) para as restrições de scheduling e algumas restrições de roteamento do problema (FP).

BIBLIOGRAFIA

AGARWAL, Y.K. (1985), "Set Partitioning Approach to Vehicle Routing" - Logistics Resource - Leaseway Trans.Corp.

ANGEL, R; CAUDLE, W; NOONAN, R and WHINSTON, A. (1972), "Computer Assisted School Bus Scheduling" - *MANAGEMENT SCIENCE* 18(6), pp: 279-288.

APPELGREN, L.H. (1969), "A Column Generation Algorithm for a Ship Scheduling Problems", *TRANSPORTATION SCIENCE* 3, pp: 53-68.

APPELGREN, L.H. (1971), "Integer Programming Methods for a Vessel Scheduling Problem", *TRANSPORTATION SCIENCE* 5(1), pp: 64-78.

ARABEYRE, J; FEARNLEY, J; STEIGER, F; TEATHER, W.(1969); "The Airline Crew Scheduling Problem : A Survey", *TRANSPORTATION SCIENCE*, 3(2), pp: 140-163.

ARISAWA, S.; ELMAGHRABY, S (1977); "The "hub" and "wheel" Scheduling Problems: Part I and II, *TRANSPORTATION SCIENCE*, 2, pp: 124-163.

ARTHUR, J. L.; FRENDEWEY, J. O. (1988), "Generating Travelling-Salesman Problems with Know Optimal Tours", *JOURNAL of OPERATIONAL RESEARCH SOCIETY*, 39(2), pp: 153-159.

ASGHRAZADEH, K. and NEWELL, G. (1978), "Optimal Dispatching Strategies for Vehicles Having Exponentially Distributed Trip Times", *NAVAL RESEARCH LOGISTICS QUARTERLY*, 25(3) pp: 489-509.

ASSAD, A. A. (1988), "Modeling and Implementation Issues in Vehicle Routing", *VEHICLE ROUTING: METHODS AND STUDIES*, edited by GOLDEN, B. L. and ASSAD, A. A., NORTH HOLLAND.

BAKER, E. K. (1983), "An Exact Algorithm for the Time-Constrained Travelling Salesman Problem", *OPERATIONS RESEARCH*, 31(5), pp: 938-945.

BAKER, E. K.; BODIN, L.; FINNEGAN, W. and PONDER, R. (1979), "Efficient Heuristic Solutions to an Airline Crew Scheduling Problem", *AIIE TRANS.* 11(2), pp: 79-85

BAKER, E. K.; FISCHER, M. (1981), "Computational Results for Very Large Air Crew Scheduling Problems", *OMEGA* 9, pp: 613-618.

BAKER, E. K.; CRABILL, T. and MAGAZINE, M. (1973), "An Optimal Procedure for Allocating Manpower with cyclic Requeriments", *AIIE TRANS.* 5, pp: 119-126.

BALAS, E. and CHRISTOFIDES, N. (1981), "A Restrired Lagrangean Approach to the Traveling Salesman Problem", *MATHEMATICAL PROGRAMMING* 21, pp: 19-46.

BALAS, E. and HO, A. (1980), "Set Covering Algorithms Using Cutting Planes, Heuristics and Subgradients Optimization: A Computational Study", *MATHEMATICAL PROGRAMMING STUDY* 12, pp: 37-60.

BALINSKI, M. and QUANDT, R. (1964), "On an Integer Program for a Delivery Problem", *OPERATIONS REASEARCH* 12, pp: 300-304.

BALL, M.; BODIN, L. and DIAL, R. (1983), "A Matching Based Heuristics for Scheduling Mass Transit Crew and Vehicles", *TRANSPORTATION SCIENCE* 17(1), pp: 4-31.

BALL, M.; GOLDEN, B.; ASSAD, A. and BODIN, L. (1983), "Planning for truck fleet size in the presence of a common carrier option", *DECISION SCIENCES*, 14(1), pp: 103-120.

BALL, M.; MAGAZINE, M. (1981), "The design and analysis of heuristics", *NETWORKS*, 11, pp: 215-219.

BAZARAA, M.; GOODE, J. (1977), "The traveling Salesman: a duality approach", *MATHEMATICAL PROGRAMMING*, 13, pp: 221-237.

BARTHOLDI, . and QUANDT, R. (1964), "On an Integer Program for a Delivery Problem", *OPERATIONS RESEARCH* 12, pp: 300-304.

BEASLEY, J (1981), "Adapting the savings Algorithm for varying inter-customer travel times", *OMEGA*, 9, pp: 658-659.

BECKER, R. (1987), "Programação de uma Frota de Petroleiros: Proposta de um Método para Longo Curso", *TESE DE MESTRADO. COPPE-UFRJ.*

BECKER, R.; MACULAN, N.; OCHI, L.S. (1988), "Um método de otimização para a geração de percursos ótimos de uma frota de navios petroleiros", *IV-CLAIO, RJ, BRASIL.*

BELARDO, S.; DUCHESSI, P. and SEAGLE, J. (1985), "Microcomputer graphis in support of vehicle fleet routing", *INTERFACES* 15(6), pp: 84-92.

BELLMAN, R. (1962), "Dynamic Programming Treatment of the Traveling Salesman Problem", *J. ACM.* 9, pp: 61-63.

BELLMAN, R. and COOKE, K. (1969), "The Konigsberg Bridges Problem Generalized" *J. MATH. ANAL. APPL.* 25, pp: 1-7.

BELLMORE, M.; BENNINGTON, G. and LUBORE, S. (1971), "A

Multivehicle tanker Scheduling Problem", *TRANSPORT. SCIENCES* 5(1), pp: 36-74.

BELLMORE, M. and HONG, S. (1974), "Transformation of Multi-Salesman Problem to the standart traveling Salesman Problem", *J. ACM.* 21, pp: 500-504.

BELLMORE, M. and MALONE, J. (1971), "Pathology of traveling-salesman subtour-elimination algorithms", *OPERATIONS RESEARCH*, 19, pp: 278-307.

BELLMORE, M. and NEMHAUSER, G. (1968); "The traveling salesman Problem: A Survey", *OPERATIONS RESEARCH*, 16, pp: 538-558.

BELTRAMI, E. and BODIN, L. (1974), "Networks and Vehicle Routing for Municipal Waste Collection", *NETWORKS*, 4, pp: 65-94.

BENNET, B. and GAZIS, D. (1972), "School Bus Routing by Computer", *TRANSPORTATION RESEARCH* 6(4), pp: 317-325.

BERENQUER, X (1979), "A Characterization of Linear Admissible Transformations for the M-Traveling Salesman Problem" *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, 3(3), pp: 232-238.

BERMAN, O. and SIMCHI-LEVI, D. (1986), "Minisum Location of

a Traveling Salesman". *NETWORKS*, 16, pp: 239-254.

BLAIS, J. Y.; ROUSSEAU, J. (1981), "HASTUS, for bus operator scheduling: The Man-Machine Interface"., *UNIVERSITE DE MONTREAL, CENTRE DE RECHERCHE SUR LES TRANSPORTS*, 231.

BLADLEY, G.; BROWN, G. and GRAVES, G. (1977), "Design and Implementation of Large Scale Primal Transshipment algorithms, *MANAGEMENT SCIENCES*, 24(1), pp: 1-34.

BODIN, L. and BERMAN, O. (1979), "Routing and Scheduling of School Buses by Computer", *TRANSPORTATION SCIENCES*, 13(2), pp: 113-129.

BODIN, L. and GOLDEN, B. (1981), "Classification in Vehicle Routing and Scheduling", *NETWORKS*, 11(2), pp: 97-108.

BODIN, L; GOLDEN, B; ASSAD, A. and BALL, M. (1983), "Routing and Scheduling of Vehicles and Crews: The State of the Art", *COMPUTERS & OPERATIONS RESEARCH*, 10(2), pp: 63-211.

BODIN, L.; GOLDEN, B; SCHUSTER, A. and ROMIG, W. (1980), "A Model for the Blocking of Trains.", *TRANSPORTATION; RESEARCH* 14B, pp: 115-120.

BODIN, L. and KURSH, S. (1978), "A Computer-Assisted System for the Routing and Scheduling of Street Sweepers, *OPERATIONS RESEARCH* 26(4), pp: 525-537.

BODIN, L.; KYDES, A. and ROSENFELD, D. (1975), "Approximation Techniques for Automated Manpower Scheduling", *WORKSHOP ON AUTOMATED TECHNIQUES FOR SCHEDULING OF VEHICLE OPERATORS FOR URBAN PUBLIC TRANSPORTATION SERVICES*.

BONOMI; and LUTTON, S.L. (1984), "The N-City Traveling Salesman Problem: Statistical Mechanics and the Metropolis Algorithms", *SIAM REVIEW*, 26(4), pp: 551-568.

BOOKBINDER, J.H. and REECE, E. (1988), "Vehicle Routing Considerations in distribution System Design", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*. 37, pp: 204-213.

BROWN, G. and GRAVES, G. (1981), "Real time Dispatch of Petroleum Tank Trucks", *MANAGEMENT SCIENCE*, 27, pp: 19-32.

BROWN, G.; GRAVES, G.; RONEN, D. (1987), "Scheduling Ocean Transportation of Crude Oil", *MANAGEMENT SCIENCE*, 33(3), pp: 335-345.

BURNESS, R. and WHITE, J. (1976), "The Traveling Salesman Location Problem", *TRANSPORTATION SCIENCES*, 10(4), pp: 348-360.

BUXEY, G. (1985), "Designing routes to minimize fleet operating costs", *EUROPEAN JOURNAL OF OPERATIONAL*

RESEARCH, 21, pp: 57-64.

CARPANETO, G. DELL'AMICO, M.; FISCHETH, M.; TOTH, P. (1987), "A Branch-and-Bound Algorithm for the Multiple Depot Vehicle Scheduling Problem" , RESEARCH REPORT. OR/87/2, UNIVERSITY OF MODENA, ITALY.

CARPANETO, G.; FISCHETTI, M.; TOTH, P. (1988), "New Lower Bounds for the Symmetric Travelling Salesman Problem", TECHNICAL REPORT OR/87/6, DEIS, UNIV. BOLOGNA, ITALY (TO APPEAR IN MATH. PROG. B).

CARPANETO, G.; TOTH, P. (1980), "Some new Branching and Bouding Criteria for the Asymmetric Traveling Salesman Problem", MANAGEMENT SCIENCE 26(7), pp: 736-743.

CARPANETO, G.; MARTELO, S. and TOTH, P. (1986), "An Algorithm for the Bottleneck Traveling Salesman Problem", OPERATIONS RESEARCH, pp: 380-389.

CASCO, D.O.; GOLDEN, B.L. and WASIL, E.A. (1988), "Vehicle Routing with Backhauls: Models, Algorithms and Case Studies", VEHICLE ROUTING: METHODS AND STUDIES, NORTH-HOLLAND, pp: 127-148.

CAVALIER, T.M.; CIZANDRA, M.J. (1986), "Heuristic Algorithm for Assigning Crews Among Bases in an Airlift Operation", JOURNAL OF THE OPERATION RESEARCH SOC. 37(4), pp: 381-386.

CEDER, A. and STERN, H. (1981), "Deficit Function Bus Scheduling with Deadheading Trip Insertions for Fleet Size Reduction", *TRANSPORTATION SCIENCES*, 15, pp: 338-363.

CELESTINO, N.B. (1973), "Contribuição ao Estudo do Despacho de Veículos", *TESE DE MESTRADO, PUC-RJ*.

CERDEIRA, J.O. e COELHO, J.D. (1984), "Optimização de Percursos de Distribuição", *UNIVERSIDADE DE LISBOA*.

CHAPLEAU, L.; FERLAND, J.; LAPALME, G. and ROUSSEAU, J. (1982), "A Parallel Insert Method for the Capacited arc routing Problem", *UNIVERSITE DE MONTREAL, CENTRE DE RECHERCHE SUR LES TRANSPORTS, PUBLICATION 234*.

CHAPLEAU, L.; FERLAND, J. and ROUSSEAU, J.M. (1981), "Clustering for Routing in Dense Area", *UNIVERSITE DE MONTREAL, CENTER DE RECHERCHE SUR LES TRANSPORTS, PUBLICATION 206*.

CHISMAN, J. (1975), "The Clustered Traveling Salesman Problem", *COMPUTERS & OPERATIONS RESEARCH* 2(2), pp: 115-119.

CHRISTOFIDES, N. (1972), "Bounds for the Traveling Salesman Problem", *OPERATIONS RESEARCH* 20, pp: 1044-1056.

CHRISTOFIDES, N. and EILON, S. (1969), "An Algorithm for the

Vehicle Dispatching Problem", *OPERATIONAL RESEARCH QUARTELY* 20, pp: 309-318.

CHRISTOFIDES, N.; MINGOZZI, A. and TOTH, P. (1979), "The Vehicle Routing Problem", *COMBINATORIAL OPTIMIZATION - JOHN WILEY & SONS*. pp: 315-338.

CHRISTOFIDES, N.; MINGOZZI, A. and TOTH, P. (1981a), "Exact Algorithms for the Vehicle Routing Problem based on Spanning Tree and Shortest Path Relaxation", *MATHEMATICAL PROGRAMMING*. 20, pp: 255-282.

CHRISTOFIDES, N. MINGOZZI, A. and TOTH, P. (1981b), "State Space Relaxation Prodedures for the Computation of Bounds to Routing Problems", *NETWORKS* 11(2), pp: 145-164.

CHRISTOFIDES, N. VALDES, R.A. and TAMARIT, J.M. (1987), "Project Scheduling with Resource Constraints:: A Branch-and-Bound Approach", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*. 29, pp: 262-273.

CLARKE, G. and WRIGHT, J. (1964), "Scheduling of Vehicles From a Central Depot to a Number of Delivery Points", *OPERATIONS RESEARCH*. 12, pp: 568-581.

COOK, T. and RUSSELL, R. (1978), "A Simulation and Statistical Analysis of Stochastics Vehicle Routing with timing Constraints", *DECISION SCIENCE* 9, pp: 673-687.

CORNUEJOLS, G.; FONLUPT, J. and NADDEF, D. (1985), "The Traveling Salesman Problem on a Graph and Some Related Integer Polyhedra", *MATHEMATICAL PROGRAMMING*. 33, pp: 1-27.

CORNUEJOLS, G. and NEMHAUSER, G. (1978), "Tight Bounds on Christofide's Traveling Salesman Problem Heuristic", *MATHEMATICAL PROGRAMMING*. 14, pp: 115-121.

COSMADAKIS, S.S. and PAPADIMITRIOU, C.H. (1984), "The Travelling Salesman Problem with Many Visits to few Cities", *SIAM J. COMP.* 13(1), pp: 99-108.

CRAINIC, T. and ROUSSEAU, J.M. (1986), "The Column Generation Principle and the Airline Crew Scheduling Problem", *UNIVERSITE DE MONTREAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* 291.

CROES, G. (1958), "A Method for Solving Traveling Salesman Problems", *OPERATIONS RESEARCH*. 6, pp: 791-812.

CROWDER, H. and PADBERG, M. (1980), "Solving Large-Scale Symmetric Traveling Salesman Problems to Optimality", *MANAGEMENT SCIENCE* 26(5), pp: 495-509.

CULLEN, F.; JARVIS, J. and RATLIF, H. (1981), "Set Partitioning Based Heuristics for Interactive Routing", *NETWORKS*. 11(2), pp: 125-144.

CURRIN, D.C. (1986), "Transportation Problems with Inadmissible Routes", *JOURNAL OF THE OPERATIONAL RESEARCH SOC.* 37(4), pp: 387-396.

CUTHER, M. (1980), "Efficient Special Case Algorithms for the N-Line Planar Traveling Salesman Problem", *NETWORKS*, 10, pp: 183-195.

DAGANZO, C.F. (1985), "The Distance traveled to visit N points with a maximum of C stops per vehicle: An Analytic model and an application", *OPERATIONS RESEARCH*, pp: 331-350.

DANTZIG, G. and FULKERSON, D. (1954), "Minimizing the number of tankers to meet a fixed schedule", *NAVAL RES. LOGISTIC QUART.* 1, pp: 217-222.

DANTZIG, G.; FULKERSON, D. and JOHNSON, S. (1954) "Solution of a Large-scale traveling salesman problem", *OPERATIONS RESEARCH* 2(4), pp: 393-410.

DANTZIG, G.; FULKERSON, D. and JOHNSON, S. (1959), "On a linear Programming Combinatorial approach to the traveling salesman Problem", *OPERATIONS RESEARCH* 7(1), pp: 58-66.

DANTZIG, G. and RAMSER, J. (1959), "The Truck dispatching problem", *MANAGEMENT SCIENCE* 6, pp: 81-91.

D'ATRI, G. (1980), "A note on heuristics for the traveling salesman problem", *MATHEMATICAL PROGRAMMING* 19(1), pp: 111-114.

DÉJAX, P.J. and CRAINIC, T.G. (1987), "A Review of Empty Flows and Fleet Management Models in Freight Transportations", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

DESROCHERS, M. (1986), "La Fabrication D'horaires de travail pour les conducteurs D'autobus par une méthode de génération de colonnes", THÈSE DE PH.D, DIRIGÉE PAR FRANÇOIS SOUMIS, *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* 470.

DESROCHERS, M. et SOUMIS, F. (1985a), "Un algorithme de marquage permanent généralisé pour le problème de plus court chemin avec contraintes d'horaires", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* - 394 F.

DESROCHERS, M. and SOUMIS, F. (1985b), "A generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time-Windows", *UNIVERSITÉ DE MONTRÉAL CENTRE DE RECHERCHE SUE LES TRANSPORTS* (394A).

DESROCHERS, M. and SOUMIS, F. (1988), "A Reotimization Algorithm for the Shortest Path Problem with Time-Windows", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 35, pp 242-254.

DESROCHERS, M.; LENSTRA, J.K.; SAVELSBERGH, M.W.P. and SOUMIS, F. (1988), "Vehicle Routing with Time-Windows", *VEHICLE ROUTING: METHODS AND STUDIES-NORTH-HOLLAND*, pp: 65-84.

DESROSTIERS, J.; FERLAND, J.; ROUSSEAU, J.; LAPORTE, G. and CHAPLEAU, L. (1980), "A Multi-period School Bus: Routing and Scheduling System", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (164)*.

DESROSTIERS, J.; FERLAND, J.A.; ROUSSEAU J.M., LAPALME, J. and CHAPLEAU, L. (1980), "An Overview of School Busing System", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (219/417)*.

DESROSTIERS, J.; DUMAS, Y. and SOUMIS, F. (1984), "A dynamic Programming Method for the Large-Scale Single Vehicle Dial-a-Ride Problem with Time-Windows", *UNIVERSITÉ DE MONTRÉAL CENTRE DE RECHERCHE SUR LES TRANSPORTS (361)*.

DESROSTIERS, J.; LAPORTE, G. and SOUMIS, F. (1985), "Vehicle Routing with Full Loads", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (449)*.

DESROSTIERS, J.; SAUVÉ, M. and SOUMIS, F. (1985), "Shortest Path with Time-Windows Problems", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (389)*.

DESROSIERS, J.; SAUVÉ, M. and SOUMIS, F. (1986), "Lagrangean Relaxation Methods for Solving the minimum fleet size multiple traveling salesman problem with Time-Window", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

DESROSIERS, J.; SOUMIS, F. and DESROCHERS, M. (1983), "Routing with Time-Window by Column Generation", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

DESROSIERS, J.; SOUMIS, F. ; DESROCHERS, M. and SAUVÉ, M. (1983), "Routing and Scheduling by Branch-and-Bound on Time-windows", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (278)*.

DIJKSTRA, E. (1959), "A note on two problems in connection with graphs", *NUMER. MATH. 1*, pp: 269-271.

DINKEL, J.; KLEINDORFER, G.; KOCHENBERGER, G. and WONG, S. (1976), "Environmental inspection routes and the constrained traveling salesman problem", *COMPUTERS & OPERATIONS RESEARCH 3(4)*, pp: 269-282.

DINKEL, J. and WONG, D. (1984), "Applicatin of Sensitivity analysis to the optimization of petroleum drilling operations. *COMPUTERS & OPERATIONS RESEARCH 11(2)*, pp: 129-140.

DISCENZA, J. (1981), "A more compact formulation of the symmetric multiple traveling salesman problem with fixed charges", *NETWORKS* 11(1), pp: 73-75.

DOLL, C.L. (1980), "Quick and Dirty vehicle Routing Procedure", *INTERFACES*, 10(1), pp: 84-85.

DROR, M. (1988), "Cost Allocation: The Traveling Salesman, Binpacking, and the Knapsack Problem", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (577).

DROR, M. and FERLAND, J.A. (1985), "Cost Allocation in Combinatoric Optimization Problems: The TSP case", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (547).

DROR, M.; LAPORTE, G. and TRUDEAU, P. (1988), "Vehicle Routing with Stochastic Demands: Models and Issues", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (579).

DROR, M. and LEVY, L. (1986), "A Vehicle Routing Improvement Algorithm Comparison of a "Greedy" and a matching implementation for inventory routing", *COMPUTERS & OPERATIONS RESEARCH* 13(1), pp: 33-45.

DROR, M. and TRUDEAU, P. (1986), "Stochastic Vehicle Routing

with Modified Savings Algorithm", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 23, pp: 228-235.

DROR, M.; TRUDEAU, P. and LADANY, S.P. (1988), "Network Models for Seat Allocation on Flights", *TRANSPORTATION RESEARCH (B)* 22B, n^o 4, pp: 239-250.

DURAN, F. (1987), "A Large mixed integer production and distribution program", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*. 28, pp: 207-217.

EASTMAN (1958), "Linear Programming with Pattern Constraints" *HARVARD UNIVERSITY, CAMBRIDGE, MA.*

EDMONDS, J. and JOHNSON, E. (1973), "Matching, Euler tours, and the Chinese Postman", *MATHEMATICAL PROGRAMMING* 5, pp: 88-124.

EULER, L. (1953), "The Konigsberg bridges", *SCIENTIFIC AMERICAN* 189 pp: 66-70.

EVANS, S.R. and NORBACK, J.P. (1984), "An heuristic Method for Solving time-sensitive routing problems", *J. OPS. RES. SOC.* 35(5), pp: 407-414.

EVANS, S.R. and NORBACK, J.P. (1985), "The impact of a Decision Support System for Vehicle Routing in a Foodservice Supply Situation", *J. OPS. RES. SOC.* 36(6), pp:

467-472.

FEDERGUEN, A. and ZIPKIN, P. (1980), "A combined vehicle routing and inventory allocation problem", *RESEARCH WORKING PAPER. 345A, COLUMBIA UNIV.*

FERLAND, J.A. (1985), "The single depot vehicle scheduling problem", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (385).*

FERLAND, J.A. and FORTIN, L. (1987), "Vehicles Schedulings with Sliding Time-Window", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS.*

FERLAND, J.A. and MICHELON, P. (1987), "Vehicle Scheduling with Multiple Vehicle Types", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS.*

FISHER, M. (1980) "Worst-Case Analysis of heuristic Algorithms", *MANAGEMENT SCIENCE 26(1)*, pp: 1-17.

FISHER, M. (1981), "The Lagrangian Relaxation Method for Solving Integer Programming Problems", *MANAGEMENT SCIENCE 27*, pp: 1-12.

FISHER, M. (1988), "Lagrangian Optimization Algorithms for Vehicle Routing Problems", *OPERATIONAL RESEARCH 87 (INFOR 88)* pp: 635-649.

FISHER, M.; GREENFIELD, A.J.; JAIKUMAR, R. and LESTER, J.T (1982), "A Computerized Vehicle Routing Application", *INTERFACES* 12(4), pp: 42-51.

FISHER, M. and JAIKUMAR, R. (1981), "A generalized assignment heuristic for vehicle routing", *NETWORKS* 11(2), pp: 109-124.

FISHER, M. and JAIKUMAR, R. "The local delivery problem: Algorithms and Applications", *CORIGEM DESCONHECIDA*.

FISCHETTI, M and TOTH, P. (1987), "An Additive Bounding Procedure for Combinatorial Optimization Problems", *TECHNICAL REPORT-DEIS-UNIV. OF BOLOGNA-ITALY, CTO APPEAR IN OPERATIONS RESEARCH*.

FISCHETTI, M. and TOTH, P. (1988a), "An Additive Bounding Procedure for the Asymmetric Traveling Salesman Problem", *TECHNICAL REPORT-DEIS-UNIV. OF BOLOGNA-ITALY CTO APPEAR IN MATHEMATICAL PROGRAMMING A*.

FISCHETTI, M. and TOTH, P. (1988b), "An additive approach for the optimal solution of the prize-collecting travelling salesman problem", *VEHICLE ROUTING: METHODS AND STUDIES, NORTH-HOLLAND, pp: 319-344*.

FISCHETTI, M.; MARTELLO, S. and TOTH, P. (1987), "The fixed

job schedule problems with working-time constraints",
RESEARCH REPORT OR/87/1-DEIS-UNIV. OF. BOLOGNA-ITALY.

FLEISCHMAN, B. (1985), "A cutting plane procedure for the traveling salesman problem on road networks", *EUR. J. OPS. RES* 21 pp: 307-317.

FLEISCHMAN, B. (1988), "A New Class of Cutting Planes for the Symmetric Travelling Salesman Problems", *MATHEMATICAL PROGRAMMING* 40, pp: 225-246.

FLOOD, M. (1956), "The Traveling Salesman Problem", *OPERATIONS RESEARCH* 4, pp: 61-75.

FOSTER, B. and RYAN, D. (1976), "An integer programming approach to the vehicle scheduling problem", *OPS. RES. QUART.* 27 pp: 367-384.

FOX, K.; GAVISH, B. and GRAVES, S. (1980), "An n-constraints formulation of the (time dependent) traveling salesman problem", *OPERATIONS RESEARCH* 28(4), pp: 1018-1021.

FREDERICKSON, G. (1979), "Approximation Algorithm for some postman problems", *J. ACM.* 26, pp: 538-554.

FREDERICKSON, G.; HECHT, M. and KIM, C. (1978), "Approximation Algorithms for some routing problems", *SIAM JOURNAL COMP.* 7(2), pp: 178-193.

FRIEDMAN, M. F. (1988), "Decision Analysis and Optimality in Air Traffic control conflict resolution", *TRANSPORTATION RESEARCH B*, 22B(3), pp: 207-216.

FRIEZE, A.; , GALBIATI, G. and MAFFIOLI, F. (1982), "On the worst-case performance of some algorithms for the asymmetric traveling salesman problem", *NETWORKS* 12(1), pp: 23-39.

GARCIA-DIAZ (1985), "A heuristic circulation-network approach to solve the multi-travelling salesman problem", *NETWORKS* 15, pp: 455-467.

GAREY, M. R. and JOHNSON, D. S. (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness", *W. H. FREEMAN AND COMPANY-NY*.

GARFINKEL, R. (1973), "On Partitioning the feasible set in a Branch-and-Bound Algorithm for the Asymmetric Traveling Salesman Problem", *OPERATIONS RESEARCH* 2(1), pp: 340-342.

GARFINKEL, R. (1977), "Minimizing Wallpaper Wast, Part I: A class of traveling salesman problems", *OPERATIONS RESEARCH* 25(5), pp: 741-751.

GARFINKEL, R. and GILBERT, K. (1978), "The Bottleneck Traveling Salesman Problem: Algorithms and Probabilistic Analysis". *J. ACM* 25(3), pp: 435-448.

GARVIN, W.; CRANDALL, H.; JOHN, J. and SPELLMAN, R. (1957), "Applications of vehicle routing in the oil industry", *MANAGEMENT SCIENCE* 3, pp: 407-430.

GASKELL, T. (1967), "Bases for vehicle fleet scheduling", *OP. RES. QUART.* 18, pp: 281-295.

GAVISH, B. (1976), "A note on the formulation of the M-salesman traveling salesman problem", *MANAGEMENT SCIENCE*, 22, pp: 704-705.

GAVISH, B. (1978), "On obtaining the "best" multipliers for a lagrangean relaxation for integer programming", *COMPUTER & OPERATIONS RESEARCH* 5, pp: 55-71.

GAVISH, B. and GRAVES, S. (1978), "The Traveling Salesman Problem and Related Problems", *WORKING PAPER, GRADUATE SCHOOL OF MANAGEMENT, UNIVERSITY OF ROCHESTER.*

GAVISH, B. and SCHWEITZER, P. (1974), "An Algorithm for Combining Truck Trips", *TRANSP. SCIENCE* 8, pp: 13-23.

GAVISH, B.; SCHWEITZER, P. and SHLIFER, E. (1978), "Assigning Buses to Schedules in a Metropolitan Area", *COMPUTER & OPERATIONS RESEARCH* 5, pp: 129-138.

GENSCH, D. (1978), "An Industrial Application of the

Traveling Salesman's subtours problems", *AIIE TRANS.* 10(4), pp: 362-370.

GEOFFRION, A. (1974), "Lagrangian Relaxation for Integer Programming", *MATHEMATICAL PROGRAMMING STUDY 2*, pp: 82-114.

GILLETT, B. and JOHNSON, J. (1976), "Multi-terminal vehicle-dispatch algorithm", *OMEGA* 4, pp: 711-716.

GILLETT, B. and MILLER, L. (1974), "A heuristic algorithm for the vehicle dispatch problem", *OPERATIONS RESEARCH* 22, pp: 340-349.

GILMORE, P. and GOMORY, R. (1964), "Sequencing a one State-Variable machine: a solvable case of the traveling salesman problem", *OPERATIONS RESEARCH*, 12(5), pp: 655-679.

GOLDEN, B. (1977a), "Evaluating a Sequential Vehicle Routing Algorithm", *AIIE TRANS.* 9, pp: 204-208.

GOLDEN, B. (1977b), "A statistical approach to the TSP", *NETWORKS* 7, pp: 209-225.

GOLDEN, B. and ASSAD, A.A. (1986), "Perspectives on Vehicle Routing: Exciting new Developments", *OPERATIONS RESEARCH* 34(5), pp: 803-810.

GOLDEN, B. and ASSAD, A.A. (1988), "Vehicle Routing: Methods

and Studies", *NORTH-HOLLAND*.

GOLDEN, B.; ASSAD, A.A.; LEVY, L. and GHEYSENS, F. (1984),
 "The fleet size and mix vehicle routing problems",
MANAGEMENT SCIENCE & STATISTICS, WORKING PAPER 82-020, UNIV.
OF MAYLAND at COLLEGE PARK.

GOLDEN, B.; BODIN, L.; DOYLE, T.; STEWART, W. (1980),
 "Approximate Traveling Salesman Algorithms", *OPERATIONS*
RESEARCH, 28(3), pp: 694-711.

GOLDEN, B.; BODIN, L. and GOODWIN, T. (1986),
 "Microcomputer-Based Vehicle Routing and Scheduling
 Software", *COMPUTER & OPERATIONS RESEARCH 13(2 e 3), pp:*
277-285.

GOLDEN, B.; DE-ARMON, J. and BAKER, E. (1981),
 "Computational experiments with algorithms for a class of
 routing problems", *MANAG. SCIENCE & STATISTICS, WORKING*
PAPER 81033 UNIV. OF MAYLAND at COLLEGE PARK.

GOLDEN, B.; LEVY, L. and DAHL, R. (1981), "Two
 Generalizations of the Traveling Salesman Problem", *OMEGA*
9(4), pp: 439-441.

GOLDEN, B.; MAGNANNTI, T. and NGUYEN, H. (1977),
 "Implementing vehicle routing algorithm", *NETWORKS, 7,*
pp: 113-148.

GOLDEN, B. and SKISCIM (1986), "Using Simulated Annealing to Solve Routing and Location Problems", *NAVAL RESEARCH LOGIST. QUARTELY*, 33, pp: 261-279.

GOLDEN, B. and WASIL, E. A. (1987), "Computerized Vehicle Routing in the Soft Drink Industry", *OPERATIONS RESEARCH*, 35(1), pp: 6-17.

GOLDEN, B. and WONG, R. (1981), "Capacited arc Routing Problems", *NETWORKS* 11(3), pp: 305-315.

GOLDEN, B. and YEE, J. (1979), "A framework of probabilistic vehicle routing", *AIIE TRANS.* 11(2), pp: 109-112.

GOMORY, R. E. (1958), "Outline of a Algorithm for Integer Solutions to Linear Program", *BULL. AMER. MATH. SOC.* pp: 275-278.

GRAHAN, D. and NUTLE, H. L. W. (1986), "A comparison of heuristics for a school bus scheduling problems", *TRANSPORTATION RESEARCH B*, 20(2), pp: 175-182.

GROTSCHEL, M. (1980a), "On the Symmetric Traveling Salesman Problem: Solution of 120 city problem", *MATHEMATICAL PROGRAMMING STUDY* 12, pp: 61-77.

GROTSCHEL, M. (1980b), "On the monotone Symmetric Traveling

Salesman Problem: Hypohamiltonian hypotractable graph and facets", *MATH. OPS. RES.* 5, pp: 285-292.

GROTSCHER, M. and PADBERG, M. (1979), "On the Symmetric Traveling Salesman Problem I: Inequalities", *MATHEMATICAL PROGRAMMING* 16(3), pp: 265-280.

GUINARD, M. and KIM, S. (1987a), "A Model Yielding stronger Lagrangean bounds", *MATHEMATICAL PROGRAMMING* 39, pp: 215-228.

GUINARD, M. and KIM, S. (1987b), "Lagrangean Decomposition for Integer Programming: Theory and Applications", *TECHNICAL REPORT 93, UNIV. OF PENNSYLVANIA.*

GUPTA, J. (1978), "A Search Algorithm for the traveling Salesman Problem", *COMPUTERS & OPERATIONS RESEARCH* 5(4), pp: 243-250.

HADLEY, G. (1964), "Nonlinear and Dynamic Programming", *ADDISON WESLEY, READING MA*, pp: 267-296.

HALTON, J. and TERADA, R. (1982), "A fast algorithm for the Euclidean Traveling Salesman Problem Optimal with Probability one", *SIAM J. COMP.* 11(1), pp: 28-46.

HANSEN, K. and KRARUP, J. (1974), "Improvements of the Held-Karp Algorithm for the Symmetric Traveling Salesman

Problem", *MATHEMATICAL PROGRAMMING* 7, pp: 87-96.

HARDGRAVE, W. and NEMHAUSER, G. (1962), "On the Relation Between the Traveling Salesman Problem and the Longest Path Problem", *OPERATIONS RESEARCH* 10, pp: 647-657.

HAYER, E. (1971), "Fleet selection for Public Transportation Routes", *TRANSPORTATION SCIENCES*, 5(1), pp: 1-21.

HELD, M.; HOFFMAN, A. J.; JOHNSON, E. L. and WOLPE, P. (1964), "Aspects of the Traveling Salesman Problem", *IBM J. RES. DEVELOP.* 28(4), pp: 476-486.

HELD, M. and KARP, R. (1962), "A dynamic programming approach to sequencing problems", *SIAM J.* 10, pp: 196-210.

HELD, M. and KARP, R. (1970), "The Traveling Salesman Problem and Minimum Spanning Trees", *OPERATIONS RESEARCH* 18, pp: 1138-1162.

HELD, M. and KARP, R. (1971), "The Traveling Salesman Problem and Minimum Spanning Trees: Part II", *MATHEMATICAL PROGRAMMING* 1, pp: 6-25.

HELLER, I. (1955), "On the Traveling Salesman Problem", *THE GEORGE WASHINGTON UNIVERSITY (SPEECH BEFORE THE SYMPOSIUM ON LINEAR PROGRAMMING)*.

HOFFMAN, A.J. and WOLFE, P. (1985), "History-PCV", The Traveling Salesman Problem, pp:1-14, JOHN WILEY & SONS EDITED BY E.L.LAWLER; J.K.LENSTRA; A.H.G.RINNOOY KAN; D. B. SHMOYS.

HOLMES, R. and PARKER, R. (1976), "A Vehicle Scheduling Procedure Based upon savings and a solution perturbation scheme", OPERATIONS RESEARCH QUART. 27(1), pp: 83-92.

HONG, S. and PADBERG, M. (1977), "A note on the symmetric multiple traveling salesman problem with fixed charges", OPERATIONS RESEARCH 25(5), pp: 871-874.

HOOKER, J.N. (1988), "Optimal Driving for Single-Vehicle Fuel economy", TRANSPORTATION RESEARCH A, 22A(3), pp: 183-201.

ISSAC, A. and TURBAN, E. (1969), "Some Coments on the Traveling Salesman Problem", OPERATIONS RESEARCH. 17(3), pp: 543-546.

JACOBSEN, S.K. and MADSEN, O.B.G. (1978), "On the location of transfer points in a two-level newspaper delivery system : a case study", RESEARCH REPORT 5, IMSOR, UNIV. OF DENMARK.

JACOBSEN, S.K. and MADSEN, O.B.G. (1980), "A comparative study of heuristic for a two-level routing-location problem", EUROPEAN JOURNAL OF OPERATIONAL RESEARCH 5, pp:

378-387.

JAW, J.; ODoni, A. R.; PSAFTIS, H. N. and WILSON, N. A. (1986), "A heuristic algorithm for the multi-vehicle advance request Dial-a-Ride problem with time-windows", *TRANSPORTATION RESEARCH B*, 6,; pp: 243-257.

JEROMIN, B. and KORNER, F. (1985), "On the refinement of bounds of heuristic algorithm for the traveling salesman problem", *MATHEMATICAL PROGRAMMING*, 32, pp; 114-117.

JONGENS, K. and VOLGENANT, T. (1985), "The symmetric clustered traveling salesman problem", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 19, pp: 68-75.

JONKER, R.; DELEVE, G.; DERVERDE, J. V. and VOLGENAT, A. (1980), "Bounding symmetric traveling salesman problems with an asymmetric assignment problem", *OPERATIONS RESEARCH* 28, pp: 623-627.

JORNSTEN, K. and NASBERG, M. (1986), "A New Lagrangian Relaxation approach to the generalized assignment problem", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, 27, pp: 313-323.

KANELLAKIS, P. and PAPANIMITRIOU, C. (1980), "Local search for the asymmetric traveling salesman problem", *OPERATIONS RESEARCH*, 28(5), pp: 1086-1099.

KAO, E. (1978), "A Preference order dynamic program for a stochastic traveling salesman problem", *OPERATIONS RESEARCH* 26 (6), pp: 1033-1045.

KARG, L. and THOMPSON, G. (1964), "A heuristic approach to solving traveling salesman problems", *MANAGEMENT SCIENCE*, 10 (2), pp: 225-248.

KARP, R. (1972), "Reducibility among combinatorial problems", R.E. MILLER; J.W. THATCHER (EDIT), *COMPLEXITY OF COMPUTER COMPUTATIONS*, PLENUM PRESS, NY pp: 85-103.

KARP, R. (1975), "On the computational complexity of combinatorial problems", *NETWORKS*, 5, pp: 45-68.

KARP, R. (1977), "Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane", *MATH. OPS. RES.* 2, pp: 209-224.

KARP, R. (1979), "A patching algorithm for the nonsymmetric traveling salesman problem", *SIAM J. COMP.* 8, pp: 561-573.

KIRBY, R. and MACDONALD, J. (1973), "The savings method for vehicle scheduling", *OPERATIONS RESEARCH QUART.* 24(2), pp: 305.

KNIGHT, K. and HAFER, J. (1968), "Vehicle scheduling with timed and connected calls: a case study", *OPERATIONS*

RES. QUART. 19(3), pp: 299.

KONER, F. (1986), "On the relationship of approximation algorithms for the minimum and the maximum traveling salesman problem", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 26, pp: 262-265.

KROLAK, P.; FELTS, W. and MARBLE, G. (1971), "A man-machine approach toward solving the traveling salesman problem", *COMMUN. ACM* 14, pp: 327-334.

KRUSKAL, J. (1956), "On the shortest spanning subtree of a graph and the traveling salesman problem", *PROC. AM. MATH. SOC.* 2, pp: 48-50.

KULKARNI, R.V. and BHAVE, P.R. (1985), "Integer Programming formulation of vehicle routing problems", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 20, pp: 58-67.

LABLÉ, M. and LAPORTE, G. (1986), "Maximizing user convenience and postal service efficiency in post box location", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

LADANY, S. and HERSH, M. (1977), "Non-stop Vs one stop flights", *TRANSPORTATION RESEARCH* 11(3), pp: 155-159.

LADANY, S. and MEHREZ, A. (1984), "Optimal routing of a

single vehicle with loading and unloading constraints", *TRANSP. PLANNING and TECH.* 8, pp: 301-306.

LAND, A. and DOIG, A. (1960), "An Automatic Method of Solving Discrete Programming Problems", *ECONOMETRICA* 28(3), pp: 497-520.

LAPORTE, G. (1982), "An Integer Linear Programming Approach to the Vehicle Scheduling Problem", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

LAPORTE, G. (1989), "A Survey of Algorithms for Location-Routing Problems", *INVESTIGACIÓN OPERATIVA VOL. 1(2)*.

LAPORTE, G.; CHAPLEAU, S.; LANDRY, P.E. and MERCURE, H. (1987), "A routing algorithm for postal trucks in urban areas", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

LAPORTE, G. and DEJAX, P. (1988), "Dynamic Location-Routing Problems", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (578)*.

LAPORTE, G.; MERCURE, H. and NOBERT, Y. (1985), "Finding the shortest hamiltonian circuit through n clusters: a lagrangean relaxation approach", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (392)*.

LAPORTE, G.; MERCURE, H. and NOBERT, Y. (1986), "An exact algorithm for the asymmetrical capacited vehicle routing problem", *NETWORKS*, 16, pp: 33-46.

LAPORTE, G.; MERCURE, H. and NOBERT, Y. (1985), "Generalized traveling salesman problem through n sets of nodes. the asymmetrical case", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

LAPORTE, G.; NGUYEN, T. and NOBERT, Y. (1985), "A branch-and-bound algorithm for the asymmetrical distance constrained vehicle routing problem", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (393)*.

LAPORTE, G. and NOBERT, Y. (1980a), "A cutting planes algorithm for the m-salesman problem", *J.OP.RES.SOC.* 31, pp: 1017-1023.

LAPORTE, G. and NOBERT, Y. (1980b), "Subtour elimination algorithm for the symmetrical traveling salesman problem and its variants", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS (161)*.

LAPORTE, G. and NOBERT, Y. (1981), "An exact algorithm for minimizing routing and operating costs in depot location", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH 6(2)*, pp: 224-226.

LAPORTE, G.; NOBERT, Y. and ARPIN, D. (1984), "Optimal solutions to capacited multidepot vehicle routing problems", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (355).

LAPORTE, G.; NOBERT, Y. and MERCURE, H. (1980), "The multi-depot traveling salesman problem", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (182).

LAPORTE, G.; NOBERT, Y. and PELLENTIER, P. (1980), "Generalized traveling salesman problem through n sets of nodes: an integer programming approach", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (197).

LAPORTE, G.; NOBERT, Y. and TAILLEFER, S. (1988), "Solving a family of multi-depot vehicle routing and location routing problems", *TRANSPORTATION SCIENCE*, 22(3).

LARSON, R.C. (1988), "Transporting sludge to the 106-mile site: An inventory routing model for fleet sizing and logistics system design", *TRANSPORTATION SCIENCE* 22(3).

LAVIE, S. (1985), "Optimisation des horaires D'équipages D'avions", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

LAVOIRE, S.; MINOUX, M. and ODIER, E. (1985), "A new approach of crew pairing problems by column generation and

application to air transports", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

LAWER, E. L. (1971), "A solvable case of the traveling salesman problem", *MATHEMATICAL PROGRAMMING* 1, pp: 267-269.

LAWER, E. L.; LENSTRA, J. K.; RINNOOY KAN, A. H. G. and SHMOYS, D. B. (1985), "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization", *JOHN WILEY & SONS*.

LEBEUF, D. (1985), "L'allocation des appareils pour un horaire aérien: Une approche interactive-graphique", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (415).

LEIPALA, T. (1978), "On the solutions of stochastic traveling salesman problems", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 2(4), pp: 291-297.

LENSTRA, J. and RINNOOY KAN (1976), "On general routing problems", *NETWORKS*. 6(3), pp: 273-280.

LENSTRA, J. and RINNOOY KAN (1979), "A characterization of linear admissible transformation for the m -traveling salesman problem: a result of BERENGUER", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 3(3), pp: 250-252.

LENSTRA, J. and RINNOOY KAN (1981), "Complexity of vehicle

routing and scheduling problem", *NETWORKS*, 11(2), pp: 221-227.

LEVARY, R. (1981), "Heuristic Vehicle Scheduling", *OMEGA*. 9, pp: 660-663.

LEVIN, A. (1966), "Crew scheduling on a transportation network", *M.S. THESIS, DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS M.I.T., CAMBRIDGE, MASS.*

LEVIN, A. (1971), "Scheduling and fleet routing models for transportation systems", *TRANSPORTATION SCIENCE* 5, pp: 232-255.

LIN, S. (1975), "Heuristic programming as an aid to network design", *NETWORKS* 5, pp: 33-44.

LIN, S. and KERNIGHAN, B. (1973), "An effective heuristic algorithm for the traveling salesman problem", *OPERATIONS RESEARCH* 21, pp: 498-516.

LITTLE, J.; MURTY, K.; SWEENEY, D. and KAREL, C. (1963), "An algorithm for the traveling salesman problem", *OPERATIONS RESEARCH* 11(6), pp: 972-989.

MAGNANT, T. (1981), "Combinatorial Optimization and Vehicle fleet planning: Perspectives and Prospects", *NETWORKS*, 11(2), pp: 179-214.

MALE, J.; LIEBERMAN, J. and ORLOFF, C. (1977), "An improvement of Orloff's general routing problem", *NETWORKS* 7(1), pp: 89-92.

MARQUEZ, J.D.C. and ESCALANTE, O. (1977), "A Network solution to a general vehicle scheduling problem", *EUROPEAN JOURNAL OF OPERATIOAL RESEARCH* 1(4), pp: 255-261.

MARSTEN, R. and SHEPARDSON, F. (1981), "Exact solutions of crew scheduling problems using the set partitioning model: recent sucessfull application", *NETWORKS* 11(2), pp: 167-177.

MARTIN-LOF, A. (1970), "A branch-and-bound algorithm for determining the minimal fleet size of a transportation system", *TRANSPORTATION SCIENCE* 4(2), 159-163.

MARTIN, R. K.; SWEENEY, D. J. and DOLTERTY, M. E. (1985), "The reduced cost branch-and-bound algorithm for mixed integer programming", *COMPUTER & OPERATIONS RESEARCH* 12(2), pp: 139-149.

McBRIDE, R. (1982), "Controlling left and U-turns in the routing of refuse collection vehicles", *COMPUTERS & OPERATIONS RESEARCH* 9(2), pp: 145-152.

McLAIN, D. R.; DURCHHOLZ, M. L. and WILBORN, W. B. (1984), "U.S.A.F. EDSA, Routing and Operating Location Selection

Study", *REPORT XPSR84-3, OPS.RES.DIV., SCOTT AIR BASE, IL.*

MEHREZ, A. and STERN, H.I. (1985), "Optimal refueling strategies for a mixed-vehicle fleet", *NAVAL RESEARCH LOG. QUART.* 32, pp: 315-328.

MEI-KO, K. (1962), "Graphic programming using odd or even points", *CHINESE MATH.* 1, pp: 273-277.

MILLER, G; TUCKER, A. and ZEMLIN, R. (1960), "Integer Programming formulation of travelling salesman problems", *I. ASS. COMPUT. MACH.* 7, pp: 326-332.

MILIOTIS, P. (1976), "Integer Programming approaches to the travelling salesman problem", *MATHEMATICAL PROGRAMMING* 10, pp: 367-378.

MILIOTIS, P. (1978), "Using cutting planes to solve the symmetric traveling salesman problem", *MATHEMATICAL PROGRAMMING* 15, pp: 177-188.

MINIEKA, E. (1979), "The chinese postman problem for mixed networks", *MANAGEMENT SCIENCE*, 25, pp: 643-648.

MOLE, R.H. (1979), "A survey of local delivery vehicle routing methodology", *J. OPS. RES. SOC.* 30, pp: 245-252.

MOLE, R.H. (1983), "The curse of unintended routing error: a

case from the vehicle scheduling literature", *J. OPS. RES. SOC.* 34(7), pp: 607-613.

MOLE, R. H. and JAMENSON, S. (1976), "A Sequential route-building algorithm employing a generalized savings criterion", *OPER. RES. QUART.* 27, pp: 503-511.

MOON, J. (1982), "The urban postman", *OMEGA* 10, pp: 334-337.

MORTAGY, A. K. and ELEITHY, S. A. (1980), "Optimizing Transportation Service for Oil exploration", *INTERFACES*, 10(2), pp: 13-19.

NAMBIAR, J. M; GELDER, L. F. and WASSENHOVE, L. N. V. (1989), "Plant Location and Vehicle Routing in the Malaysian Rubber Smallholder Sector: A case study", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 38, pp: 14-26.

NEMHAUSER, G. (1969), "Scheduling local and express trains", *TRANSPORTATION SCIENCE* 3, pp: 164-175.

NICHOLSON, T. (1968), "A boundary method for planar traveling salesman problem", *OPS. RES. QUART.* 19, pp: 444-452.

NYGARD, K. E.; GREENBERG, P.; BOLKAN, W. E. and SWENSON, E. J. (1988), "Generalized Assignment Methods for the Deadline vehicle Routing Problem", *VEHICLE ROUTING: METHODS AND STUDIES* (edited by B. L. GOLDEN and A. A. ASSAD), NORTH-HOLLAND,

pp: 107-126.

NOBERT, Y. (1982), "Construction D'algorithm es optimaux pour des extensions au problème du voyageur de commerce", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

NORBACK, J. and LOVE, R. (1977), "Geometric approaches to solving the traveling salesman problem", *MANAGEMENT SCIENCE* 23, pp: 1208-1223.

NORBACK, J. and LOVE, R. (1979), "Heuristic for the Hamiltonian path problem in euclidean two space", *J. OPS. RES. SOC.* 30, pp: 363-368.

OCHI, L.S. (1988), "O Problema de Roteamento e Scheduling de veículos com restrições "time-windows", *ANAIS DO VII ENEGEP. -NITERÓI-RJ*.

OCHI, L.S. (1989), "O Problema do Viajante de Compras : Formulações, Relaxações e Soluções", *ANAIS DO XII CNMAC - S. J. DO RIO PRETO-SP*.

OCHI, L.S. e MACULAN, N. (1988), "Relaxações Lineares, Lagrangeanas e Decomposições Lagrangeanas: Relações entre os limites inferiores de duas formulações do problema de roteamento de navios com restrições time-windows", *ANAIS DO XI CNMAC OURO PRETO-MG*.

OCHI, L.S. e MACULAN, N. (1989). "Routing and Scheduling of Tanker Fleet with Time-Windows", "ALIO-EURO WORKSHOP ON PRATICAL COMBINATORIAL OPTIMIZATION", RJ.

OR, I. and PIERSKALLA, W. P. (1979), "A transportation location-allocation model for the regional blood banking", *AIIE TRANSAC.* 11, pp: 86-95.

ORLOFF, C. (1974), "Routing a fleet of M vehicles to from a central facility", *NETWORKS* 4, pp: 147-162.

ORLOFF, C. (1976a), "On general routing problems: coments", *NETWORKS* 6(3), pp: 281-284.

ORLOFF, C. (1976b), "Route constrained fleet scheduling", *TRANSPORTATION SCIENCE*, 10, pp: 149-168.

ORLOFF, C. and CAPRERA, D. (1976), "Reduction and solution of large scale vehicle routing problems", *TRANSPORTATION SCIENCE* 10(4), pp: 361-375.

PADBERG, M. and HONG, S. (1980), "On the symmetric traveling salesman problem: a computational study", *MATHEMATICAL PROGRAMMING STUDY* 12, pp: 78-107.

PALESTRA, G. (1987), "A heuristic algorithm for a pseudo-cyclic delivery problem under window constraints",

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH, 31, pp: 71-77.

PAPADIMITRIOU, C. (1977), "The Euclidean Travelling Salesman Problem is NP-Complete", *T.COMP.SCI.* 4(3), pp: 237-244.

PAPADIMITRIOU, C. (1978), "The adjacency relation on the traveling salesman polytope is NP-Complete", *MATHEMATICAL PROGRAMMING*, 14(3), pp: 312-324.

PAPADIMITRIOU, C. and STEIGHITZ, K (1977), "The complexity of local search for the traveling salesman problem", *SIAM J. COMP.* 6(1), pp: 76-83.

PAPADIMITRIOU, C. and STEIGHITZ, K. (1978), "Some examples of difficult traveling salesman problems", *OPERATIONS RESEARCH*, 26, pp: 434-443.

PEARN, W. L. (1988), "New lower bounds for the capacited arc routing problems", *NETWORKS*, 118, pp: 181-191.

PICARD, J. and QUEYRANNE, M. (1978), "The time-dependent traveling salesman problem, and its application to the tardiness problem in one-machine scheduling", *OPERATIONS RESEARCH* 26(1), pp: 86-110.

PIERCE, J. (1968), "Application of combinatorial programming algorithm for a class of all zero-one integer programming problems", *MANAGEMENT SCIENCE* 15, pp: 191-209.

POLLACK, M. (1977), "Some elements of the airline fleet planning problem", *TRANSP. RES.* 11, pp: 301-310.

PSARAFTIS, H. (1980), "A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem", *TRANSP. SCIENCE*, 2, pp: 130-154.

PSARAFTIS, H. (1988), "Dynamic vehicle routing problems", *VEHICLE ROUTING: METHODS AND STUDIES* (edited by B.L.GOLDEN and A.A. ASSAD), NORTH-HOLLAND pp: 223-248.

PSARAFTIS, H.; THARAKAN, G.G. and CEDER (1986), "Optimal response to oil spills: The strategic decision case", *OPERATIONS RESEARCH* 34(2), pp: 203-217.

RANA, K. and VICKSON, R. G. (1988), "A model and solution algorithm for optimal routing of a Time-Chartered Containership", *TRANSPORTATION SCIENCE*, 22(2).

RAO, M. (1980), "A note on the multiple traveling salesman problem", *OPERATIONS RESEARCH* 28(3), pp: 628-632.

REINOSO, H.A. e MACULAN, N. (1988), "Um novo esquema de Decomposição Lagrangeana em Programação Linear Inteira", *RELATÓRIO INTERNO-COPPE-UFRJ. ES/141/88.*

RICHARDSON, R. (1976), "An optimization approach to routing

aircraft", *TRANSPORTATION SCIENCE* 10, pp: 52-71.

ROBERTS, S. and FLORES, B. (1966), "An engineering approach to the traveling salesman problem", *MANAGEMENT SCIENCE* 13(3), pp: 269-288.

RONEN, D. (1983), "Cargo ship routing and scheduling: survey of models and problems", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 12, pp: 119-126.

RONEN, D. (1986), "Short-term scheduling of vessels for shipping bulk or semi-bulk commodities originating in a single area", *OPERATIONS RESEARCH* 34(1), pp: 164-173.

RONEN, D. (1988), "Perspectives on practical aspects of truck routing and scheduling", *EUROPEAN JOURNAL OPERATIONAL RESEARCH* 35, pp: 137-145.

ROSENKRANTZ, D.; STERN, R. and LEWIS, P. (1977), "An analysis of several heuristics for the traveling salesman problem", *SIAM J. COMP.* 6, pp.: 563-581.

ROY, S.; ROUSSEAU, J. M.; LAPALME, G. and FERLAND, J. (1984a), "Routing and scheduling of transportation services for the disabled: summary report", *UNIVERSITÉ DE MONTRÉAL CENTRE DE RECHERCHE SUR LES TRANSPORTS (473A)*.

ROY, S.; ROSSEAU, J. M.; LAPALME, G. and FERLAND, J. (1984b),

"Élaboration des routes et horaires de véhicules pour le transport adapté: rapport sommaire", *UNIVERSITÉ DE MONTRÉAL, CENTRE DE RECHERCHE SUR LES TRANSPORTS (473f)*.

RUSSEL, R. (1977), "An effective heuristic for the M-tour traveling salesman problem with some side conditions", *OPERATIONS RESEARCH 25(3)*, pp: 517-524.

RUSSEL, R. and IGO, W. (1979), "An assignment routing problem", *NETWORKS 9(1)*, pp: 1-17.

SÁ, G. (1969), "Diálogo sobre Programação Matemática generalizada e o problema de despacho de carga, *IPEA-RJ*.

SAHA, J. (1970), "An algorithm for the bus scheduling problems", *OPERATIONS RESEARCH QUART. 21*, pp: 463-474.

SALHI, S. and RAND, G. K. (1987), "Improvements to vehicle routing heuristics", *J. OPS. RES. SOC. 38(3)*, pp: 293-295.

SALZBORN, F. (1970), "The minimal fleet size for a suburban railways system", *TRANSPORTATION SCIENCE 4*, pp: 383-402.

SAUNDERS, C.J. and KIRK, J. (1985), "O.R. in the company car fleet: when simple is efficient", *J. OPS. SOC. 36(8)*, pp: 713-729.

SCOTT, D.W. (1984), "A method for scheduling urban transit

vehicles with takes account of operating labor cost", *THESE DE DOCTORAT-CARNEGIE MELLON UNIVERSITY, PITTSBURGH.*

SCOTT, D.W. (1986), "Minimal fleet size in transshipment type vehicle scheduling problems", *UNIVERSITÉ DE MONTRÉAL, CENTRE DE RECHERCHE SUR LES TRANSPORTS.*

SCHRAGE, L. (1981), "Formulation and structure of more complex/realistic routing and scheduling problems", *NETWORKS 11(2), pp: 229-232.*

SCHULTZ, H. (1979), "A practical method for vehicle scheduling", *INTERFACES 9(3), pp: 13-19.*

SCULLI, D.; MOK, K.C. and CHEUNG, S.H. (1987), "Scheduling vehicles for refuse collection", *J.OPS.RES.SOC. 38(3), pp: 233-239.*

SCHERALI, H.D. and RIOS, M. (1984), "An air force crew allocation and scheduling problem", *J.OPS.RES.SOC. 35(2), pp: 91-103*

SINCLAIR, M. and DYK, E.V. (1987), "Combined routing and scheduling for the transportation of containerized cargo", *J.OPS.RES.SOC. 38(6), pp: 487-498.*

SKITT, R.A. and LEVARY, R.R. (1985), "Vehicle routing with column generation", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*

21, pp: 65-76.

SMITH, T. (1980), "A LIFO implicit enumeration algorithm for the asymmetric traveling salesman problem using a one-arborescence relaxation", *MATHEMATICAL PROGRAMMING STUDY*, 12, pp: 108-114.

SMITH, T. (1988), "IMPACS-A bus crew scheduling system using integer programming", *MATHEMATICAL PROGRAMMING* 42, pp: 181-187.

SMITH, M.L.; METZNER, H.E. and BROOKS, W.A.; "Computerized costing model for inland waterways barge transportation", *INTERFACES*, 8(4), pp: 311-34.

SMITH, T; SRINIVASAN, V. and THOMPSON, G.L. (1977), "Computational performance of three subtour elimination algorithms for solving asymmetric traveling salesman problems", *STUDIES IN INTEGER PROGRAMMING* (edited by P.H.HAMMER, E.L.JOHNSON, B.H.KORTE, G.L.NENHAUSER), pp: 495-506.

SMITH, T. and THOMPSON, G.L. (1977), "A LIFO implicit enumeration search algorithm for the symmetric traveling salesman problem using HELD and KARP's i-tree relaxation", *STUDIES IN INTEGER PROGRAMMING*, pp: 479-494.

SOLOMON, M.M. (1986), "On the worst-case performance of some

heuristics for the vehicle routing and scheduling problem with time-window constraints", *NETWORKS* 16, pp: 161-174.

SOLOMON, M.M. (1987), "Algorithms for the vehicle routing and scheduling problem with time-window constraints", *OPERATIONAL RESEARCH* 35, pp: 254-265.

SOLOMON, M.M.; BAKER, E.K. and SCHAFFER, J.R. (1988), "Vehicle routing and scheduling problems with time-window constraints: efficient implementations of solution improvement procedures", *VEHICLE ROUTING: METHODS AND STUDIES* (edited by GOLDEN and ASSAD), NORTH-HOLLAND, 85-106.

SOUMIS, F.; ÉTHIER, J. and McINNIS, D. (1987), "An optimization algorithm for trucks dispatching in an open pit mine", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS*.

SOUMIS, F.; FERLAND, J. and ROUSSEAU, J. (1979), "MAPUN: A model for assigning passengers to a flight schedule", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (142).

SOUMIS, F.; FERLAND, J. and ROUSSEAU, J. (1980), "A model for large scale aircraft routing and scheduling problems", *TRANSPORTATION RESEARCH* 14B, pp: 191-201.

STEIN, D. (1978a), "An asymmetric probabilistic analysis of

a routing problem", *MATH. OPS. RES.* 3, pp: 89-101.

STEIN, D. (1978b), "Scheduling dial-a-ride transportation systems", *TRANSPORTATION SCIENCE* 12, pp: 232-249.

STERN, H. (1980), "Bus and crew scheduling", *TRANSPORTATION RESEARCH* 14A, pp: 154.

STERN, H. (1983), "An improved lower bound to the minimum fleet size problem", *TRANSPORTATION SCIENCE* 17(4).

STERN, H. and DROR, M. (1979), "Routing electric meter readers", *COMPUTERS & OPERATIONS RESEARCH* 6, pp: 209-223.

STERN, H. and HERSH, M. (1980), "Scheduling aircraft cleaning crews", *TRANSPORTATION SCIENCE*, 14(3), pp: 277-291.

STEWART, W.R. (1985), "An improved assignment lower for the euclidean traveling salesman problem", *OPERATIONS RES. LETTERS*, 4(2), pp: 55-60.

STEWART, W.R. and GOLDEN, B. (1984), "A Lagrangean relaxation heuristic for vehicle routing", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 15, pp: 84-88.

STODOLSKY, D.; PEKNY, J.F. and MILLER, D.L. (1988), "An exact algorithm for finding Hamiltonian cycles in directed graphs", *EDRC 05-26-88, CARNEGIE-MELLON UNIVERSITY*,

PITTSBURGH, PA 15213.

STOTT, K.L. Jr. and DOUGLAS, B.W. (1981), "A model-based decision support system for planning and scheduling ocean-borne transportation", *INTERFACES*, 11(4), pp: 1-9.

SVESTKA, J. (1978), "A continuous variable representation of the traveling salesman problem", *MATHEMATICAL PROGRAMMING* 15(2), pp: 211-213.

SVESTKA, J. and HUCKFELDT, V. (1973), "Computational experience with and M-Salesman traveling salesman algorithm", *MANAGEMENT SCIENCE*, 19(7), pp: 790-799.

SWERSEY, A. and BALLARD, W. (1984), "Scheduling school buses", *YALE SCHOOL OF ORG. and MANAG. TECH. REPORT*.

SYSLO, M. (1973), "A new solvable case of the traveling salesman problem", *MATHEMATICAL PROGRAMMING* 4, pp: 347-348.

SZWARCFITER, J.L. (1985), "An ILP formulation of the many-visits traveling salesman problem", *NCE-UFRJ, BRASIL*.

TEODOROVIC, D. (1985), "A model for designing the meteorologically most reliable airline schedule", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 21, pp: 156-164.

TEODOROVIC, D. and GUBERINIC, S. (1984), "Optimal

dispatching strategy on an airline network after a schedule perturbation", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 15, pp: 178-182.

THOMAS, R. S. D. and WELLS, J. M. (1980), "Multiple-origin single destination transit routing", *INTERFACES*, 10(2).

TIBREWALA, R.; PHILIPPE, D. and BROWNE, J. (1972), "Optimal scheduling of two consecutive idle periods", *MANAGEMENT SCIENCE* 10, pp: 71-75.

TILLMAN, F. and CAIN, T. (1972), "An upper bounding algorithm for the single and multiple terminal delivery problem", *MANAGEMENT SCIENCE* 18, pp: 664-682.

TILLMAN, F. and COEHRAN, H. (1968), "A heuristic approach for solving the delivery problem", *I. IND. ENG.* 19, pp: 354-358.

TRINH, K. N. (1987), "DEUX NOUVEAUX ALGORITHMES EXACTS POUR LE PROBLÈME DES N VOYAGEURS DE COMMERCE", *UNIVERSITÉ DE MONTRÉAL-CENTRE DE RECHERCHE SUR LES TRANSPORTS* (570).

TYAGI, M. (1968), "A practical method for the truck dispatching problem", *J. OPS. SOC. JAPAN* 10, pp: 76-92.

TZENG, G. H. and SHIAN, T. A. (1988), "Multiple objective programming for bus operation: A case study for TAIPE CITY",

TRANSPORTATION RESEARCH B, 22B(3), pp: 195-206.

VAN-DER CRUYSSSEN, P. and RIJCKAERT (1978), "Heuristic for the asymmetric traveling salesman problem", *J. OPS. RES. SOC.* 30, pp: 697-701.

VANDERMONDE, A.T. (1771), "Remarques sur les problèmes de situation", *HISTORIE DE L'ACADEMIE DES SCIENCES (PARIS)*, pp: 566-574.

WALTERS, C.D.J. (1984a), "Interactive vehicle routing", *J. OPS. RES. SOC.* 35(9), pp: 821-826.

WALTERS, C.D.J. (1984b), "Vehicle scheduling revisited", *J. OPS. RES. SOC.* 35(2), pp: 145-148.

WATSON-GANDY, C.D.T. and DOHRN, P.J. (1973), "Depot location with van salesman a practical approach", *OMEGA* 1, pp: 321-329.

WEBB, M. (1971), "Some methods of producing approximate solutions to traveling salesman problem with hundreds or thousands of cities", *OP. RES. QUART.* 22(1), pp: 49-66.

WEBB, M. (1972), "Relative performance of some sequential methods of planning multiple delivery journeys", *OP. RES. QUART.* 23(3), pp: 361-372.

WEBB, M. (1973), "The savings methods for vehicle scheduling: a reply", *OP. RES. QUART.* 24(2), pp: 307.

WILLIAMS, W. and FOWLER, O.S. (1980), "Minimum cost fleet sizing for a university motor pool", *INTERFACES* 10(3), pp: 21-28.

WIORKOWSKI, J. and McELVAIN, K. (1975), "A rapid heuristic algorithm for the approximate solution of the traveling salesman problem", *TRANSPORTATION RESEARCH* 9, pp: 181-185.

WOLTERS, J. (1979), "Minimizing the number of aircraft for a transportation network", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 3(5), pp: 394-402.

WONG, K.F. and BEASLEY, J.E. (1984), "Vehicle routing using fixed delivery areas", *OMEGA* 12(6), pp: 591-600.

WREN, A. and HOLLIDAY, A. (1972), "Computational scheduling of vehicles from one or more depots to a number of delivery points", *OP. RES. QUART.* 23, pp: 333-344.

YANO, C.A. e outros (1987), "Vehicle routing at quality stories", *INTERFACES* 17(2), pp: 52-63.

YELLOW, P. (1970), "A computational modification to the savings method of vehicle scheduling", *OP. RES. QUART.* 21, pp: 281-283

ZIERER, T.K.; MITCHELL, W.A and WHITE, T.R. (1976),
"Practical applications of linear programming to SHELL'S
distribution problems", *INTERFACES* 6(4), pp: 13-26.