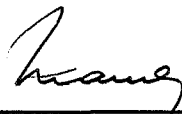


O Problema de Planejamento e Percurso de Veículos na Coleta do Lixo Urbano Domiciliar

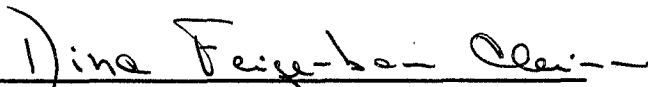
Marcos José Negreiros Gomes

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 NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

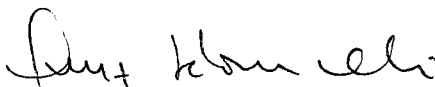
Aprovada por:



Prof. Nelson Maculan Filho, D.Sc.
(presidente)



Profa. Dina Feigenbaum Cleiman, D.Sc.



Prof. Luis Satoru Ochi, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
OUTUBRO DE 1990

GOMES, MARCOS JOSÉ NEGREIROS

O Problema de Planejamento e Percurso de Veículos na Coleta do Lixo Urbano Domiciliar [Rio de Janeiro] 1990
IV, 142 pgs., 29.7 cm, (COPPE/UFRJ, M. Sc., ENGENHARIA DE SISTEMAS E COMPUTAÇÃO, 1990)
TESE – Universidade Federal do Rio de Janeiro, COPPE
1 – Otimização em Grafos 2 – Planejamento 3 – Roteamento de Veículos
I. COPPE/UFRJ II. Título(Série).

*A minha Nenzinha, Ana Cristina, com amor,
e ao Senhor, meu fiel e sábio amigo ...*

Agradecimentos

Muitas pessoas participaram deste trabalho direta ou indiretamente. Boa parte delas, contribuíram com sua amizade e carinho, mantendo razoavelmente o meu equilíbrio para atingir este texto final. É muito duro, principalmente para mim, deixar minha casa e enfrentar solitariamente esta terra, que, apesar de tudo, trouxe à minha vida pessoal e profissional grandes e enriquecedores momentos.

Aos meus pais, Miguel Arcanjo e Maria Rosa, por todo o apoio, amor, paciência e dedicação.

Aos meus irmãos, Miguel Arcanjo Negreiros, responsável pela programação do banco de dados geográfico, e Prof. Francisco Negreiros, por todas as idéias e o apoio durante este trabalho.

À minha avó Conceição, ao irmão Antônio José, às minhas irmãs Aparecida (e Edson Filho) e Zêra, à cunhada Jussara Gomes, a todos os meus sobrinhos e parentes, aos amigos Max Padilha, Almir Torquato, Ismael Régis, Bryon, Gerardo Filho, Fernando Rivas, Antinous Padilha, Adriano Picanço, Eudes Júnior, Ivo Machado, Célia, Maria das Graças, Clevi, Cristina, Cândida, Zuí, Ana Cláudia, Path, Ivana Pontes e aos colegas do laboratório de sistemas, Edu, Adilson, Alberto, Zé, Gamal e demais professores e colegas da COPPE/Sistemas, meu maior agradecimento por sua participação com amizade e carinho.

À Construtora Marquise Ltda., na pessoa de seu Dir. Pres. José Carlos Ponte e à EMLURB, na pessoa de seu Dir. Pres. Francisco Adail Fontenele, por todo o apoio recebido no conhecimento geral do problema em Fortaleza.

Aos professores Antônio Clécio e Renato Craveiro, pela assistência e orientação em Fortaleza.

Aos meus companheiros do trabalho em campo em Fortaleza, Valber Rios, Eng. Paulo Bruno, Eng. João Alencar, Maria, Enedina, Ricardo Kildare, Gerson, Hértenes, Rodrigo e Romana, pela disposição e atenção no trabalho.

À SISCO, filial de Fortaleza, pela gentil e calorosa acolhida nos momentos mais difíceis de execução do algoritmo MARCOLIX, em seu equipamento SIX-386.

Aos professores Jano Moreira, Dina Cleiman, Arlindo Gomes e aos colegas Guilherme Travassos, Cláudio Trotta, e Flávio Foster pelas sugestões e grandes idéias ao trabalho.

Às secretárias da COPPE/Sistemas, Denise, Cláudia, Ana Paula, Da. Helena, por me aturarem tanto.

Aos grandes incentivadores deste, Renato Bueno, Luis Ochi, Murilo

Castellano, Celso Soares, Gilbert Laporte e John Beasley. Pessoas que se dedicaram e se dedicam ao estudo do problema de Roteamento com grande entusiasmo.

Aos motoristas e coletores (garis), da coleta do lixo de Fortaleza, que se mostraram sempre tão prestativos e gentis comigo, além do seu esforço diário na manutenção da saúde de minha adorada cidade.

À CAPES e CNPQ, pelo suporte financeiro.

E por fim, ao meu orientador Prof. Nelson Maculan Filho, por todas as idéias, paciência e respeito, além do meu eterno agradecimento, minha maior admiração e amizade.

Resumo da Tese apresentada à COPPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.)

O Problema de Planejamento e Percurso de Veículos na Coleta do Lixo Urbano Domiciliar

Marcos José Negreiros Gomes
Outubro de 1990

Orientador: Nelson Maculan Filho
Programa: Engenharia de Sistemas e Computação

Esta tese mostra uma revisão em Problemas de Percurso em Arcos, de modo que seja aplicado à coleta do lixo domiciliar. Discutimos os casos não capacitados, em redes não direcionadas, direcionadas e mistas, e capacitados, onde propomos algumas idéias de melhoria no algoritmo de ULOSOY (1985).

O Problema de Planejamento da Frota diz respeito à gerência e atribuição de veículos ao nosso caso estudo em Fortaleza, onde a frota já é conhecida e desejamos minimizar sua depreciação, de acordo com as restrições de tempo de operação, e também o número de viagens numa área específica, que chamamos de sub-zonas.

Estudamos o problema de Fortaleza na coleta do lixo, e modelamos o problema de percurso dos caminhões coletores como um Problema de Percurso Capacitado em Arcos (PPCA) sobre restrições de tempo de operação, numa rede mista, com demanda em arcos e nós.

Finalmente apresentamos um algoritmo heurístico original (MARCOLIX) de ordem $O(n^2 \log 2k)$ (n - número de nós da rede e k - número de elementos a servir), e mostramos a sua solução em três exemplos, onde um deles simula um pior caso numa aplicação real.

No final, mostramos alguns problemas resolvidos com o uso do algoritmo MARCOLIX, em uma rede fictícia de 32 nós e 62 arcos e em uma rede real com 2900 nós e 5900 arcos em um microcomputador PC-AT/286.

Abstract of Thesis presented to COPPE as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.)

The Problem of Planning and Routing Vehicles in Urban Refuse Collection

Marcos José Negreiros Gomes

October, 1990

Thesis Supervisor: Nelson Maculan Filho

Department: Programa de Engenharia de Sistemas e Computação

This thesis presents an overview on Routing Problems in Arcs, to be applied to the household refuse collection. We discuss the uncapacited, in undirected, directed and mixed graphs, and capacited problems, where we propose ideas to improve ULOSOY's (1985) algorithm.

The Problem of Planning Fleet concerns of a management and assignment of vehicles to our spetial case study in Fortaleza, where the fleet is already known and we want to minimize its depreciation, according time operation constraints, and also the number of travels in a specific area, that we called sub-zones.

We study the Fortaleza's problem on garbage collection, and model the problem of routing the collection vehicles as a Capacited Arc Routing Problem (CARP), under total time operation constraint, in a mixed graph, with demands in arcs and nodes.

Finally, we present an original, $O(n^2 \log 2k)$ (n - number of nodes and k - number of elements to be served), heuristic algorithm (MARCOLIX), and we show its resolution in three exemples, one of them simulating the worst case in real life application.

At the end, we show some problems we solve using the MARCOLIX algorithm, in a fictious network of 32 nodes and 63 arcs, and in a real network of 2900 nodes and 5900 arcs, in a microcomputer PC-AT/286.

Índice

I	Introdução	1
I.1	O que é o Problema de Percorso de Veículos?	2
I.2	O que é Problema de Planejamento de Veículos?	2
I.3	Como desenvolver um Sistema de Roteamento?	2
I.4	Resumo do Conteúdo da Tese	6
II	O Problema de Percorso de Veículos em Arcos	7
II.1	Histórico das Abordagens em Percursos	7
II.2	Notação e Simbologia de Grafos	13
II.3	O Início dos Problemas com Grafos	14
II.4	O Problema de <i>Matching</i>	19
II.4.1	Formulação do Problema de <i>Matching</i>	19
II.5	O Problema do Carteiro Chinês (PCC)	24
II.5.1	O Problema Não Direcionado	25
II.5.2	O Problema Direcionado (PCCd)	31
II.5.3	O Problema Misto (PCCm)	36
II.6	O Problema de Percorso Capacitado em Arcos (PPCA)	46
II.6.1	O Trabalho de Ulusoy	51
II.7	Conclusões	61
III	O Problema de Planejamento da Frota (PPF)	62
III.1	Introdução	62
III.2	Descrição do (PPF)	63

III.3 Aplicação do Planejamento e Alocação de Frotas	68
III.4 Observações e Conclusão	77
IV O Problema de Percurso na Coleta do Lixo Domiciliar	78
IV.1 Introdução	78
IV.2 Determinando o Problema da Coleta do Lixo Urbano	79
IV.3 O Algoritmo MARCOLIX	86
IV.3.1 Observações sobre o algoritmo MARCOLIX	95
IV.3.2 Exemplos de Roteamento Utilizando o Algoritmo MARCOLIX	96
IV.4 Conclusão	105
V Conclusão	106
Referências Bibliográficas	108
Apêndice 1: Complexidade Computacional dos Algoritmos de Roteamento	114
Apêndice 2: O Algoritmo Primal-Dual OUT-OF-KILTER	116
Apêndice 3: Telas do S i s R o t	120
Apêndice 4: Desempenho Computacional do Algoritmo MARCOLIX135	

Capítulo I

Introdução

Há muito tempo, no Brasil, vem-se realizando a administração da coleta do lixo urbano através da experiência e sensibilidade dos gerentes que há anos trabalham na atividade. Apesar da experiência, eles pouco podem fazer quando precisam definir uma frota econômica e hábil para o serviço de coleta, por exemplo, pois a topologia da cidade e a formação de suas ruas e avenidas não dão uma situação verídica do melhor modo de percorrê-las com uma frota adequada, numa análise puramente circunstancial. Apesar disso, bons *chutes* são dados e vai-se levando o serviço, comandando como se pode a tarefa difícil e diária de planejar e manter a limpeza urbana.

Este trabalho começou com a necessidade de se resolver o problema da administração da coleta de lixo domiciliar na cidade de Fortaleza (Ceará). Muitos eram os problemas de Fortaleza com a limpeza urbana, o mais difícil deles era e é, na verdade, definir um roteiro exequível em tempo hábil pelos motoristas responsáveis do serviço. Além disso, também era preciso definir como poderia ser realizada a operação da coleta, minimizando a depreciação do equipamento utilizado, de maneira tal que fosse garantida a constância da atividade nos períodos especificados pelas regras contratuais.

Fortaleza, depois de passar por *maus tempos* em termos de limpeza urbana, hoje é uma das capitais mais limpas do país. Gasta-se por volta de US\$ 7.318,86 diariamente apenas com a coleta domiciliar. As despesas com os veículos representam 84.83% do total, sendo 51.90% do montante diário gasto apenas com rodagem¹. Apesar dos custos com pessoal serem pouco significativos em relação ao custo de manter os veículos, todo o serviço depende dos horários da tripulação, pois os gastos com horas extras são uma variável adicional importante na composição final dos custos da atividade.

Neste capítulo, daremos uma idéia do que será discutido neste trabalho de modo a resolver o problema da coleta do lixo específico de cidades como

¹Estes dados são extra oficiais. Foram cedidos pela EMLURB (Empresa Municipal de Limpeza e Urbanização) relativos ao mês de Agosto de 1990, através do seu Diretor Presidente Francisco Adail Fontenele, e está relacionado ao valor convertido em dólar americano da cotação do cruzeiro naquele mês.

Fortaleza, e também procuraremos determinar metodologias de identificação e guia de resolução de um problema específico de roteamento.

I.1 O que é o Problema de Percurso de Veículos?

O problema de Percurso (Roteamento) de Veículos pode ser definido da seguinte forma: Dado um conjunto \mathcal{P} de k veículos, com capacidade de carregamento de Q_i ($i = 1, \dots, k$) unidades de produto respectivamente, sediados em um depósito x_0 , e um conjunto \mathcal{X} de n clientes, cada um com a demanda de q_i unidades de produto, e distantes de d_{ij} unidades entre si. Definir o conjunto de rotas $\mathcal{R} = \{R_1, R_2, \dots, R_m\}$ que estão associadas a um subconjunto dos veículos, de modo que estas possam, partindo do depósito e retornando a ele, suprir as demandas dos clientes com custo mínimo – DANTZIG & RAMSER (1959), CLARK & WRITE (1964), CHRISTOFIDES (1976), LENSTRA & RINNOOY KAN (1981).

I.2 O que é Problema de Planejamento de Veículos?

O problema de Planejamento de Veículos é o problema de se definir a sequência de rotas a realizar, de modo que sejam respeitadas as restrições de tempo de serviço e paradas obrigatórias impostas pelas contingências do trabalho – MAGNANTI (1981).

I.3 Como desenvolver um Sistema de Roteamento?

As economias que são feitas em virtude da implantação de sistemas de roteamento, é algo de se impressionar – GOLDEN ET AL (1983), TURNER ET AL (1975), BODIN ET AL (1989), etc. Mas o sucesso desta implementação sujeita-se a uma série de fatores que concernem à apreciação tanto do investidor, quanto do especialista que o desenvolverá. A iteração entre ambos, no processo, deve ser bastante harmoniosa, e a objetividade na identificação do trabalho a ser executado é uma premissa que não deve ser esquecida seja qual for a situação contingente – GOLDEN & ASSAD (1988).

Algumas perguntas básicas para o início de trabalho, foram propostas por GOLDEN & ASSAD (1988), chamadas de Macro-Questões. São elas:

1. Tamanho da frota usada pela firma;
2. Número de motoristas;

3. Número de rotas percorridas diariamente e média de paradas/dia;
4. Operações entre-cidades e dentro das cidades;
5. Custo total anual das atividades de distribuição;
6. Custo de pessoal e veículo como percentagem do custo total;
7. Demanda futura e previsão da área a servir;
8. Disponibilidade atual de computador e suporte para roteamento;
9. Integração do roteamento com outras atividades;
10. Possibilidade do sistema de roteamento trabalhar sozinho ou totalmente integrado.

As informações obtidas a partir das Macro-Questões acima, sempre permitem um acesso inicial às dimensões dos recursos que a empresa aloca às funções de distribuição. As macro características das atividades de distribuição podem ser tomadas como um roteiro grosseiro para uma estimativa dos benefícios potenciais de um sistema mais elaborado de despacho e roteamento. Tal modelo poderia requerer uma classificação dos problemas de roteamento, de acordo com a natureza das atividades de distribuição envolvidas.

Da classificação, as atividades realizadas permitem uma identificação para um modelo pertinente. As possíveis atividades são:

- Atividades de apenas recolhimento ou apenas entrega;
- Procedimento com possibilidade de reaquisição;
- Atividade combinada de recolhimento e entrega.

A partir de então, dá-se uma conotação mais prática às questões para identificação do modelo. São assim discriminadas:

Natureza da Demanda

- *Apenas recolhimento ou apenas entrega;*
- *Recolhimento/entrega com opção de reposição;*
- *Simple ou múltiplos artigos;*
- *Deve servir toda a demanda;*
- *Opção de carregamento comum (no caso de múltiplos artigos);*
- *Prioridade a clientes.*

Informações Sobre a Demanda

- *Toda a demanda é conhecida antecipadamente?*
- *Muitas demandas são repetidas?*
- *A frequência é fixa para visitas?*
- *A demanda é incerta?*
- *O fluxo de demanda é em tempo real?*

Frota de Veículos

- *A frota é homogênea ou heterogênea?*
- *Há restrições de peso e capacidade?*
- *Há problema com Compartimentos?*
- *Há restrições de carregamento/equipamento?*
- *Tipo de veículo/dependências locais;*
- *Tipo de veículo/compatibilidade de artigos;*
- *O tamanho da frota é Fixo ou Variável?*
- *A frota está baseada em múltiplos depósitos ou em um único.*

Necessidades de Pessoal

- *Estrutura de pagamento*
 - . *Dimensão do dia de trabalho;*
 - .. *Mínimo e máximo nas horas de trabalho;*
 - ... *Opção de horas extras;*
- *Número de motoristas fixo ou variado;*
- *Locais e horas de início de movimento;*
- *Almoço ou outras paradas;*
- *Viagens permitidas em múltiplos dias.*

Necessidades de Planejamento

- *Designação de clientes para os dias da semana;*
- *Janelas de tempo para recolhimento/entrega (Fixo/Variável);*
- *Horas de abertura e fechamento;*

- *Horários de carga/descarga.*

Dados Necessários

- *Banco de Dados Geográfico;*
- *Localização e endereço de clientes;*
- *Horas de viagens;*
- *Informações de localização de veículos;*
- *Crédito de clientes e informações de contas.*

Assim, com o que temos acima caracterizado, poderemos definir como resolver o problema de roteamento, usando a técnica adequada e produzindo um sistema compatível com as operações do investidor.

Devemos, também, levar em conta as restrições do nosso problema. Enumeramos, a seguir, algumas das quais poderiam advir após a caracterização, com relação às rotas. São elas:

1. Duração da rota;
2. Locais de início para a rota/motorista;
3. Hora de início das rotas;
4. Viagens à noite e regras trabalhistas associadas;
5. Paradas intermediárias;
6. Restrições de peso e volume no carregamento;
7. Reservas de carregamento e restrições;
8. Território dos motoristas e regiões de entrega;
9. Limites naturais ou legais das regiões;
10. Agrupamento de clientes;
11. Regras de divisão de entregas (Se houver);
12. Importância das rotas balanceadas;
13. Múltiplas rotas por veículo;
14. Uso de rotas fixas e paradas fixas nas rotas;
15. Modificação nas rotas baseadas em novas ordens que entram;
16. Regras especiais para segmentos de vias;
17. Ruas de mão dupla;
18. Evitando retornos em U, na mesma rua, ou outras regras de resguardo.

A partir de então o especialista poderá desenvolver o seu trabalho, contando com o apoio do investidor para eventuais operações que estão fora das enumeradas no problema. No nosso caso, usaremos os mesmos procedimentos que vêm sendo praticados pela literatura na execução destes sistemas, supra citados, como guia para o nosso trabalho.

I.4 Resumo do Conteúdo da Tese

Do ponto de vista teórico, procuramos formular nossos estudos totalmente voltados à nossa aplicação, Coleta de Lixo Domiciliar, que, por sua abrangência, nos levará ao uso frequente de técnicas de Teoria dos Grafos, Programação Inteira (Combinatória) e Percurso de Veículos.

No Capítulo II, procuramos desenvolver nossos estudos buscando uma identificação com o problema prático que iremos atacar. Estudaremos, assim o Problema de Percursos em Grafos não Capacitados e Capacitados.

Nos Capítulos III e IV, abordamos os conceitos de planejamento e roteamento na coleta domiciliar, respectivamente. No caso do Capítulo III, procuramos modelar e resolver o Problema de Planejamento de saída da frota para coleta, e distribuição racional dos veículos na execução do serviço. Já no Capítulo IV, discutimos o problema de percurso na coleta domiciliar e propomos um algoritmo heurístico que mostra alto desempenho, para solução do problema, nos casos testes citados.

Nos Apêndices 1 e 2, discutimos a complexidade dos algoritmos mais comuns de roteamento, e um pouco sobre o algoritmo Out-of-Kilter usado na resolução do problema de percurso direcionado. Já nos Apêndices 3 e 4, mostramos alguns relatórios e características de um sistema computacional desenvolvido para auxiliar na administração da coleta do lixo, e analisamos alguns testes realizados com o uso do algoritmo MARCOLIX proposto.

Capítulo II

O Problema de Percurso de Veículos em Arcos

II.1 Histórico das Abordagens em Percursos

O Problema de Percurso de Veículos e Planejamento, vem merecendo bastante atenção nos últimos trinta anos pelos cientistas da área de otimização em grafos. O primeiro trabalho publicado foi feito por DANTZIG & RAMSER (1959), no qual formulam o problema visando especificamente a distribuição de gasolina em postos. Neste trabalho eles propuseram uma heurística sub-ótima estudada para um caso teste de 12 (doze) postos, partindo os veículos de um depósito único (garagem).

A formulação do problema de *Despacho de Caminhões*, como foi chamado inicialmente, foi assim definida naquele artigo:

1. Dado um conjunto de n Estações pontos P_i ($i = 1, 2, \dots, n$) as quais é preciso fazer entregas a partir do ponto P_0 , chamado de *Ponto Terminal* ou *Depósito*;
2. Dada uma matriz de distância $[d_{ij}]$, a qual especifica a distância $d_{ij} = d_{ji}$ entre todos os pares de pontos ($i, j = 1, 2, \dots, n$);
3. Um vetor de entrega $[q_i]$ é definido, o qual especifica a medida q_i que será entregue a todo ponto P_i ($i = 2, \dots, n$);
4. A capacidade (carga nominal) do caminhão é Q , onde $Q > \max_i(q_i)$;
5. Seja x_{ij} uma variável que define se a ligação entre os pares $P_i P_j$ será realizada ou não, sendo $x_{ij} = x_{ji} = 1$ caso seja necessário emparelhar $P_i P_j$, e $x_{ij} = x_{ji} = 0$ caso contrário. Onde ($i, j = 0, 1, 2, \dots, n$), e $x_{ii} = 0 \forall i$.
6. O problema é então encontrar aqueles valores para x_{ij} tornando a distância total,

$$D = \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}$$

um mínimo, dentro das condições especificadas de 3. a 5.

Em linguagem de programação matemática, a formulação acima pode ser assim expressa, GOLDEN ET AL (1977):

Sejam:

C_{ij} - Distância entre i e j , ($i, j = 0, \dots, n$);

q_i - Demanda do cliente i , ($i = 1, \dots, n$);

Q^k - Capacidade do veículo k ;

$x_{ij}^k = \begin{cases} 1, & \text{se o veículo } k \text{ vai de } i \text{ para } j \text{ diretamente} \\ 0, & \text{caso contrário.} \end{cases}$

O problema formula-se como:

$$(PPV) \quad \text{Min} \quad \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n C_{ij} x_{ij}^k \quad (\text{II.1})$$

sujeito a

$$\sum_{i=0}^n \sum_{k=1}^m x_{ij}^k = 1 \quad (j = 1, \dots, n) \quad (\text{II.2})$$

$$\sum_{j=0}^n \sum_{k=1}^m x_{ij}^k = 1 \quad (i = 1, \dots, n) \quad (\text{II.3})$$

$$\sum_{i=0}^n (x_{ij}^k - x_{ji}^k) = 0 \quad (j = 0, \dots, n) \quad (k = 1, \dots, m) \quad (\text{II.4})$$

$$\sum_{i=1}^n q_i \sum_{j=0}^n x_{ij}^k \leq Q^k \quad (k = 1, \dots, m) \quad (\text{II.5})$$

$$x_{ij}^k \in \{0, 1\} \quad (i, j = 0, \dots, n), \quad (k = 1, \dots, m) \quad (\text{II.6})$$

Cinco anos mais tarde, a comunidade científica conheceu o trabalho de CLARKE & WRITE (1964), que discutia o problema abordado por DANTZIG & RAMSER (1959), onde a heurística buscava mais exatamente agrupar demandas, visando atender a capacidade do caminhão, ao invés de minimizar a distância.

CLARKE & WRITE (1964), propuseram então uma heurística baseada em *economias de ligação*, entre demandas, visando equilibrar o processo de serviço, considerando uma frota cujos veículos pudessem ter cargas diferenciadas. Tal heurística é até hoje consagrada, pois no que se trata de otimização do número de veículos, para problemas com grandes quantidades de pontos de demanda, ela é quase sempre mais eficiente, além da facilidade de implantação de novas restrições ao problema e vantagem computacional (rapidez, facilidade de programação, etc.) – BUENO FILHO (1989).

Muitos trabalhos, a partir de então vieram, todos fazendo modificações na heurística base, como exemplos temos: GASKELL (1967), que examina as versões sequenciais e paralela do algoritmo de economias e introduz alternativas para a medida das *economias*; YELLOW (1970), introduz uma melhoria computacional no algoritmo de economias o qual reduz tanto a memória quanto o tempo computacional, e foi resolvido aí um problema com 1000 pontos de demanda em cinco minutos, num IBM 360/50; MOLE & JAMESON (1976), o critério de economias é generalizado e novos parâmetros são introduzidos o que provocam uma mudança no formato das rotas, dependendo do seus valores (algoritmo sequencial); GOLDEN ET AL (1977), onde se discute estruturas de dados eficientes para a implementação do algoritmo das economias, o que também estende o método para o caso de vários depósitos numa implementação com extrema rapidez para um número considerável de pontos de demanda, no caso, foi resolvido um problema com mais de 600 pontos de demanda e dois depósitos, em 50 seg, num IBM 370/168.

Outros procuraram resolver o problema utilizando outra forma de pensamento, como por exemplo: GILLET & MILLER (1974), que descreve uma heurística de duas fases para PCV (Problema do Caixeiro Viajante), que se tornou conhecida como algoritmo de *Varredura* ou *Sweep* – Pontos de demanda de determinado agrupamento na ordem a qual são *Varridos* pela rotação imaginária de um raio que parte do depósito, de modo que quando a capacidade do agrupamento alcança a capacidade do veículo, um novo agrupamento é iniciado e a *Varredura* continua. Na segunda fase os pontos de demanda são sequenciados, de modo a produzir uma rota; CHRISTOFIDES & EILON (1969), converte o PPV em um PCV e o resolve exatamente, para 30 pontos de demanda, usando a técnica *Branch-and-Bound*, estendem o método r -ótimo (3-ótimo) de LIN (1973) para o PPV, e dá resultados computacionais para este método em problemas acima de 100 pontos de demanda. Os problemas testes que foram aqui gerados servem como parâmetros para os testes realizados com outros algoritmos da literatura; FISHER & JAIKUMAR (1981), descrevem um método em duas fases, que é, essencialmente a primeira iteração do Método de Decomposição de Benders, dos mesmos autores. Na primeira fase de atribuição dos pontos de demanda às rotas, m pontos de demanda são escolhidos como sementes para iniciar a construção dos m grupos de rotas. Os custos de inserção são computados para todos os outros pontos de modo a indicar o custo de atribuir um ponto de demanda a uma semente (e portanto a um grupo correspondente).

Usando estes custos uma generalização do problema de atribuição é resolvido para produzir a rota dos grupos. A segunda fase resolve o PCV para cada grupo de rotas, de modo a produzir a solução do PPV. Bons resultados computacionais são obtidos para problemas com mais de 200 (duzentos) pontos de demanda.

A característica do problema de roteamento, que foi inicialmente abordado pela literatura, é exclusivamente em vértices de um grafo $G = (V, E)$, completo e cuja matriz de distância entre dois pontos é uma função euclidiana, ou seja:

$$d_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2, \quad \forall (i, j = 0, 1, \dots, n)$$

onde, (x_i, y_i) são coordenadas de P_i no espaço euclidiano.

Esta versão resolve a parte fundamental do problema, que é sequenciar os pontos de demanda que devem ser visitados por veículo determinado. O estudo de complexidade deste problema, LENSTRA & RINNOOY KAN (1981), mostra que é do tipo \mathcal{NP} -Hard, ou seja, não há algoritmo de tempo polinomial que o solucione exatamente, para o caso geral. Sendo mais claro, se tentarmos transformá-lo em um novo problema similar, cairemos sempre em um problema também \mathcal{NP} -Hard. No nosso caso, como notamos, sempre teremos que resolver um PCV para satisfazer as restrições do caixeiro viajante, inclusas na formulação do modelo, e este é, originalmente \mathcal{NP} -Hard (Ver Apêndice 1).

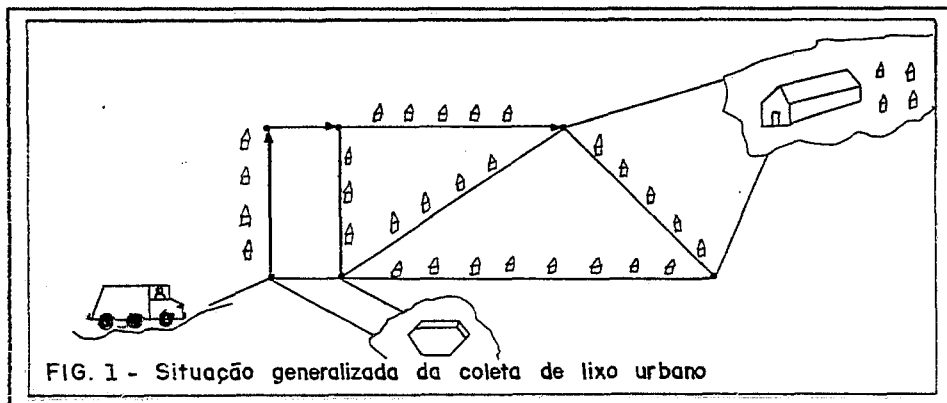
Paralelamente, MEI-KO KWAN (1962) introduziu o problema do carteiro, o qual teria que passar por todos os quarteirões de uma determinada zona da cidade, percorrendo a menor distância possível, antes de retornar à agência local do correio. Este problema foi denominado de Problema do Carteiro Chinês (PCC), e foi resolvido por EDMONDS & JOHNSON (1973), através de um problema similar *1-Matching* de menor comprimento.

O trabalho de EDMONDS & JOHNSON (1973) estaria completo caso não esbarrasse em um problema onde o carteiro tivesse que obedecer o sentido de tráfego, numa região de ruas que pudessem ser de mão dupla e mão única, ou seja, num grafo misto com matriz de custos sem simetria. Este problema foi estudado por MINIEKA (1979) e FREDERICKSON (1979), cada um propondo uma heurística diferente, sendo a de MINIEKA (1979) utilizando uma modificação do grafo e propondo uma solução utilizando o algoritmo de *Fluxo com Ganhos* para definir a sequência de travessia do grafo e aqueles que deveriam ser multiplicados de modo a permitir a transformação do grafo em grafo euleriano; e FREDERICKSON (1979) utiliza-se de um trabalho mais detalhado, onde explica duas heurísticas. A primeira, *Mixed 1* procura abordar o problema resolvendo como passo inicial o *1-Matching* desconsiderando a direção dos arcos, em seguida realiza a definição de travessia de modo a fazer conservação de fluxo, minimizando o fluxo dos arcos direcionados, ajusta a direção dos arcos não direcionados de modo a manter a paridade nos nós do grafo, por fim, ajusta um circuito euleriano. A segunda, *Mixed 2*, inicia fazendo o processo de conservação de fluxo, faz o *1-Matching* e por fim ajusta o circuito euleriano.

O trabalho de FREDERICKSON (1979) pode ser considerado como de grande valor científico, pois ambos os algoritmos estão relacionados com a solução ótima em torno de $5/3$, quando o pior caso sugere que a relação entre Solução Heurística/Solução Ótima ≤ 2 . Mais interessante ainda, é saber que tais algoritmos são polinomiais.

O problema de roteamento deixou de ser, portanto, encarado como pontual. Em 1974, Orloff, após a defesa de sua tese de doutorado, publicou um artigo completo sobre o Problema Fundamental de Roteamento, e determinou variantes no problema de roteamento, as quais podemos citar: O Problema do Carteiro Rural (PCR), onde um carteiro deve visitar todas as quadras de uma determinada sub-região da área total, com percurso minimizado, e o Problema Geral de Roteamento (PGR), onde tanto se deve passar por todos os pontos de demanda, como todos os arcos que possuem demanda. Os dois problemas estão na classe dos problemas \mathcal{NP} -Hard, LENSTRA & RINNOOY KAN (1981).

Um exemplo clássico do PGR pode ser assim descrito: imagine que um caminhão de lixo precisa passar por todos as ruas de uma determinada região de sua cidade, e, também deve recolher o lixo de pontos desta região, como determinados pontos comerciais ou *Shopping Centers*. Este problema pode ser assim visualizado, em um grafo:



Vários trabalhos nesta área foram desenvolvidos BODIN & BELTRAMI (1974), uma aplicação com resolução do PCC direcionado, com aplicação à coleta do lixo municipal na cidade de New York; TURNER & HOUGLAND (1975) com uma aplicação modificada do algoritmo de CLARKE & WRITE (1964), adaptada para a coleta de lixo em regiões rurais, BOAVENTURA ET AL (1982), uma implementação interativa, utilizando o PCC direcionado na geração das rotas dos veículos, dentre outros. Tais trabalhos obtiveram bons resultados práticos, mas estão longe de abordar o problema de roteamento em arcos e nós.

Como vemos, a característica do problema da coleta de lixo pode ser analisada de vários modos:

- Quando a região é toda um grafo não - orientado e o caminhão pode realizar a coleta, como um carteiro que precisa passar por todos os quarteirões, pois estes necessitam serviço (PCC);

- Quando a região é toda um grafo orientado e o caminhão pode realizar a coleta como um carteiro (PCCd)
- Quando a região é toda um grafo misto e o caminhão pode realizar a coleta como um carteiro (PCCm) e
- Quando um ou mais caminhões têm que fazer a coleta, até que o lixo seja retirado completamente, passando por todas as ruas que necessitam de serviço (PPCA).

Discutiremos, a seguir, as abordagens acima, procurando exemplificar cada uma delas conforme os algoritmos estudados, aplicando diretamente ao problema da coleta do lixo urbano.

II.2 Notação e Simbologia de Grafos

Antes de iniciarmos quaisquer discursões acerca dos enfoques abordados anteriormente, procuraremos definir conceitos fundamentais da teoria de grafos, tal que seja facilitada a compreensão deste texto.

Chamaremos de grafo indireto ou não-direcionado de $G = (V, E)$, onde $V \neq \emptyset$ é o conjunto de vértices ou nós e $E \neq \emptyset$ é o conjunto de elos. Elos paralelos serão todos aqueles que possuírem seus vértices iniciais e finais iguais, e denotaremos por ij para um nó $i \in V$, denotamos o conjunto de elos incidentes a i por $g(i)$, e sua cardinalidade por $d_G(i)$, que representa o grau do vértice i no grafo G . Um vértice de grau par é aquele que possui cardinalidade $d_G(i)$ par, e um vértice de grau ímpar, possui cardinalidade $d_G(i)$ ímpar.

Digrafos ou grafos direcionados, $D = (V, A)$, onde $V \neq \emptyset$ é o conjunto de vértices ou nós e $A \neq \emptyset$ o conjunto de arcos. Chamaremos um arco $a \in A$ incidente de i e incidente para j por $a = (i, j)$. Para um nó $i \in V$, denotaremos o conjunto de arcos incidentes de i (incidentes para i) por $g^+(i)$ ($g^-(i)$), e sua cardinalidade denotada por $d^+(i)$ ($d^-(i)$), o número de arcos que possuem como *Rabo* (*Cabeça*) o vértice i .

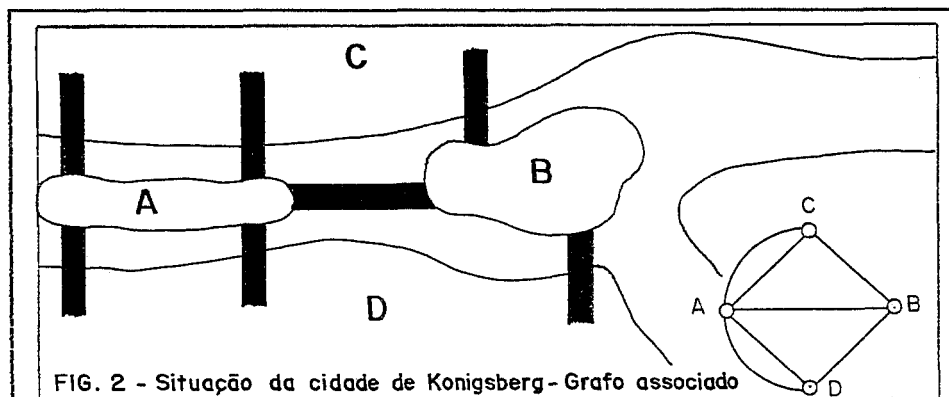
Grafos mistos, $G = (V, E, A)$, onde $V \neq \emptyset$ é o conjunto de vértices ou nós, E o conjunto de elos e A o conjunto de arcos. Num grafo misto, o grau de um vértice i , será a soma do número de elos incidentes com i , o número de arcos incidentes de i , e o número de arcos incidentes para i .

Seja $G = (V, E)$ um grafo e $u, v \in V$. Uma sequência $\omega = \langle i_0 i_1, i_1 i_2, \dots, i_{k-1} i_k \rangle$ de elos de E com $i_0 = u$ e $i_k = v$ é chamado um *Passeio* entre u e v . ω é *Aberto* se $u \neq v$ e *Fechado*, caso contrário. ω é um *Caminho* se $i_s \neq i_t$, para $s \neq t$, $s \geq 0$, $t \leq k$. Se ω é um *Caminho* e $i_k i_0 \in E$, então a sequência $\langle i_0 i_1, i_1 i_2, \dots, i_{k-1} i_k, i_k i_0 \rangle$ é um *Ciclo*. Se um *Passeio* existe entre quaisquer pares de vértices de V , então G é dito ser *Conexo*. Um passeio de G contendo cada elo de E exatamente uma vez é chamado de *Passeio de Euler*. E um grafo que possui um *Passeio de Euler*, é um *Grafo Euleriano*.

Seja $D = (V, A)$ um digrafo, e u e v dois nós em V . Uma sequência $\omega = \langle (i_0, i_1), (i_1, i_2), \dots, (i_{k-1}, i_k) \rangle$ de arcos em A com $i_0 = u$ e $i_k = v$ é um *Dipasseio* ou *Passeio Direcionado* de u para v . ω é *Aberto* se $u \neq v$ e *Fechado* caso contrário; e ω é um *Dicaminho* ou *Caminho Direcionado* se $i_s \neq i_t$ para $s \neq t$, $s \geq 0, t \leq k$. Se ω é um *Dicaminho* e $(i_k, i_0) \in A$, então a sequência $\langle (i_0, i_1), (i_1, i_2), \dots, (i_{k-1}, i_k) \rangle$ é um *Diciclo* ou *Ciclo Direcionado*. Com qualquer digrafo $D = (V, A)$, podemos associar o grafo $G(V, E)$, onde $E = \{ij \mid (i, j) \in A\}$. Dizemos que D é *Conexo* se G é *Conexo*. Um *Passeio Direcionado Fechado* em D contendo cada arco de A exatamente uma vez é chamado um *Dipasseio Euleriano* e um digrafo D que possui um *Dipasseio Euleriano* é um *Digrafo Euleriano*.

II.3 O Início dos Problemas com Grafos

Em 1736, o matemático suíço Leonhard Euler (1707 - 1783) publicou, pela academia de ciência de St. Petersburg, o primeiro estudo de grafos que se conhece na história, baseado no problema chamado de *Problema das pontes de Königsberg*. A cidade de Königsberg (hoje Kaliningrad), ao leste da Prússia, foi construída na junção de dois rios e duas ilhas formadas por eles (Ver fig.2). Ao todo, haviam sete pontes ligando as ilhas e o restante da cidade. O problema era, então: *Como poderia um morador de Königsberg iniciar de sua casa um passeio por todas as pontes, atravessando-as somente uma vez e retornando para casa?*



Para solução ao Problema das Sete Pontes de Königsberg, foi enunciado o seguinte teorema:

TEOREMA - CHRISTOFIDES (1975): *Seja um subgrafo conexo e indireto $G = (V, E)$, este só contém um circuito (Passeio) euleriano se e somente se o número de vértices de grau ímpar é nulo para o caso do circuito, ou nulo ou dois para o caso do caminho.*

Portanto, como se era de prever, o habitante de Königsberg não conseguiria fazer seu passeio, da forma como imaginava. E, aqui iniciamos a nossa preocupação em resolver problemas de percurso, que já remonta de vários séculos, como o primeiro problema proposto e discutido na literatura, na área de grafos.

Para o caso de um arco direcionado, ou digrafo, o seguinte teorema foi proposto:

TEOREMA - CHRISTOFIDES, (1975): *Um subgrafo direto $D' \subset D$, contém um Dircircuito (Dipasseio) Euleriano se e somente se os graus positivos e negativos dos vértices satisfazem as condições:*

- Para o caso de um Dircircuito: $d^+(i) = d^-(i), \forall i \in V;$
- Para o caso de um Dipasseio:

$$d^+(i) = d^-(i), \forall i \neq p \text{ ou } q;$$

$$d^+(q) = d^-(q) + 1$$

$$d^+(p) = d^-(p) - 1$$

Onde p é o vértice inicial e q é o vértice final do Percurso Euleriano.

Como poderíamos, então, resolver o problema do habitante de Königsberg, de modo que ele faça seu passeio pela cidade passando por todos as pontes e retornando para casa, repetindo o menor número de vezes a utilização de quaisquer das pontes. Resolveremos este problema através de um algoritmo que visará tornar qualquer grafo indireto que possua, para todo vértice ímpar pelo menos um nó adjacente ímpar, em grafo euleriano.

Seja então a situação em que temos um grafo $G = (V, E)$ indireto e conexo. O seguinte algoritmo considera as variáveis como:

$d(i)$ – Grau de i

$NL(ij)$ – Número de elos paralelos ij

Algoritmo EULER¹

Passo 1: Identifique todos os nós de grau ímpar. Se $\exists d(i) \mid i \in V$ ímpar, então $G \equiv G_{Euler}$, Pare.

Passo 2: Identifique, dentre os nós de grau ímpar os que são adjacentes, e o número de ligações entre eles.

Passo 3: Enquanto houver nó adjacente de grau ímpar, faça:

$d(i) \leftarrow d(i) + 1$
 Se $(d(i) = \text{Par} \text{ ou } d(j)+1 = \text{Par})$ e $(NL(ij) = \text{Ímpar})$
 Então
 $NL(ij) \leftarrow NL(ij) + 1;$
 $d(j) \leftarrow d(j) + 1;$
 Senão $d(i) \leftarrow d(i) - 1;$

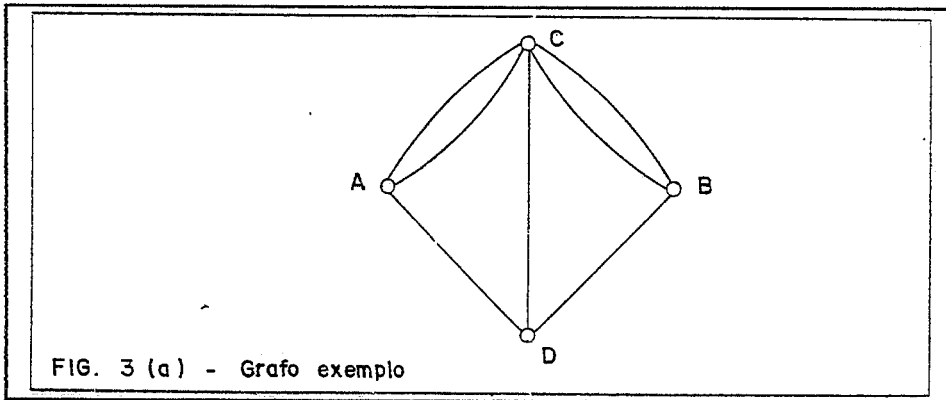
Exemplos:

(1) A conside o seguinte grafo abaixo:

Passo 1: $d(A) = 3, d(B) = 3, d(C) = 5, d(D) = 3$

Passo 2: $NL(AC) = 2, NL(AD) = 1, NL(BC) = 2, NL(BD) = 1,$
 $NL(CD) = 1$

¹Este algoritmo não possui a intensão de resolver o problema para um grafo qualquer



Passo 3:

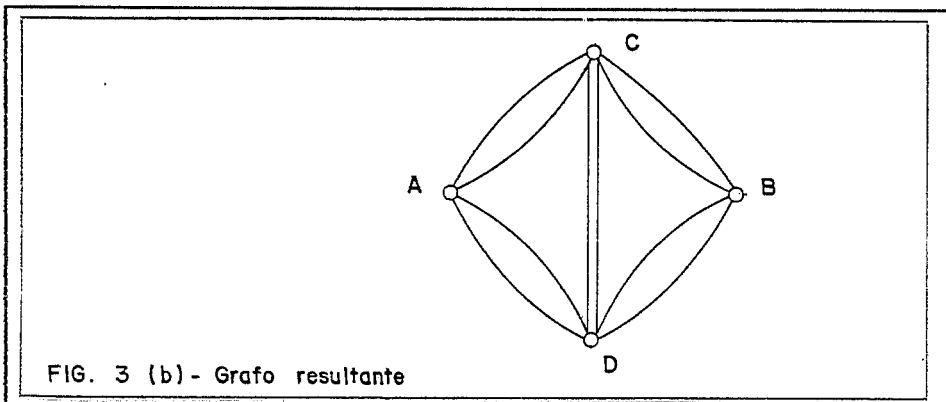
$$d(A) \leftarrow 4, d(C) + 1 = 6, NL(AC) = 2 \Rightarrow d(A) \leftarrow 3$$

$$d(A) \leftarrow 4, d(D) + 1 = 4, NL(AD) = 1 \Rightarrow d(D) \leftarrow 4, NL(AD) \leftarrow 2$$

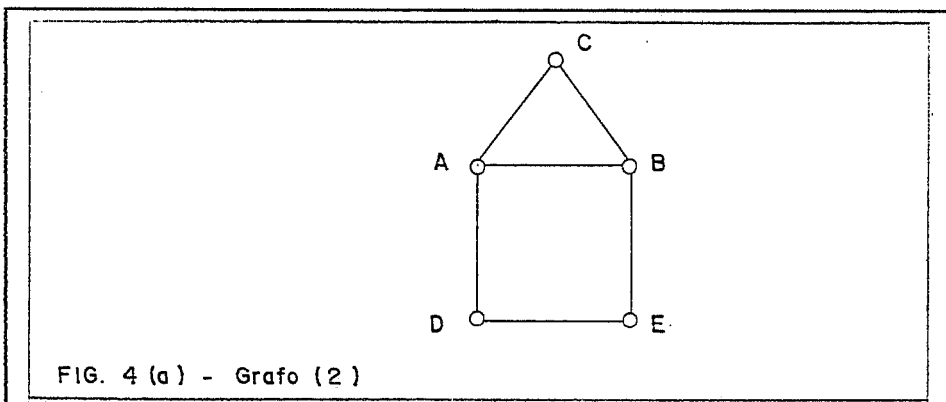
$$d(B) \leftarrow 4, d(C) + 1 = 6, NL(BC) = 2 \Rightarrow d(B) \leftarrow 3$$

$$d(B) \leftarrow 4, d(D) + 1 = 5, NL(BD) = 1 \Rightarrow d(D) \leftarrow 5, NL(BD) \leftarrow 2$$

$$d(C) \leftarrow 6, d(D) + 1 = 6, NL(CD) = 1 \Rightarrow d(D) \leftarrow 6, NL(CD) \leftarrow 2$$



(2) Considerare agora o grafo da Figura 4.a.:

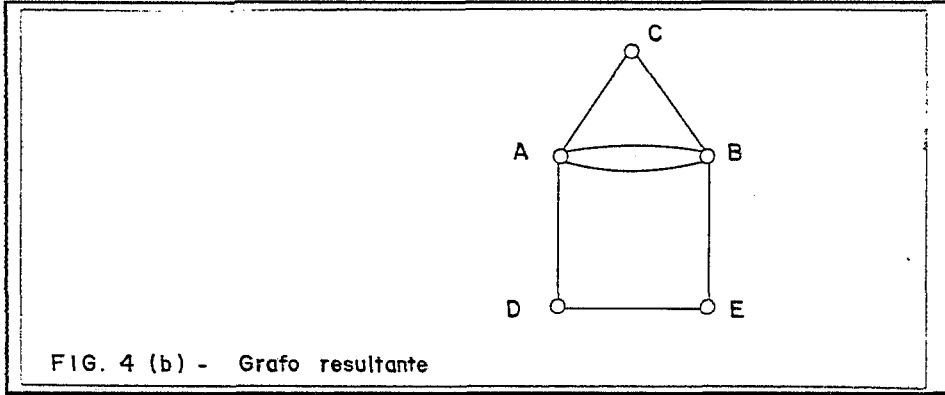


Passo 1: $d(A) = 3, d(B) = 3$

Passo 2: $NL(AB) = 1$

Passo 3: $d(A) \leftarrow 2, d(B) + 1 = 2, NL(AB) = 2 \Rightarrow d(B) \leftarrow 2, NL(AB) \leftarrow 2$

O grafo resultante será, pois:



O algoritmo Euler, possui complexidade polinomial, de ordem $O(n + m)$, onde n é o número de vértices de G , e m é o número de elos de G . O pior caso foi analisado no exemplo 1, onde todos os vértices do grafo possuem grau ímpar.

Admitimos agora que temos um grafo euleriano, G_E . Deseja-se realizar um ciclo euleriano em G_E . Mostraremos, a seguir, um algoritmo polinomial de ordem $O(m)$, onde m é o número de elos de G , que resolve este problema, proposto por EDMONDS & JOHNSON (1973).

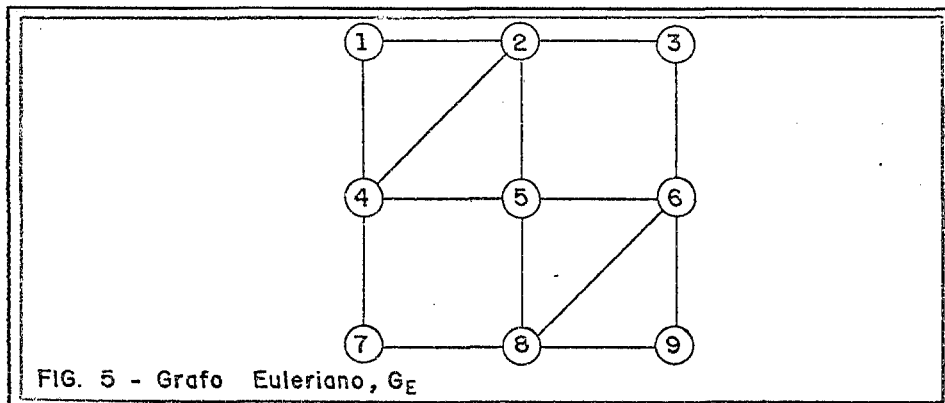
Algoritmo C - EULER

Passo 1: Partindo de s , escolha um vértice x daqueles ligados a s . Armazene (s, x) como elo visitado;

Passo 2: Continue tomando os vértices x' ligados a x não visitados, até que $x' \equiv s$. Armazene (x, x') como elo visitado;

Passo 3: Caso haja algum elo para visitar, vá para o passo 2, senão Termine.

Exemplo: Seja o grafo $G_E = (V, A)$, abaixo:



Admitindo que desejamos gerar um ciclo euleriano, a partir do vértice (1), temos então:

Passo 1: Pilha de elos

$E(1) \leftarrow (1, 2)$
 $E(2) \leftarrow (1, 4)$
 $E(3) \leftarrow (2, 3)$
 $E(4) \leftarrow (2, 4)$
 $E(5) \leftarrow (2, 5)$
 $E(6) \leftarrow (3, 6)$
 $E(7) \leftarrow (4, 5)$
 $E(8) \leftarrow (4, 7)$
 $E(9) \leftarrow (5, 6)$
 $E(10) \leftarrow (5, 8)$
 $E(11) \leftarrow (6, 8)$
 $E(12) \leftarrow (6, 9)$
 $E(13) \leftarrow (7, 8)$
 $E(14) \leftarrow (8, 9)$

Elo de Partida $E(1), (1,2)$

Passo 2:

Elo	$E()$	Elos Visitados - $E()$
(2,3)	3	1,3
(3,6)	6	1,3,6
(6,8)	11	1,3,6,11
(8,9)	14	1,3,6,11,14
(9,6)	12	1,3,6,11,14,12
(6,5)	9	1,3,6,11,14,12,9
(5,2)	5	1,3,6,11,14,12,9,5
(2,4)	4	1,3,6,11,14,12,9,5,4
(4,5)	7	1,3,6,11,14,12,9,5,4,7
(5,8)	10	1,3,6,11,14,12,9,5,4,7,10
(8,7)	13	1,3,6,11,14,12,9,5,4,7,10,13
(7,4)	8	1,3,6,11,14,12,9,5,4,7,10,13,8
(4,1)	2	1,3,6,11,14,12,9,5,4,7,10,13,8,2

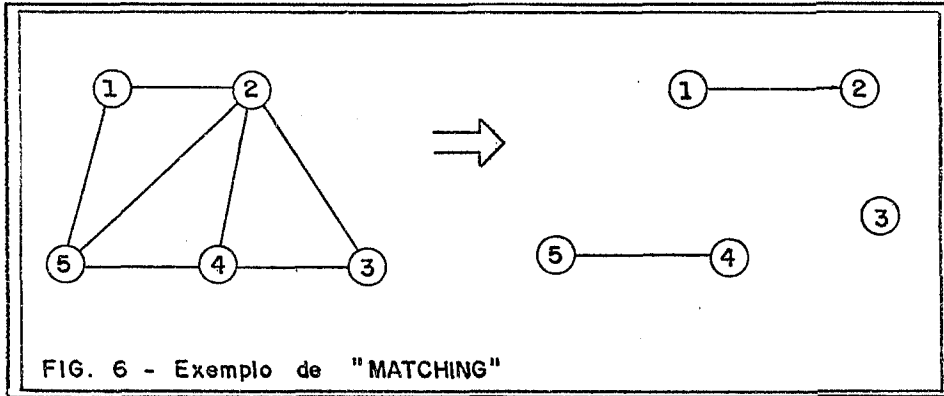
Passo 3: Número de elos visitados = 14 = Número de elos do grafo,
então: Termine.

Caso tenham um grafo qualquer $G = (V, E)$, se desejarmos realizar um passeio euleriano, teremos então que acrescentar ao algoritmo de EDMONDS & JOHNSON (1973), o Algoritmo Euler (Caso G possua em todos os vértices ímpares pelo menos um ímpar adjacente), ou um algoritmo de *Matching*, para o caso geral que será estudado a seguir como passo inicial, de modo a identificar o grafo e transformá-lo caso necessário.

II.4 O Problema de *Matching*

Em termos de grafos, um *Matching* pode ser definido como um conjunto de elos em um grafo, tal que cada vértice é incidente a no máximo um elo deste conjunto. - MINIEKA (1978).

Um exemplo de *Matching* pode ser visto no grafo $G = (V, E)$, mostrado na Figura 6.



O problema de *Matching* possui um grande número de aplicações, como exemplo citamos as agências de *Corações Solitários*, onde pode-se utilizar o conceito de *Matching* para definir o melhor emparelhamento entre casais que determinam suas características e a do parceiro(a) que deseja se relacionar; em problemas que envolvem relações de paridade, que necessitam de minimização/maximização de custos no casamento destas relações; determinação de pessoas a serviços de acordo com sua maior adaptação, dentre muitas outras.

II.4.1 Formulação do Problema de *Matching*

Seja $G = (V, E)$, o nosso grafo não direcionado. Seja ainda S um subconjunto tal que $S \subset E$. Um subconjunto $M \subset E$ é chamado de *Matching* de G se $d_M(i) \leq 1$ para todo $i \in V$. Ou seja, um *Matching* em um grafo é um conjunto de elos tais que dois elos do conjunto nunca incidem sobre o mesmo nó. Note que todo grafo possui trivialmente um *Matching*, uma vez que $d_G = 0$ para todo $i \in V$.

Sem perda de generalidade, podemos assumir que G é conexo, caso contrário os resultados podem ser aplicados separadamente aos componentes de G .

Um *Matching* é dito máximo se $|\mathcal{M}^*| = \max\{|\mathcal{M}|\}$, e \mathcal{M} é um *Matching*.

Fazendo c_j representar o peso ou o custo associado a um elo e_j , podemos definir o peso ou custo de um *Matching* \mathcal{M} como:

$$P(\mathcal{M}) = \sum_{e_j \in \mathcal{M}} c_j \quad (\text{II.7})$$

O *Matching* \mathcal{M}^* é chamado de *Matching* de custo máximo se $P(\mathcal{M}^*) \geq P(\mathcal{M})$ para todos os *Matchings* de \mathcal{M} . Outra generalização é permitir que $d_{\mathcal{M}}(i) \leq b_i$, onde $b_i \in \mathbb{Z}_+$. Dado um vetor $b = (b_1, \dots, b_m)$, \mathcal{M} é chamado de *b-Matching* se $d_{\mathcal{M}}(i) \leq b_i, \forall i \in V$.

Façamos $e \in x$ representarem $|A|$ - vetores, onde c_j é o custo de $e_j \in x_j$, igual a 1 se e_j é um *b-Matching*, e 0 caso contrário. O problema de determinar um *b-Matching* máximo é equivalente ao seguinte PPL:

$$(\text{PM}) \quad \text{Max} \sum_{j \in E} c_j^t x_j \quad (\text{II.8})$$

sujeito a:

$$A x \leq b \quad (\text{II.9})$$

$$x \in \{0, 1\}_E \quad (\text{II.10})$$

Onde: A é a matriz de incidência de G , b um vetor para o problema *b-Matching* de peso máximo ou *Matching* de máxima cardinalidade.

Ilustremos o seguinte exemplo para o caso do máximo *Matching* associado ao grafo da Figura 7.(a):

Se desejarmos o *1-Matching* do grafo acima, admitindo que o custo dos arcos e_j é unitário para todo $j \in E$. Temos que o modelo matemático para o problema seria:

$$\text{Max } x_{12} + x_{13} + x_{23} + x_{24} + x_{34}$$

s.a.

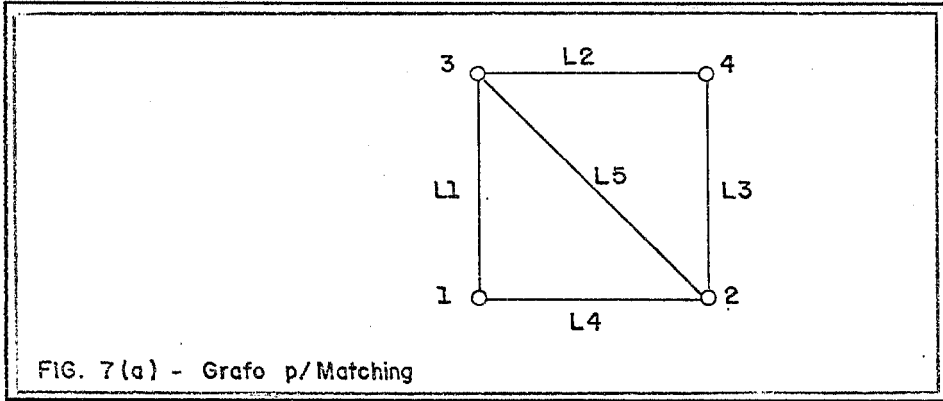
$$x_{12} + x_{13} \leq 1$$

$$x_{12} + x_{23} + x_{24} \leq 1$$

$$x_{12} + x_{23} + x_{34} \leq 1$$

$$x_{24} + x_{34} \leq 1$$

$$x \in \{0, 1\}_E$$



O problema acima também poderia ser visto da seguinte forma:

$$\text{Max } L_1 + L_2 + L_3 + L_4 + L_5$$

s.a.

$$L_2 + L_4 \leq 1$$

$$L_3 + L_4 + L_5 \leq 1$$

$$L_1 + L_2 + L_5 \leq 1$$

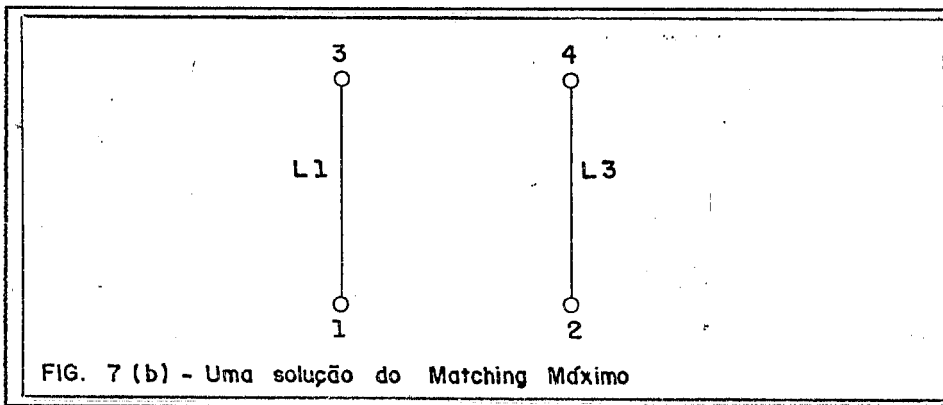
$$L_2 + L_3 \leq 1$$

$$L_j \in \{0, 1\} \mathbb{E}$$

Como vemos o problema possui mais de uma solução ótima, dentre elas temos:

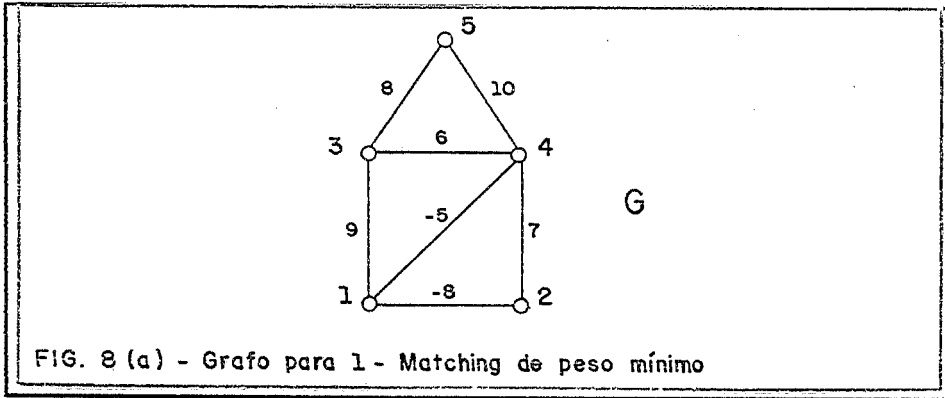
$$\begin{aligned} L_1 = L_3 = 1 \\ L_2 = L_4 = L_5 = 0 \end{aligned}$$

Portanto, o máximo *1-Matching* do grafo dado será pois:



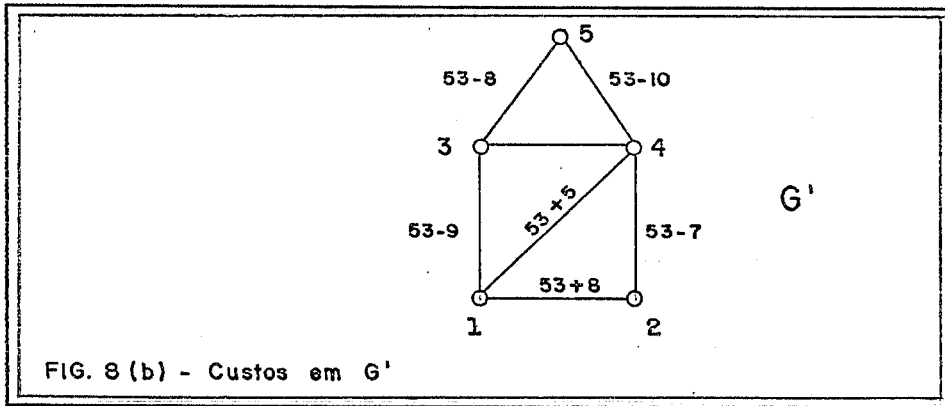
Há situações em que desejamos obter o *Matching* de custo mínimo. Para tanto, considere a soma dos módulos dos custos de cada elo do grafo, $\sum_j |c_j|$, $\forall j$ elo de G , em seguida gere um grafo G' onde todo elo terá seu custo alterado por $c'_j = \sum_j |c_j| - c_j$. Aplique um algoritmo de *1-Matching* máximo para G' , e este será o *1-Matching* mínimo de G .

Consideremos o seguinte exemplo:

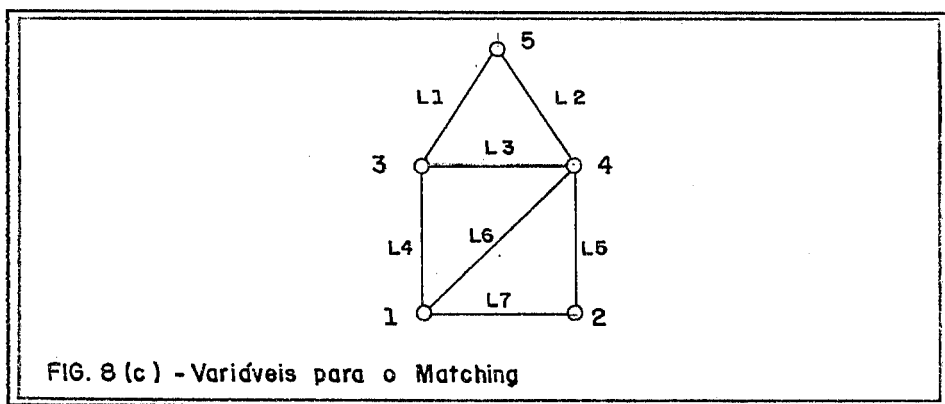


$$C_E = (8, 10, 6, 9, 7, -5, -8), \sum_j |C_j| = 53$$

Se desejarmos encontrar o mínimo *1-Matching* de G , temos então que o novo grafo G' , abaixo, é àquele que possui o mínimo *1-Matching* de G . Portanto,



$$C_{E'} = (45, 43, 47, 44, 46, 58, 61)$$



O PPL associado ao *1-Matching* máximo de G' , que é o mínimo *1-Matching* de G , será pois:

$$\text{Max } 45L_1 + 43L_2 + 47L_3 + 44L_4 + 46L_5 + 58L_6 + 61L_7$$

s.a

$$L_4 + L_6 + L_7 \leq 1$$

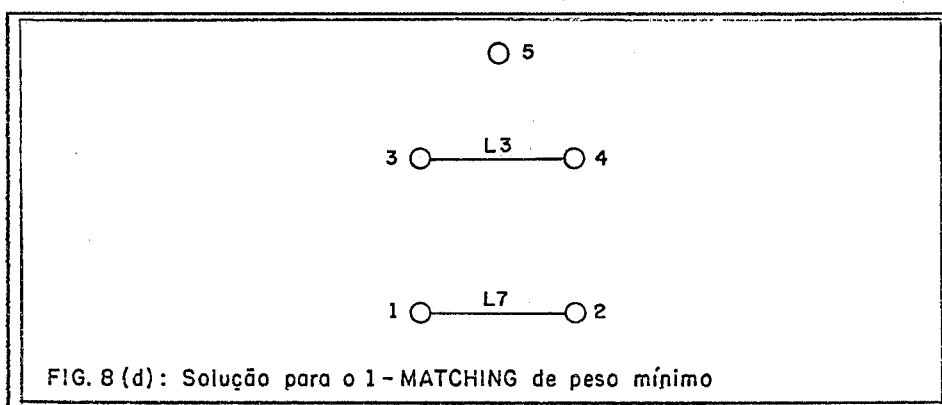
$$L_5 + L_7 \leq 1$$

$$\begin{aligned}
 L_1 + L_3 + L_4 &\leq 1 \\
 L_2 + L_3 + L_5 + e_6 &\leq 1 \\
 L_1 + L_2 &\leq 1 \\
 L_j &\in \{0, 1\}_{\mathbb{R}}.
 \end{aligned}$$

O problema, então, possui a seguinte solução:

$$\begin{aligned}
 L_3 = L_7 &= 1 \\
 L_1 = L_2 = L_4 = L_5 = L_6 &= 0
 \end{aligned}$$

Assim, em G o 1 -Matching de custo mínimo será dado por:



Em resumo, o b -Matching de peso mínimo é relativo a um grafo $G_{\mathcal{M}}^*$ que possui o máximo de b -emparelhamentos de peso total mínimo.

O estudo básico teórico do problema de b -Matching de mínimo custo é de fundamental importância na aplicação dos problemas de roteamento. A vasta literatura em algoritmos para solução de problemas, pode ser encontrada nos seguintes artigos e livros: EDMONDS & JOHNSON (1973), CHRISTOFIDES (1973), LAWLER (1976), MINOUX & GONDRAN (1979), SYSLO ET AL (1988), BURCKARD & DERIGS (1980), MINIEKA (1978), dentre outros.

Todo o estudo que completamos até aqui, determina subsídio para introduzirmos a seguir os conceitos e as soluções dos problemas de percursos em arcos.

II.5 O Problema do Carteiro Chinês (PCC)

Se admitirmos no problema de roteamento, que o caminhão possui capacidade suficiente para suprir uma determinada área em tempo que jamais ultrapasse aquele destinado à tripulação, temos que o problema é de construir um circuito euleriano de custo mínimo na área de trabalho, ou grafo $G = (V, E, A)$. Tal problema, como já dissemos, foi proposto por MEI-KO KWAN (1962), e a literatura a partir de então o chamou de *O Problema do Carteiro Chinês (PCC)*.

Desta definição, surgiram as diversas abordagens do PCC. Dentre elas temos:

- (a) O Problema do Carteiro Chinês em grafos não-orientados (PCC) – EDMONDS & JOHNSON (1973) ;
- (b) O Problema do Carteiro Chinês em grafos orientados (PCCd) — EDMONDS & JOHNSON (1973)
- (c) O Problema do Carteiro Chinês em grafos mistos (PCCm) – EDMONDS & JOHNSON (1973);
- (d) O Problema do Carteiro Chinês *Íngrime* (PCCI) – MINIEKA (1979)

Os problemas não-direcionados, direcionados e mistos, consideram apenas o esforço único na tarefa do carteiro em sua caminhada, admitindo que a cidade é plana. Já o carteiro *Íngrime*, admite o esforço do carteiro quando ele passa por ruas íngrimes, e o objetivo é também diminuir o esforço na entrega das correspondências. Esta idéia, completa fortemente o enfoque do problema de percurso em arcos sem restrições de capacidade ou tempo.

Nosso trabalho concentra-se no estudo dos casos (a),(b) e (c). Já o caso (d) pode ser melhor visto em GUAN (1984) e ZAW WIN (1989), onde nos dois artigos propõe-se uma metodologia de solução para o PCCI. Acreditamos que este estudo poderá dar uma visão ampla da solução dos problemas, dissimulando um pouco o problema misto, e abrindo espaço para futuras implementações computacionais.

Admitamos então que o carteiro deseja percorrer um grafo $G = (V, E, A)$ qualquer e fortemente conexo, com o comprimento do seu percurso total mínimo. Seja C_{ij} o custo de atravessar o arco (i, j) ou o elo ij , e seja x_{ij} o número de vezes que o carteiro deve passar pelo arco (i, j) ou elo ij . Como o percurso é direcionado, consideremos que cada elo possui um arco direcionado a ele associado. Portanto o modelo proposto por Stricker, GOLDEN ET AL (1983), define em linguagem matemática o problema do carteiro, da seguinte forma:

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (\text{II.11})$$

sujeito a:

$$\sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = 0 \quad (i = 1, \dots, n) \quad (\text{II.12})$$

O fluxo total em cada vértice é nulo;

$$x_{ij} + x_{ji} \geq 1 \quad \forall (i, j) \in E \cup A \quad (\text{II.13})$$

Cada arco/elo deve ser percorrido em uma direção ou na outra pelo menos uma vez;

$$x_{ij} \in \mathcal{Z}_+ \quad \forall (i, j) \in E \cup A \quad (\text{II.14})$$

O número de vezes que se deve atravessar o arco/elo é um número inteiro positivo.

II.5.1 O Problema Não Direcionado

Admitindo que para o grafo $G = (V, E, A)$, o conjunto $A \equiv \emptyset$, e portanto $G = (V, E)$ é conexo. Podemos dizer que é possível percorrer todos os elos de E desde que todo vértice possua grau par, ou seja: $\forall i \in V \mid d(i)$ é par.

Temos então dois tipos de arbordagens:

1. Todos os vértices de G têm grau par e
2. Nem todos os vértices de G têm grau par.

Considerando o primeiro caso, podemos notar, como já foi discutido, que o grafo já é euleriano, portanto, qualquer que seja o caminho que se tomar, cada elo será percorrido apenas uma vez, e esta solução é de custo mínimo, saindo de um vértice s e retornando a ele.

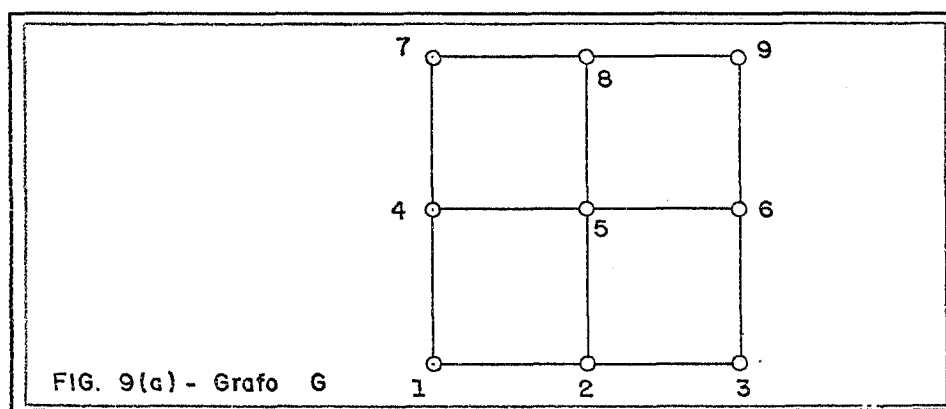
Provavelmente, do ponto de vista prático, é necessário que se construa rotas que aproveitem ao máximo os caminhos que determinam sequência de elos que mantêm uma continuidade no trabalho, desfavorecendo a realização de monobras

desagradáveis, como por exemplo, dobrar a esquerda e retornos em U na mesma rua.

O problema acima foi abordado por BODIN & KURSH (1979) e mais tarde por MCBRIDGE (1982). No primeiro caso, foi desenvolvido um algoritmo baseado no problema de atribuições, onde se determinava custos para conjunto de arcos Predecessores-Sucessores candidatos à formação de manobras. BODIN & KURSH (1979) resolveram o problema, como um problema de atribuição com gargalo. No caso de MCBRIDGE (1982), ele percebeu que a solução do Problema de Atribuições com Gargalo - BURCKARD & DERIGS (1980), na geração do ciclo euleriano em grafo euleriano, poderia gerar retornos em U e monobras à esquerda desnecessárias. Propôs, então, um algoritmo que busca combinar ciclos, onde eles são ditos combinados em uma interseção se os atributos de saída são modificados naquela interseção para criar um novo ciclo. Quando se combina os ciclos, são considerados somente os segmentos de rua nos ciclos de modo que se permita a criação de um único ciclo. Este algoritmo é executado antes de se executar o Problema de Atribuição com Gargalo. Este procedimento visa reduzir o número de execuções do algoritmo para atribuição dos Predecessores-Sucessores.

Recomendamos a leitura minuciosa deste texto, e achamos interessante a sua aplicação em situações do tipo rotear-primeiro-agrupar depois. A seguir, mostraremos um exemplo que introduz a idéia dos algoritmos que discutimos acima, de modo a formar uma rota euleriana com critério. Apesar do exemplo ser intuitivo, pretendemos, com ele alertar ao projetista de rotas a necessidade de estudar este problema detalhadamente.

Exemplo: Considere o seguinte grafo da Figura 9(a).

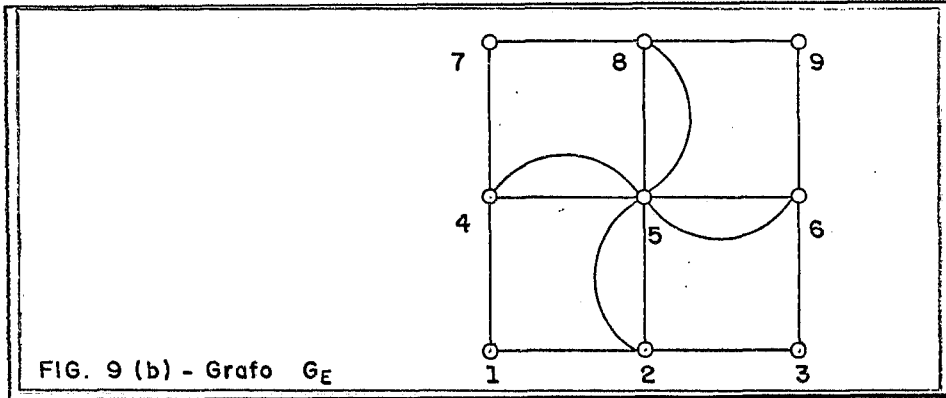


Partindo de 5, desejamos então gerar um percurso euleriano que procure ao máximo manter a direita do veículo, de modo que fique mais fácil de manobrá-lo, e que mantenha o máximo possível em caminhos retilíneos, sem retornos em U.

O grafo euleriano de G , será, pois:

A solução do problema pode então ser vista da seguinte forma:

- {5 ↔ 4 ↔ 7 ↔ 8 ↔ 5 ↔ 2 ↔ 1 ↔ 4 ↔ 5 ↔ 6 ↔ 3 ↔ 2 ↔ 5 ↔ 6 ↔ 9 ↔ 8 ↔



5}

- Retornos em $U = 0$
- Curvas a esquerda = 3
- Ciclos = 4

Para o segundo caso, onde o grafo não é euleriano, em algum passeio do carteiro, o número de vezes que ele entra em um vértice é o número de vezes que ele deve sair deste vértice. Consequentemente, se o vértice x não possui grau par, então pelo menos um vértice x deve ser repetido pelo carteiro, MINIEKA (1979).

Faça x_{ij} o número de vezes que o elo ij é repetido no grafo. Então o elo ij é atravessado $x_{ij} + 1$ vezes pelo carteiro. É claro que, x_{ij} deve ser um inteiro não negativo, como vimos. Note, também que x_{ij} não contém informação sobre direção de travessia do elo ij .

Construa um novo grafo $G^* = (V, E^*)$ que contém $x_{ij} + 1$ cópias de cada elo ij no grafo G . Como podemos notar, um ciclo de euler no grafo G^* corresponde à rota do carteiro no grafo G .

Como devemos pressupor, o carteiro precisa selecionar valores das variáveis x_{ij} , tais que:

- (a) O grafo G^* é euleriano;
- (b) $\sum c_{ij} x_{ij}$, o comprimento total dos elos repetidos (paralelos), é minimizado.

Se o vértice x é de grau ímpar no grafo G , então um número ímpar de elos incidentes ao vértice x tem grau par. Igualmente, se o vértice x é um vértice grau par no grafo G , então um número par de elos incidentes ao vértice x deve ser repetido pelo carteiro, tal que no grafo G^* o vértice x tem grau par.

Se traçarmos uma *Corrente* (Sequência de elos adjacentes) tão longa quanto possível, mesmo que seja de elos repetidos iniciando de um vértice de grau ímpar, esta corrente deve necessariamente terminar em outro vértice de grau ímpar.

Assim, os elos repetidos das correntes serão os vértices iniciais e finais onde os vértices possuam grau ímpar. Deve-se notar que esta corrente pode conter um vértice de grau par como um de seus vértices intermediários, conseqüentemente o carteiro deve decidir:

- (a) Quais vértices de grau ímpar devem ser juntados ou ligados por uma corrente de elos repetidos;
- (b) A composição precisa de cada *Corrente*.

O carteiro pode determinar cada par de vértices de grau ímpar que serão unidos por uma corrente de elos repetidos, como segue: Construir um grafo $G' = (V, E')$, cujos conjuntos de vértices em G e cujo conjunto de elos contém um elo unindo cada par de vértices. Faça o custo de cada elo ser igual a um número muito alto menos o comprimento do menor caminho entre os dois vértices correspondentes no grafo G .

Em seguida o carteiro pode encontrar o *1-Matching* de maior custo para o grafo G' usando algoritmo de *1-Matching* de custo máximo. Desde que o grafo G' tem um número par de vértice e cada par de vértice é unido em G' por um elo, o *1-Matching* de peso máximo cobrirá cada vértice exatamente uma vez. Este *1-Matching* liga dois vértices de grau ímpar do grafo G . Os elos na menor *Corrente* ligando um par de vértices *Casados* de grau ímpar serão repetidos pelo carteiro. Desde que este *1-Matching* tem peso total máximo, a rota do carteiro deve ter o comprimento total mínimo.

O algoritmo pode ser assim resumido, CHRISTOFIDES (1973):

Passo 1: Faça $[c_{ij}]$ ser a matriz custo do grafo G . Usando algoritmo de menor caminho, construir a matriz $[d_{ij}]$, onde d_{ij} é o custo do menor caminho do vértice i ao vértice j .

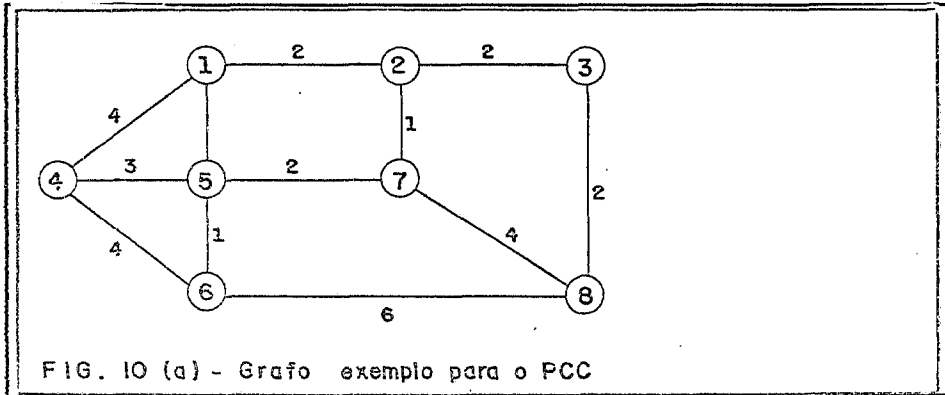
Passo 2: Encontrar o par de vértices de grau ímpar correspondendo a \mathcal{M}^* , aos quais estão ligados por uma cadeia de menor distância — *1-Matching* de mínimo custo.

Passo 3: Se o vértice α corresponde a outro vértice β , identifique o caminho de mínimo custo $\mu_{\alpha\beta}$ (de α para β) correspondente ao custo $d_{\alpha\beta}$ do passo 1. Insira elos artificiais em G correspondentes a elos em $\mu_{\alpha\beta}$ e repita para todos os outros caminhos no *1-Matching* \mathcal{M}^* , para o sub-grafo $G'_{\mathcal{M}^*}$.

Passo 4: A soma dos custos da matriz $[c_{ij}]$ de todos os elos em $G'_{\mathcal{M}^*}$ — tomando o custo de um elo artificial para ser o custo do elo real em paralelo a ele — é o custo mínimo de um circuito atravessando G . O número de vezes que este circuito atravessa o elo ij , x_{ij} , é o número total de elos em paralelo entre i e j em $G'_{\mathcal{M}^*}$.

Note que desde o passo 2 estamos usando o *1-Matching* mínimo, e que não há dois caminhos mínimos μ_{ij} e μ_{pq} em tal *matching*. Isto significa que o grafo G^* não tem mais que dois elos em paralelo entre dois vértices, isto é, o circuito ótimo nunca atravessa nenhum elo de G mais que duas vezes.

Exemplo: Seja o grafo da Figura 10(a). Desejamos encontrar o grafo G_E de custo mínimo.

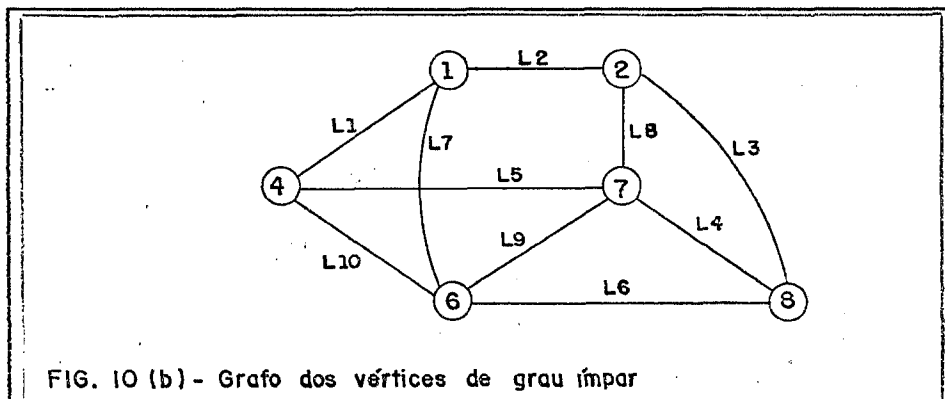


Solução:

Passo 1: Matriz de distâncias mínimas. (Simétrica, $d_{ij} = d_{ji}$)

$$[d_{ij}] = \begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & - & 2 & 4 & 4 & 1 & 2 & 3 & 7 \\ 2 & & - & 2 & 6 & 3 & 4 & 1 & 4 \\ 3 & & & - & 6 & 3 & 4 & 3 & 2 \\ 4 & & & & - & 3 & 4 & 5 & 9 \\ 5 & & & & & - & 1 & 2 & 6 \\ 6 & & & & & & - & 3 & 6 \\ 7 & & & & & & & - & 4 \\ 8 & & & & & & & & - \end{array}$$

Passo 2: Grafo G' de vértices de grau ímpar



$$E = (4, 2, 4, 4, 5, 6, 2, 8, 3, 4), \sum_j l_j = 42$$

$$E' = (38, 40, 38, 38, 37, 36, 40, 34, 39, 38)$$

Temos então o seguinte PPL associado ao problema de *1-Matching* de custo máximo para G' :

$$\text{Max } 38 L_1 + 40 L_2 + 38 L_3 + 38 L_4 + 37 L_5 + 36 L_6 + 40 L_7 + 34 L_8 + 39 L_9 + 38 L_{10}$$

s.a.

$$L_1 + L_2 + L_7 \leq 1$$

$$L_2 + L_3 + L_8 \leq 1$$

$$L_1 + L_5 + L_{10} \leq 1$$

$$L_6 + L_7 + L_9 + L_{10} \leq 1$$

$$L_4 + L_5 + L_8 + L_9 \leq 1$$

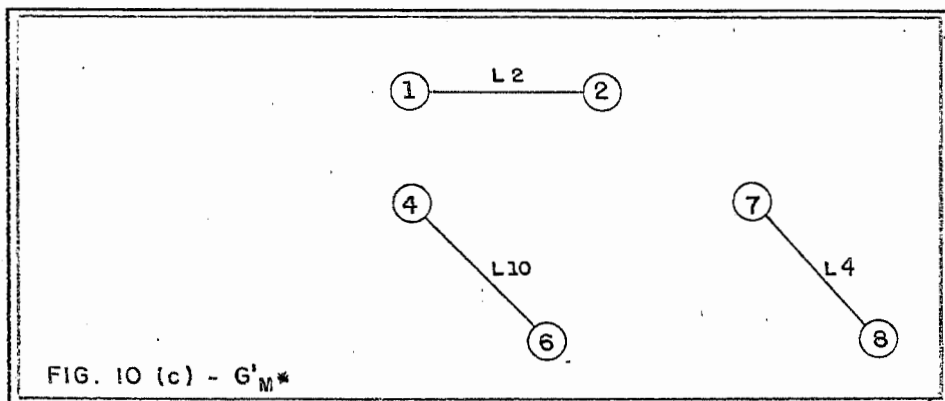
$$L_3 + L_4 + L_6 \leq 1$$

$$L_j \in \{0, 1\}$$

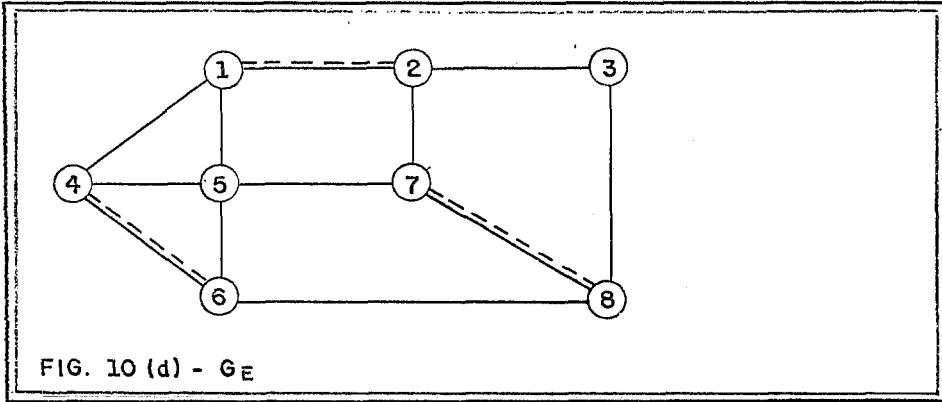
A solução para este problema é pois:

L	1	2	3	4	5	6	7	8	9	10
L()	0	1	0	1	0	0	0	0	0	1

Passo 3:



Passo 4:



II.5.2 O Problema Direcionado (PCCd)

Neste caso, o nosso carteiro deverá estar motorizado, haja vista ter que realizar suas entregas passando por todas as vias de sua área de serviço, respeitando o sentido de tráfego de cada via. Neste caso, a área de serviço do carteiro possui todas as suas vias com mão única.

Ao contrário do problema não-direcionado, o problema direcionado pode não possuir solução. Basta que de quaisquer dos vértices de $G = (V, A)$ apenas haja arcos saindo ou chegando nele, ou seja, o grafo $G = (V, A)$ não possui componente fortemente conexa. Tal condição é necessária e suficiente para que seja possível um circuito do carteiro partindo de um determinado ponto e retornando a ele.

Assim como é necessário verificar a conectividade do grafo $G = (V, E)$, é também necessário verificar a forte conectividade, ou melhor, a existência de um componente fortemente conexo no grafo $G = (V, A)$.

O melhor algoritmo de forte conectividade da literatura, pode ser encontrado em TARJAN (1972). Para efeitos de análise em redes viárias, implementamos um algoritmo baseado em caminhos mínimos, que permite chegar a existência de um componente fortemente conexo. A discussão deste, não será realizada, pois foge do escopo deste trabalho.

Então, para o caso do problema direcionado do carteiro, dois casos também devem ser considerados:

[1] O grafo é euleriano ($d^+(i) = d^-(i), \forall i \in V$)

[2] O grafo não é euleriano.

No primeiro caso, se o grafo é euleriano, então é possível para o carteiro realizar a sua rota sem repetir arco, isto é, a solução ótima do carteiro é a rota de euler.

Um ciclo euleriano no grafo $G_E = (V, A)$, pode ser encontrado usando a mesma técnica abordada no algoritmo G-Euler, ou até mesmo no algoritmo de MCBRIDGE (1982).

No segundo caso, como antes, faça f_{ij} ser o número de vezes que o carteiro repete o arco (i, j) . O carteiro deseja valores inteiros não-negativos de f_{ij} tal que minimize:

$$\sum_i \sum_j c_{ij} f_{ij} \quad (\text{II.15})$$

que é o comprimento total dos arcos repetidos tais que eles entram e saem de cada vértice i o mesmo número de vezes, onde:

$$d^-(i) + \sum_j f_{ji} = d^+(i) + \sum_j f_{ij} \quad (\text{II.16})$$

Reescrevendo a equação (II.16),

$$\sum_j (f_{ij} - f_{ji}) = d^-(i) - d^+(i) \equiv D(i) \quad (\text{II.17})$$

Assim, o carteiro deseja minimizar a expressão (II.15) tal que a equação (II.17) seja satisfeita para todo vértice $i \in V \mid G = (V, A)$. Este problema de minimização é meramente o problema de fluxo de custo mínimo. Nos vértices onde $D(i) < 0$ são origens com demandas igual a $-D(i)$. Os vértices onde $D(i) > 0$ são destinos com suprimento igual a $D(i)$. Os vértices com $D(i) = 0$ são vértices intermediários. Todas as capacidades dos arcos são infinitas.

Este problema de fluxo de redes de custos mínimo pode ser resolvido pela colocação de dois nós, um *Super-Origem* e outro *Super-Destino*, no grafo; onde o *Super-Origem*, está conectado com os nós origem, e os nós destinos conectados ao nó *Super-Destino*. Faça a capacidade de cada arco saindo do vértice *Super-Origem* igual ao suprimento do seu vértice terminal. Faça a capacidade de cada arco dentro de *Super-Destino* igual à demanda do seu vértice inicial. Encontre, então um fluxo de custo mínimo que satisfaz todas as necessidades dos vértices origem e destino usando um algoritmo apropriado – ver Apêndice 2, o algoritmo OUT-OF- KILTER.

Desde que todo o lado direito das equações em (2.17) são inteiros, sabemos que o fluxo de custo mínimo produzirá valores ótimos para $f(i, j)$ que são inteiros não-negativos.

Após encontrar os valores inteiros ótimos para as variáveis $f(i, j)$, crie um grafo G^* com $f(i, j) + 1$ do arco (i, j) para todo $(i, j) \in A$. Pela equação (2.15) o grafo G^* é euleriano. O algoritmo C-Euler pode então ser aplicado de modo a ser

encontrada a rota do carteiro. Desde que os valores ótimos de $f(i, j)$ minimizam a expressão 2., esta rota deve ser admitida como a rota ótima do carteiro no grafo direcionado.

Então, o algoritmo para o carteiro chinês direcionado pode ser descrito da seguinte forma:

Passo 0: Examine o grau dos vértices, e determine os vértices origem e destino.
Caso o grafo G seja simétrico, vá para o Passo 4.

Passo 1: Crie um vértice O , *Super-Origem*, e o ligue a todo vértice destino. Atribua sua capacidade inferior ao suprimento de cada destino d . Faça o mesmo ligando os vértices origem, ao *Super-Destino*, D , e atribua a sua capacidade inferior à demanda no origem o .

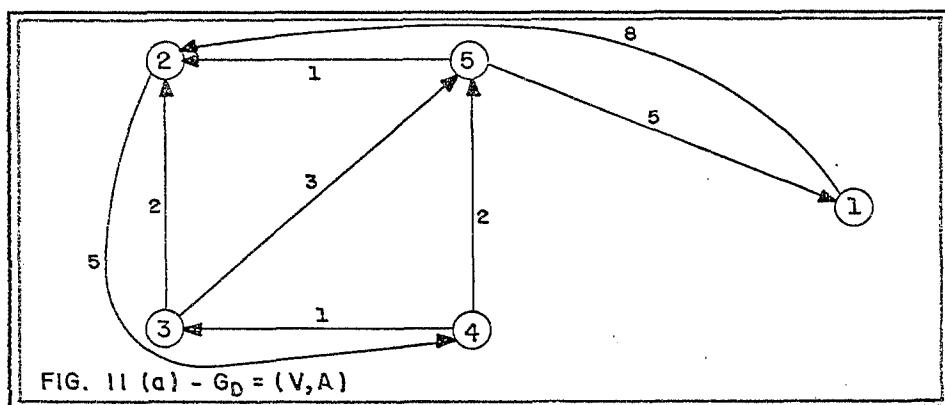
Passo 2: Aplique o algoritmo OUT-OF-KILTER ao grafo $G^h = (V, A)$

Passo 3: Faça G receber G^h se $\forall(i, j), f(i, j) > 0$ então repita a existência do arco (i, j) em G , $f(i, j) + 1$ vezes.

Passo 4: Aplique o algoritmo de C-Euler ou de McBridge.

Passo 5: A rota apresentada é ótima.

Exemplo: Seja o grafo da Figura 11(a):



Desejamos encontrar a rota ótima do carteiro chinês em G_D , partindo do vértice (1).

Solução:

Passo 0: $d^+(1) = 1, d^-(1) = 1 \Rightarrow D(1) = 0$ [Indeterminado]

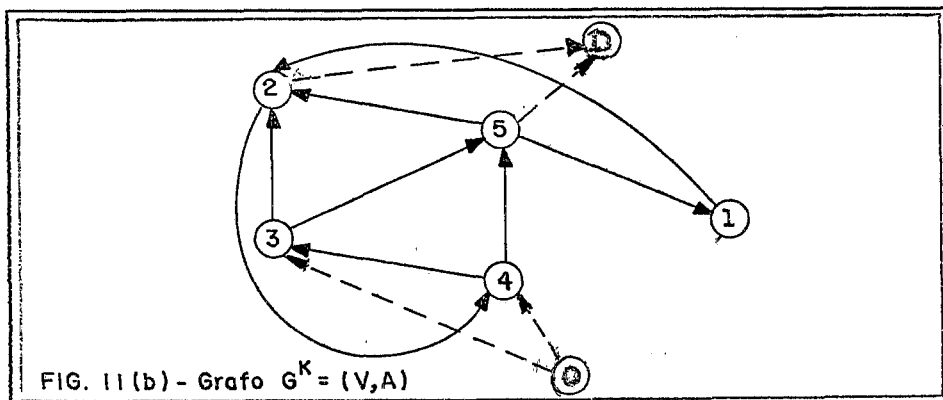
$d^+(2) = 3, d^-(2) = 1 \Rightarrow D(2) = -2$ [Origem]

$d^+(3) = 1, d^-(3) = 2 \Rightarrow D(3) = 1$ [Destino]

$d^+(4) = 1, d^-(4) = 2 \Rightarrow D(4) = 1$ [Destino]

$d^+(5) = 2, d^-(5) = 2 \Rightarrow D(5) = 0$ [Indeterminado]

Preparação para algoritmo Out-of-Kilter ...



l_{ij} - Fluxo mínimo (Limite Inferior)

u_{ij} - Fluxo máximo (Limite Superior)

c_{ij} - Custo associado ao arco

Arco	$[l_{ij}, u_{ij}, c_{ij}]$
(1,2)	(0, ∞ , 8)
(2,4)	(0, ∞ , 5)
(3,2)	(0, ∞ , 3)
(3,5)	(0, ∞ , 2)
(3,D)	(1, ∞ , 2)
(4,1)	(0, ∞ , 2)
(4,3)	(0, ∞ , 1)
(4,D)	(1, ∞ , 0)
(5,1)	(0, ∞ , 5)
(5,2)	(0, ∞ , 1)
(O,2)	(2, ∞ , 0)

Passo 2: Aplicação do algoritmo OUT-OF-KILTER, BAZARAA (1977)

$$f(1,2) = f(3,2) = f(3,5) = f(4,1) = f(4,5) = f(5,1) = f(5,2) = 0$$

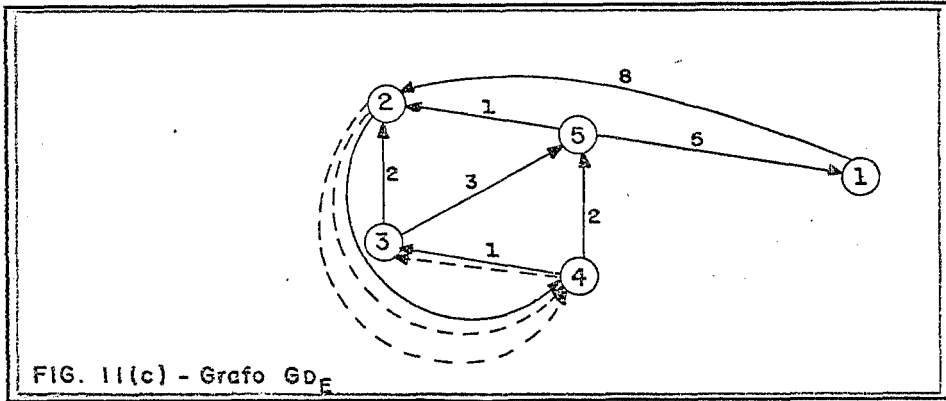
$$f(2,4) = 2, f(4,3) = 1$$

Passo 3: Grafo G_{DB}

Passo 4: Rota usando C-Euler,

$$\{(1) \rightarrow (2) \rightarrow (4) \rightarrow (5) \rightarrow (2) \rightarrow (4) \rightarrow (3) \rightarrow (2) \rightarrow (4) \rightarrow (3) \rightarrow (5) \rightarrow (1)\}$$

O problema do carteiro chinês em um grafo direcionado, pode também ser abordado como um problema de transporte, como foi estudado por EDMONDS & JOHNSON (1974) e amplamente discutido em BODIN & BELTRAMI (1974), onde é solucionado o problema de roteamento em arcos usando uma rotina chamada *Branch*. Uma outra abordagem de solução para o problema, podemos encontrar



em YAXIONG & YONGCHANG (1988), que utiliza a idéia de condições de otimalidade e folgas complementares, bastante semelhante ao que empregamos aqui ao utilizarmos o algoritmo primal-dual *OUT-OF-KILTER*.

Uma outra abordagem para o problema direcionado, pode ser encontrada em CHRISTOFIDES ET AL (1986), onde este formula o problema do carteiro rural, no qual deseja-se passar por todos os arcos de um sub-grafo de G , $S \subset G$, onde G é direto. Tal problema é do tipo, \mathcal{NP} -Hard, LENSTRA & RINNOOY KAN (1981), e pois, foi abordado através de considerações heurísticas.

II.5.3 O Problema Misto (PCCm)

A maioria das aplicações práticas de roteamento em arcos, leva em consideração a existência a de uma rede viária onde há a presença de ruas com mão-dupla e mão-única. Na situação em que o carteiro esteja motorizado, a abordagem do problema seria passar por todas as ruas desta malha viária considerada, obedecendo as direções de tráfego das ruas da cidade. Este é, pois, o Problema Misto do Carteiro Chinês, que procura dar uma direção única de travessia dos arcos, de modo que seja viável a realização do percurso, caso exista algum possível.

Para o PCC não direcionado há uma rota usando cada elo do grafo apenas uma vez, se a seguinte propriedade for satisfeita: O grau de cada vértice deve ser par. Já para o PCCd, a propriedade que deve ser satisfeita é a seguinte: Para cada vértice no grafo, os arcos incidentes são iguais aos arcos que saem de qualquer vértice. Para o grafo misto, se ambas as propriedades são satisfeitas, então a rota euleriana existe. Assim, um método de solução para o PCCm consiste de encontrar o custo mínimo de aumento do grafo que satisfaz as propriedades suficientes, e então identificam a rota sobre o grafo aumentado.

O algoritmo de EDMONDS & JOHNSON (1973) para o PCCm, consiste essencialmente de dois passos: O primeiro, é para modificar o grafo e tornar o grau de cada vértice par, tratando cada arco como elo. O segundo, é tornar o grau de incidência igual ao grau de saída para cada vértice do arco, enquanto preserva a propriedade de que cada vértice tem grau par. Uma rota pode ser facilmente encontrada pelo grafo resultante. A seguir especificamos as subrotinas de EDMONDS & JOHNSON (1973), propostas por FREDERICKSON (1979).

ALGORITMO MIXED 1

Entrada: Um grafo misto $G = (V, E, A)$ e a função de custo dos arcos e elos sobre $E \cup A$.

Saída: Uma rota que cobre todos os elos e arcos de G .

1. Chame a rotina `EVENDEGREE`;
2. Chame a rotina `INOUTDEGREE` com a saída de `EVENDEGREE`;
3. Chame a rotina `EVENPARITY` com a saída de `INOUTDEGREE`;
4. Encontre uma orientação para os elos não-direcionados no grafo resultante, e então atravesse todos os arcos e elos orientados.

Rotina EVENDEGREE

Entrada: Um grafo misto $G = (V, E, A)$ com a função custo sobre $E \cup A$.

Saída: Um grafo misto $G = (V, E, A)$ onde $E \subseteq E'$, $A \subseteq A'$ e o grau de cada vértice, ignorando a direção dos arcos, é par.

1. Identifique os vértices de grau ímpar;
2. Encontre todos os caminhos mínimos entre os vértices de grau ímpar ignorando as direções dos arcos;
3. Desenvolva um algoritmo de *1-Matching* de custo mínimo dos vértices de grau ímpar usando as distâncias mais curtas dos caminhos mais curtos;
4. Coloque os multiconjuntos A' em A e E' em E . Insira arcos nos menores caminhos usados no *Matching* em A' e elos no menor caminho usado no *Matching* em E' .

Observe que os três primeiros passos da rotina EVENDEGREE, está intimamente relacionado ao problema não-direcionado do carteiro chinês.

A rotina INOUTDEGREE faz cópias adicionais de arcos e elos e orienta alguns elos, de modo a fazer com que cada vértice $i \in V$, possua $d^+(i) = d^-(i)$. EDMONDS & JOHNSON (1973) formularam o problema como um problema de fluxo em redes de mínimo custo, no qual o custo dos arcos adicionais e elos é minimizado. Os conjuntos E_1 e E_2 de elos direcionados correspondem ao conjunto E que introduzimos. O conjunto E_2 será usado para determinar a orientação daqueles que são orientados pela rotina INOUTDEGREE. O conjunto E , será usado para determinar quantas cópias adicionais de um elo orientado serão necessárias.

Rotina INOUTDEGREE

Entrada: Grafo misto $G = (V, E, A)$ com função custo c_{ij} .

Saída: Um multiconjunto M contendo os arcos de G pelo menos uma vez e elos orientados, tais que para cada vértice i , $d^+(i) = d^-(i)$. Também, um conjunto $U \subseteq E$ de elos que não foram orientados e inseridos em M .

1. Faça E_1 ser um multiconjunto de elos orientados, tais que para cada elo (v, ω) em E há dois elos orientados (v, ω) e (ω, v) em E_1 . Faça E_2 ser composto na mesma forma. Faça $E_i = E_1 \cup E_2 \cup A_j$;
2. Para cada vértice v , faça b_v o número de arcos em A , diretos, entrando em v menos o número de arcos em A saindo de v ;
3. Minimize z sujeito a

$$z = \sum_{x \in A} c_s x_s + \sum_{x \in E_1} c_s x_s$$

x_s , um inteiro não negativo para $s \in A \cup E_1$

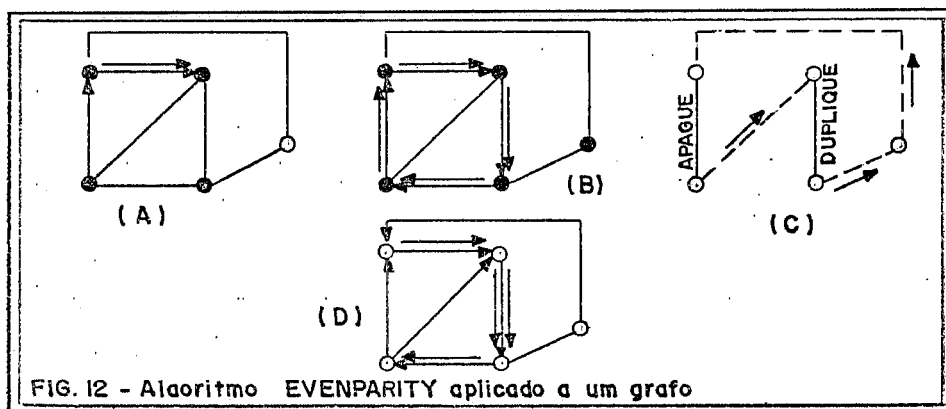
$x_s \in \{0, 1\}$ para $s \in E_2$

$\sum\{x_s \mid s \in E_t \text{ é direto saindo de } v\} - \sum\{x_s \mid s \in E_t \text{ é direto entrando em } v\} = b_v$, onde x_s é o número de cópias adicionais do arco s de A , ou o número de cópias de elos orientados s de E_1 ou E_2 ;

4. Inicialize U para ser vazio e M igual a A . Para cada arco s em A , insira x_s cópias adicionais de s dentro de M . Para cada elo orientado s em E_1 , insira x_s cópias de s em M ;
5. Para cada par de elos correspondentes orientados $s_1 = \langle \omega, V \rangle$ em E_2 , se $x_{s_1} + x_{s_2} = 1$, então insira s_1 e x_{s_2} cópias de s_2 em M . Senão, insira (v, ω) em U .

A complexidade da rotina INOUTDEGREE é $O(|A| \max\{|A|, |E|\}^2)$, FREDERICKSON (1979).

EDMONDS & JOHNSON (1973) mostraram que há uma solução de mínimo custo para suas restrições de fluxo em redes que preservam a propriedade de cada vértice que tem grau par. Entretanto, notamos que as restrições do fluxo em redes não são suficientes para selecionar uma das soluções que preservam o grau par. Assim, a rotina INOUTDEGREE não mantém necessariamente o grau par. Como um exemplo, considere o grafo na Figura 12. Um aumento que deve ser produzido pela rotina INOUTDEGREE e que poderia satisfazer as condições de EDMONDS & JOHNSON é mostrado na Figura 12(b). Observamos que claramente não preserva o grau par. Especificaremos agora a rotina EVENPARITY, para o qual, o custo dos arcos adicionais e elos não irá aumentar. A saída da EVENPARITY será um grafo satisfazendo as propriedades em que cada vértice o grau é par, e o grau de entrada é igual ao grau de saída.



Rotina EVENPARITY

Entrada: Grafo $G(V, E, A)$ e os conjuntos M e U , os quais são saída para o algoritmo INOUTDEGREE.

Saída: Conjunto M' e U' satisfazendo os mesmos critérios que os conjuntos M e U de INOUTDEGREE, tal que os vértices em (V, U', M') são de grau par.

1. Identifique o conjunto de vértices V' com respeito a (V, U, M) . Faça M' igual a M , e U' igual a U ;
2. Faça $M'' \subset M$ ser o conjunto de arcos adicionais e elos orientados criados pelo INOUTDEGREE;
3. Chame ADJUSTCYCLES.

A rotina ADJUSTCYCLES identifica ciclos consistindo de caminhos alternados em M'' e U , com cada caminho apoiado no final de cada vértice ímpar. Enquanto os ciclos são cobertos, os arcos são orientados sobre os ciclos que serão tanto duplicados quanto apagados, e elos nos ciclos serão orientados. Isto é feito tal que a paridade aos vértices ímpares é mudada, enquanto mantém o grau de entrada igual ao grau de saída para cada vértice.

Rotina ADJUSTCYCLES

1. Enquanto V' não está vazio;
2. Selecione v de V' , e chame v_s de v
3. Enquanto v está em V'
4. Remova v de V'
5. Repita
 - . Remova (v, ω) de M''
 - . Se (v, ω) é direto entrando em ω ,
 - . Então
 - . Insira uma cópia duplicada de (v, ω) em M' ;
 - . Senão apague uma cópia de (ω, v) de M'
 - . Faça v igual a ω
6. Até que v esteja em V'
7. Remova v de V'
8. Repita
 - . Remova (v, ω) de U'
 - . Oriente (v, ω) de v para ω , e insira-o em M'
 - . Faça $v = \omega$
 Até que v esteja em V' ou $v = v_s$

LEMA, FREDERICKSON (1979): *Faça M e U ser saída da rotina INOUTDEGREE para o qual o grafo de entrada é G' , um grafo com todo vértice tendo grau par. Então o algoritmo EVENPARITY terá como saída M' e U' de custo total igual àquele de M e U , tal que cada vértice tem grau par com respeito a $M' \cup U'$.*

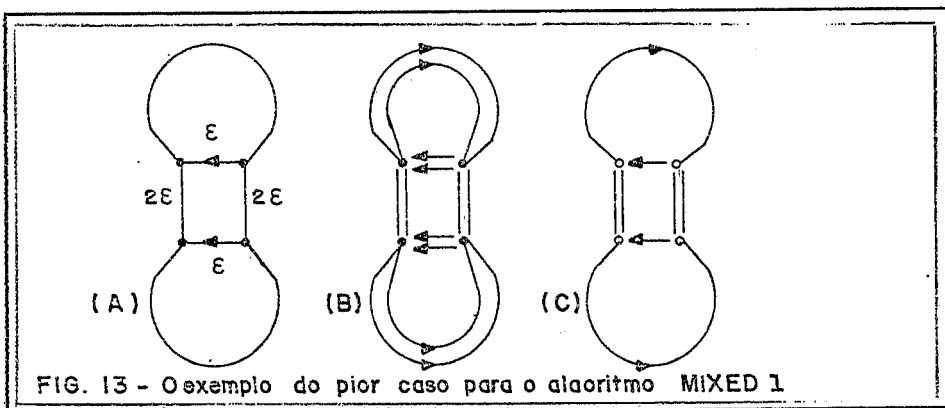
Consideramos um exemplo de EVENPARITY aplicado ao grafo na Figura 12(b), o qual deve ser o resultado da aplicação da rotina INOUTDEGREE para o grafo na Figura 12(a). Admita que cada elo e arco tenha custo unitário. A Figura 12(c) mostra os vértices ímpares, com os caminhos dos elos conectado-os mostrado em linhas tracejadas e os caminhos de arcos e elos orientados mostrado em linhas cheias. Neste exemplo, há pelo menos um ciclo. Se o vértice marcado v , é o primeiro vértice examinado, então os arcos e elos serão ajustados como indicados na Figura 12(c). A figura 12(d) mostra o grafo modificado.

Limitaremos, agora, o comprimento do algoritmo de EDMONDS & JOHNSON. Enquanto as rotinas EVENDEGREE e INOUTDEGREE, cada uma, modificam o grafo de entrada de modo a obter a configuração de custo mínimo, encontraremos que o algoritmo como um todo não produz necessariamente a solução ótima.

TEOREMA, FREDERICKSON (1979): *Se C^* é o custo de uma solução ótima para o PCCm, e \bar{C} é o custo da rota gerada pelo algoritmo MIXED 1, então:*

$$\bar{C}/C^* \leq 2$$

e o limite é aproximável. A complexidade do algoritmo MIXED 1 é no pior caso $O(\max\{|V|^3, |A| (\max\{|A|, |E|\})^2\})$.



Uma aproximação alternativa para o PCCm é o reverso da aproximação de EDMONDS & JOHNSON. Primeiro, o grau de entrada e o de saída são iguais. Então o grau de cada vértice é tornado par, enquanto preserva a propriedade que para cada vértice o grau de entrada é igual ao grau de saída. Daremos uma rotina que faz uso da rotina INOUTDEGREE e que também atinge o limite do pior caso de 2, para o PCCm.

Rotina LARGE CYCLES

Entrada: Saída da rotina INOUTDEGREE

Saída: Rota cobrindo todos os arcos e elos de G .

1. Identifique os vértices de grau ímpar no $G' = (V, U)$. Senão há vértices ímpares, vá para o Passo 4;
2. Encontre todos os caminhos entre os vértices sobre o grafo $G = (V, E)$;
3. Faça o algoritmo de *1-Matching* de custo mínimo dos vértices de grau ímpar usando as menores distâncias. Insira os elos usados no *matching* dentro de U .
4. Encontre uma rota que passe por todos os arcos e elos orientados em M e vértices não orientados em U .

Algoritmo MIXED 2

Entrada: Um grafo misto $G = (V, E, A)$ e a função de custo dos arcos e elos em $E \cup A$

Saída: Uma rota que percorre todos os arcos e elos em G .

1. Chame a rotina INOUTDEGREE
2. Chame a rotina LARGECYCLES

TEOREMA, FREDERICKSON (1979): Se C^* é o custo de uma rota ótima para o PCCm, e \bar{C} é o custo de uma rota gerada pelo algoritmo MIXED 2, então:

$$\bar{C}/C^* \leq 2$$

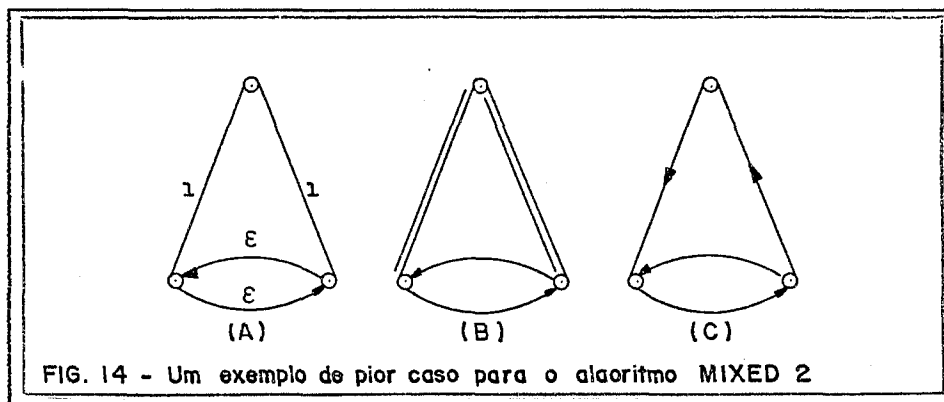
e o limite é aproximado. A complexidade computacional é a mesma do algoritmo MIXED 1.

Verificando cuidadosamente os dois piores casos, exemplo mostrados, notamos que o algoritmo Mixed 1 resolve o pior caso do algoritmo Mixed 2 e o algoritmo Mixed 2 resolve o pior caso do algoritmo Mixed 1. Tal fato, nos leva a propor um algoritmo generalizado para o PCCm de acordo com FREDERICKSON (1979).

Algoritmo MIXED-GERAL

Entrada: Um grafo $G = (V, E, A)$ com custos dos arcos e elos

Saída: Uma rota que cobre todos os elos e arcos de G .



1. Chame a rotina MIXED 1
2. Chame a rotina MIXED 2
3. Selecione a rota de menor custo

TEOREMA, FREDERICKSON (1979): *O algoritmo Mixed-Geral produz uma rota tal que:*

$$\bar{C}/C^* \leq 5/3$$

e a complexidade computacional é a mesma para o algoritmo Mixed 1.

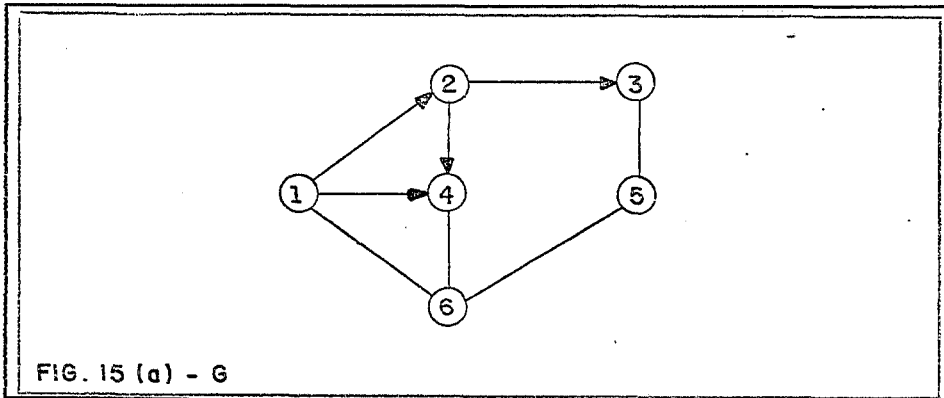
O último algoritmo aqui proposto resolve com um bom limite superior o PCCm. É bem verdade que este algoritmo pode necessitar de um esforço computacional impraticável em grandes redes viárias, apesar de sua complexidade assintótica ser polinomial. Encontrar um algoritmo que possa apresentar resultados mais rápidos, nos parece uma tarefa bastante interessante para os pesquisadores da área, haja vista termos poucas heurísticas que solucionem o problema produzindo soluções viáveis boas. O trabalho de MINIEKA (1979), utilizando direcionamento com inclusões de nós artificiais de travessia, e resolvendo-o através do algoritmo de fluxo de custo mínimo com ganhos, é outra heurística por nós estudada, mas que não foi aqui explorada pois o trabalho de Minieka não determina limites para a sua heurística, além da pouca facilidade de compreensão dos passos do algoritmo.

Enunciamos aqui todo o trabalho de Frederickson, por acharmos adequado ao uso da idéia central do propósito de nosso estudo, como também demonstrar um pouco da facilidade com que se pode implementar os seus algoritmos, devido as técnicas de programação serem mais facilmente absorvidas.

Em termos práticos, de pouco vale se a solução é ótima do ponto de vista teórico ou que é uma boa solução para uso geral, haja vista as necessidades mais evidentes de uma solução que oriente o carteiro no seu trabalho diário. O enfoque que desejamos oferecer, como se comenta em revistas como OR/MS TODAY - Publication of ORSA and TIMS, Feb.1990, que a importância maior está na busca de soluções adequadas e não exatamente a ótima.

Como ponto de partida, acreditamos que uma implementação do algoritmo Mixed 2 seja um trabalho valioso, pois poderá dar início a boas soluções para problemas como na coleta de lixo, na inspeção de linhas elétricas, na distribuição de gás butano, etc. Faremos pois, a seguir, uma exemplificação do algoritmo Mixed 2, para sua compreensão e uma posterior implementação computacional, como objeto de um novo trabalho de dissertação.

Exemplo: Seja o grafo misto $G = (V, A, E)$, da Figura 15(a)



Apesar de topologicamente incorreto, considere que todos os arcos possuem custo unitário. Determinaremos G_E de G , utilizando o algoritmo Mixed 2.

Passo 1: INOUTDEGREE - Fig.15(b)

1. $b_1 = 0 - 2 = -2$
 $b_2 = 1 - 2 = -1$
 $b_3 = 1 - 0 = 1$
 $b_4 = 2 - 0 = 2$
 $b_5 = b_6 = 0$

2. Min $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_{17} + x_{18} + x_{19} + x_{20}$
 sujeito a:

$$(x_2 + x_{10} + x_{17} + x_{20}) - (x_1 + x_9) = -2$$

$$(x_{18} + x_{19} - x_{17} = -1$$

$$(x_8 + x_{16}) - x_{18} = 1$$

$$(x_4 + x_{12}) - (x_3 + x_{11} + x_{19} + x_{20}) = 2$$

$$(x_6 + x_7 + x_{14} + x_{15}) - (x_5 + x_8 + x_{13} + x_{16}) = 0$$

$$(x_1 + x_5 + x_9 + x_{13}) - (x_2 + x_4 + x_6 + x_{10} + x_{12} + x_{14}) = 0$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{17}, x_{18}, x_{19}, x_{20} \in \mathcal{Z}_+$$

$$x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16} \in \{0, 1\}$$

Solução do Modelo

i	1	2	3	4	5	6	7	8	9	10
x()	2	0	0	1	0	0	0	0	1	0
i	11	12	13	14	15	16	17	18	19	20
x()	0	1	0	1	0	1	1	0	0	0

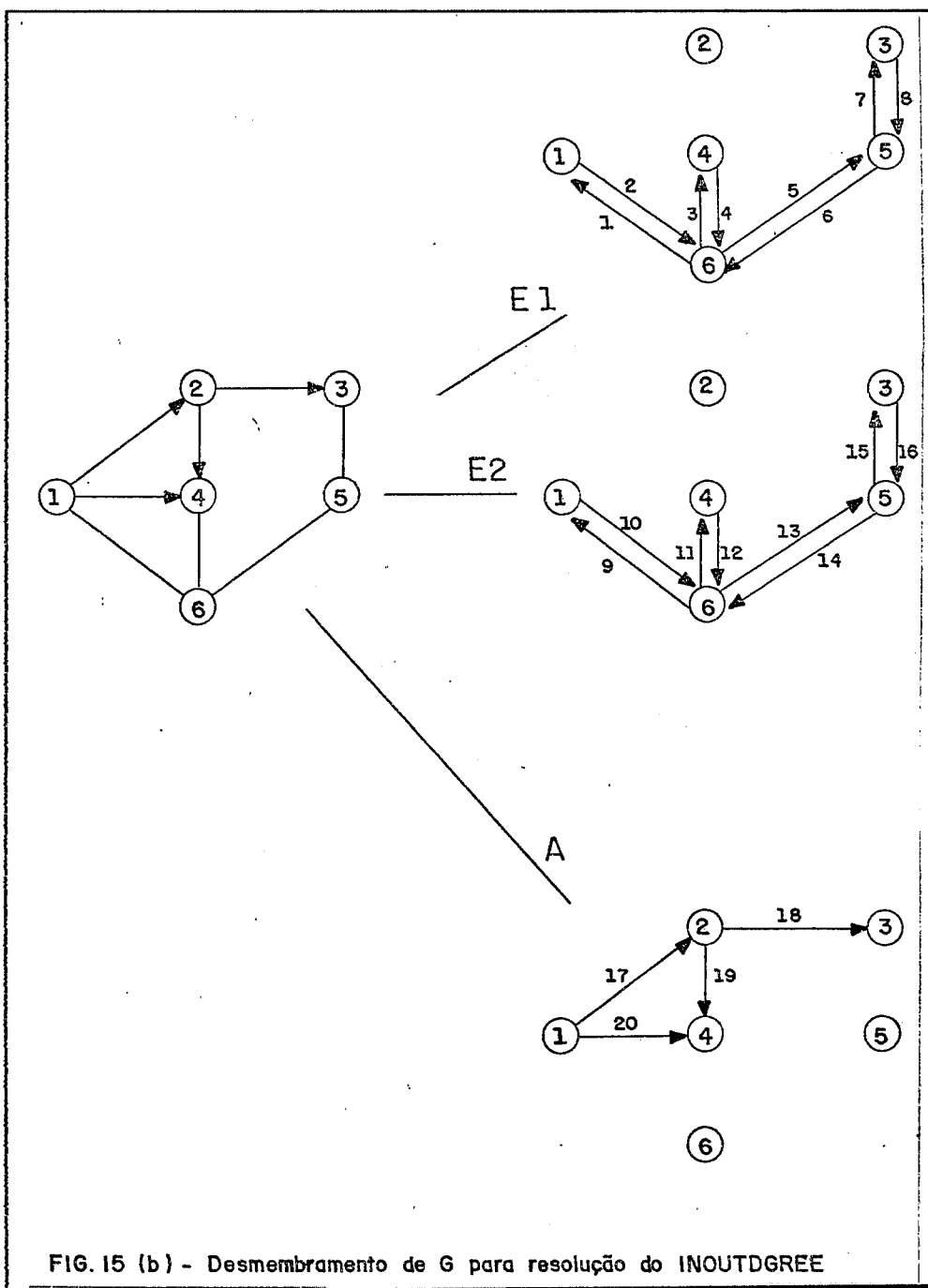


FIG. 15 (b) - Desmembramento de G para resolução do INOUTDGREE

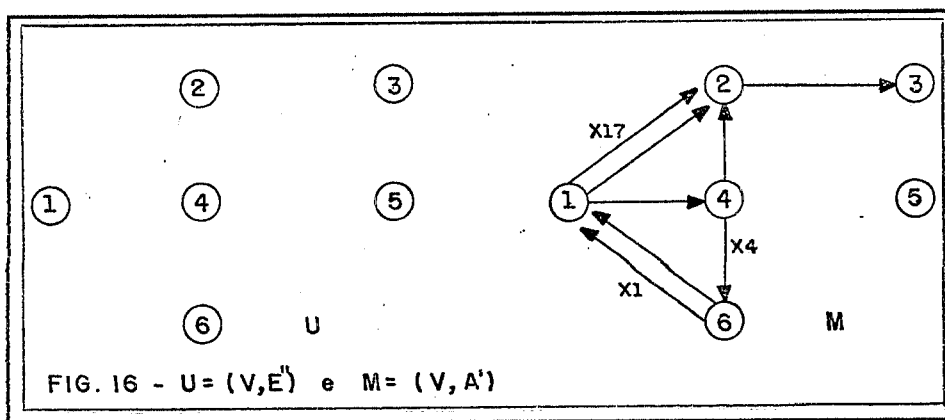
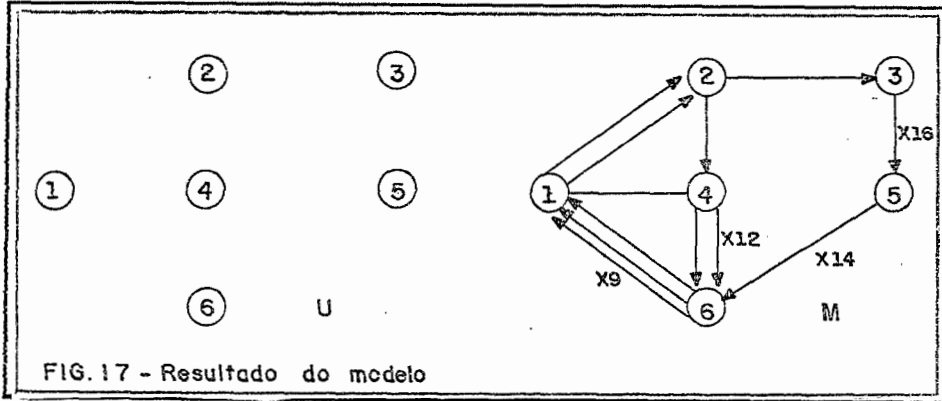


FIG. 16 - $U = (V, E')$ e $M = (V, A')$

3.

$$\begin{aligned}
 x_9 + x_{10} = 1 &\Rightarrow \text{copie } x_9 \text{ em } M \\
 x_{11} + x_{12} = 1 &\Rightarrow \text{copie } x_{12} \text{ em } M \\
 x_{13} + x_{14} = 1 &\Rightarrow \text{copie } x_{14} \text{ em } M \\
 x_{15} + x_{16} = 1 &\Rightarrow \text{copie } x_{16} \text{ em } M
 \end{aligned}$$



Passo 2: LARGE CYCLES

1. β vértice de grau ímpar em U
4. O grafo $G_E \equiv M$

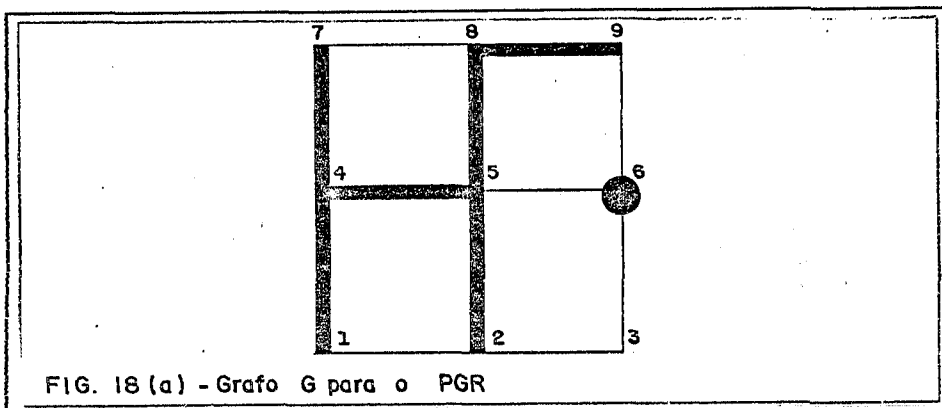
II.6 O Problema de Percurso Capacitado em Arcos (PPCA)

Após a proposição do PCC, MEI-KO KWAN (1962), onze anos mais tarde, quase que simultaneamente, EDMONDS & JOHNSON (1973) e CHRISTOFIDES (1973) publicaram artigos que resolvem o PCC. No caso do artigo de CHRISTOFIDES (1973), além da solução do PCC, ele propôs uma heurística para resolver o problema dos caminhões removedores de neve, na inspeção de linhas elétricas, de gás e na entrega de cartas. Nestas aplicações, o professor Christofides considerou que cada elo da rede viária, além do custo em distância, também possuía um outro custo, que para o caso da remoção de neve, era a quantidade de sal, estimada por segmento de rua, que cada veículo removedor teria que aplicar em sua passagem.

BODIN & GOLDEN (1981), classificaram o problema da remoção de neve, como o Problema do Carteiro Chinês Capacitado (PCCC). No PCCC, como podemos notar, toda a rede solicita serviço. Assim, um conjunto mínimo de carteiros ou removedores, devem sair para realizar o trabalho, de modo que a demanda total seja satisfeita.

Com o trabalho de GOLDEN & WONG (1981), pensou-se em resolver um problema bem mais geral que o PCCC. O Problema de Percurso Capacitado em Arcos (PPCA), como foi chamado, admite que não necessariamente todos os elos da rede viária precisam de serviço, e que a frota pode ser heterogênea, e sediada em um único depósito ou garagem.

Em seu trabalho, GOLDEN & WONG (1981) mostraram que o PCC, PCR, PCCC e o PGR, são todos subproblemas do PPCA. No caso do PGR, como há nós com demanda, a modificação de um nó em um arco, como proposta por ORLOFF (1974a), poderá ser feita de modo que não haja modificação em sua estrutura (Ver Fig. 18(a) e Fig. 18(b))



O modelo matemático para o PPCA pode ser assim descrito:

$$(PPCA) \quad \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^n \sum_{p=1}^L c_{ij} x_{ij}^p \quad (II.18)$$

sujeito a:

$$\sum_{k=1}^n x_{ki}^p - \sum_{k=1}^n x_{ik}^p = 0 \quad i = (1, \dots, n), p = (1, \dots, L) \quad (\text{II.19})$$

$$\sum_{p=1}^L (l_{ij}^p + l_{ji}^p) \leq 1 \quad (i, j) \in E \quad (\text{II.20})$$

$$x_{ij}^p \geq l_{ij}^p \quad (i, j) \in E, p = (1, \dots, L) \quad (\text{II.21})$$

$$\sum_{i=1}^n \sum_{j=1}^n l_{ij}^p q_{ij} \geq W(p) \quad p = (1, \dots, L) \quad (\text{II.22})$$

$$\sum_{i \in \hat{Q}} \sum_{j \in \hat{Q}} x_{ij}^p - n^2 y_{1\hat{Q}}^p \leq |\hat{Q}| - 1 \quad (\text{II.23})$$

$$\sum_{i \in \hat{Q}} \sum_{j \notin \hat{Q}} x_{ij}^p + y_{2\hat{Q}}^p \geq 1 \quad (\text{II.24})$$

$$y_{1\hat{Q}}^p + y_{2\hat{Q}}^p \leq 1, \quad y_{1\hat{Q}}^p, y_{2\hat{Q}}^p \in \{0, 1\} \quad (\text{II.25})$$

$$x_{ij}^p, l_{ij}^p \in \{0, 1\} \quad (\text{II.26})$$

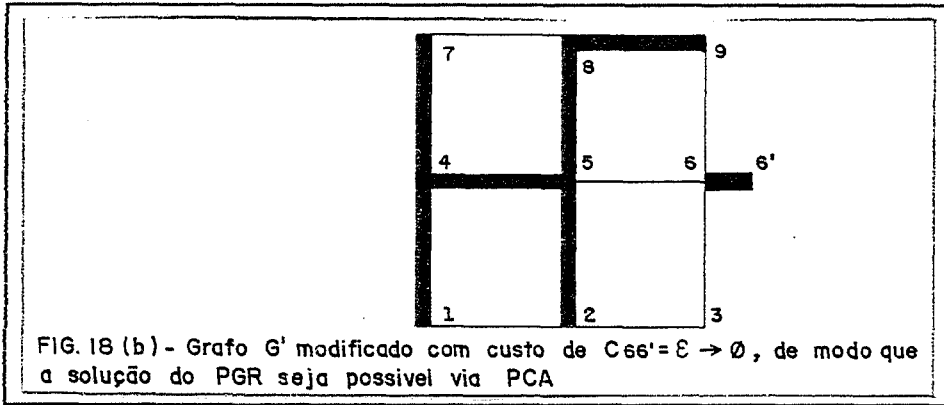
para $p = (1, \dots, L)$, $\hat{q} = (1, \dots, 2^{(n-1)} - 1)$ e todo subconjunto não vazio \hat{Q} de $\{2, \dots, n\}$

Onde,

n - Número de nós;

L - Número de veículos disponíveis;

$W(p)$ - Carga do veículo p , $W(p) \geq \max\{q_{ij}\}$, $\forall p$



c_{ij} - O comprimento do elo (i,j)

$$x_{ij}^p = \begin{cases} 1, & \text{se o elo } (i,j) \text{ é descrito pelo veículo } p; \\ 0, & \text{caso contrário} \end{cases}$$

$$l_{ij}^p = \begin{cases} 1, & \text{se o elo } (i,j) \text{ é servido pelo veículo } p; \\ 0, & \text{caso contrário} \end{cases}$$

A função objetivo procura a distância total mínima. A restrição (II.19) permite a continuidade da rota. A restrição (II.20) garante que o elo (i,j) pode ser servido pelo veículo p apenas se ele atravessar o elo (i,j) . A capacidade do veículo não pode ser violada e as restrições de (II.23) a (II.25) impede a formação de subrotas desconexas à rota do veículo p .

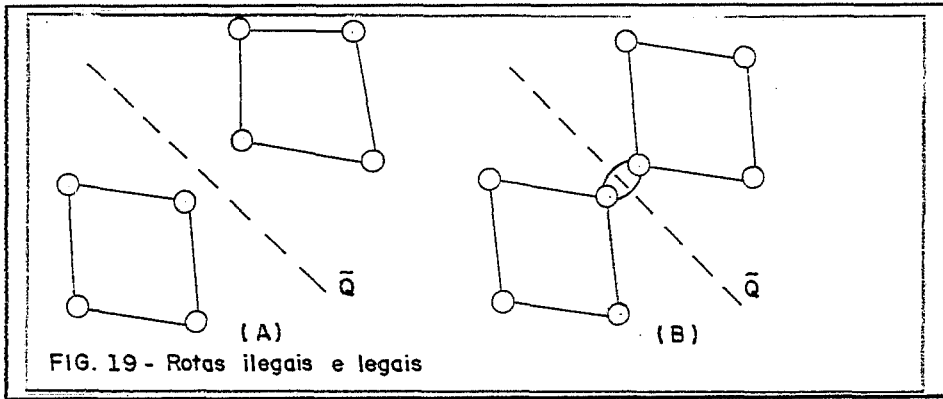
Como vemos na figura, abaixo, caso as restrições de (II.23) a (II.25) não fossem colocadas, a situação seria possível em (a), formando rotas ilegais a um mesmo veículo p . Nas restrições (II.23) e (II.24), desejamos prevenir a formação mas permite-se um outro tipo de configuração. Toma-se este cuidado, desde que:

$$\sum_{i \in \hat{Q}} \sum_{j \notin \hat{Q}} x_{ij}^p > \hat{Q} - 1 \Rightarrow y_{1\hat{Q}}^p = 1, y_{2\hat{Q}}^p = 0 \Rightarrow \sum_{i \in \hat{Q}} \sum_{j \notin \hat{Q}} x_{ij}^p \geq 1$$

A formulação matemática para o PPCA é bastante importante por diversas razões. Dentre elas, o modelo permite uma estrutura unificada para que se visualize problemas de otimização combinatória correlatos. Uma outra razão, é que a estrutura matemática do modelo e a complexidade do problema de programação linear-inteira torna-se visível. Como vemos, a formulação consiste de $3L(2^{(n-1)} - 1)$ desigualdades.

O conjunto de restrições de (II.23) a (II.25) podem ser modificados para a seguinte formulação:

$$\sum_{k=1}^n f_{ik}^p - \sum_{k=1}^n f_{ki}^p = \sum_{j=1}^n l_{ij}^p \quad (i = 2, \dots, n), \quad (p = 1, \dots, L) \quad (\text{II.27})$$



$$f_{ij}^p \leq n^2 x_{ij}^p \quad (i, j) \in E, \quad (p = 1, \dots, L) \quad (\text{II.28})$$

$$f_{ij}^p \geq 0 \quad (\text{II.29})$$

Note que f_{ij}^p é uma variável de fluxo a qual pode admitir valores positivos apenas se $x_{ij}^p = 1$. As novas restrições mantêm a formulação de subrotas ilegais. No entanto, o número de restrições na nova formulação não é menor que $3Ln^2$, o que parece ser extremamente atraente quando temos uma rede viária consideravelmente grande.

O limite inferior para o PPCA pode ser definido como $\bar{C}/C^* \leq 1.5$, em colorário proposto por GOLDEN & WONG (1981).

Devido ser um problema do tipo \mathcal{NP} -Hard, algumas heurísticas foram propostas para a solução do PPCA. Dentre elas podemos citar o algoritmo de GOLDEN & WRONG (1981), que utiliza uma metodologia muito próxima à de CLARKE & WRITE (1964) para nós, ou seja, é uma heurística baseada em economias de ligações entre arcos; a heurística que usa transformação de elos em nós, PEARN ET AL (1987), que se utiliza de um procedimento de quebra dos elos em quatro outros, de modo que se possa utilizar uma heurística apropriada do PPV em nós. O objetivo do trabalho de PEARN ET AL (1987), é estabelecer uma equivalência entre os problemas de roteamento em nós e em arcos.

Outras heurísticas como: *Matching Lower Bound (MLB)* e *Node Scanning Lower Bound (NSLB)*, ASSAD ET AL (1987); método de inserção paralela, CHAPLEAU ET AL (1984); algoritmo modificado de construção - quebra, PEARN (1989); o Problema Capacitado do Carteiro Canadense, SERGE ROY & J.M. ROUSSEAU (1988), e o método de determinação de frotas para o PPCA, ULUSOY (1985), podem ser vistas na literatura como grandes trabalhos na área.

Haja vista termos nos preocupado primordialmente com o PCC, e o nosso desejo de estudar um algoritmo eficiente para uma aplicação geral, no roteamento em arcos, achamos no trabalho de ULUSOY (1985) a maior proximidade

ao que até aqui desenvolvemos. A seguir, então, procederemos um estudo minucioso desta heurística, do tipo rotear primeiro e agrupar depois, bastante atraente do ponto de vista prático.

II.6.1 O Trabalho de Ulusoy

A literatura tem feito algumas tentativas de resolver o PPCA com um conjunto de m -veículos, iniciando suas rotas de um depósito central único, com o objetivo de minimizar a distância total das viagens, ou o tempo destas, conforme foi visto.

No trabalho que aqui estudamos, também possuímos custos fixos associados aos veículos, incluídos na função objetivo. O procedimento de solução consiste em quatro fases. Na primeira, um PCC ou PCR é resolvido, dependendo se todos ou alguns arcos da rede viária demandam serviço, com o objetivo de minimizar a distância total percorrida. Isto resulta em uma viagem chamada *Grande Rota*. Na segunda, a *Grande Rota* é particionada em subrotas de um veículo, viáveis com respeito às restrições. Uma nova rede é construída com o conjunto de arcos da *Grande Rota*, e com o conjunto de arcos consistindo das subrotas da *Grande Rota*. Os custos dos arcos incluem os custos fixos das variáveis das subrotas. A terceira fase resolve o problema de caminhos mínimos nesta nova rede. Na última fase, um pós-processamento é aplicado à solução anterior para melhorá-la. O procedimento é repetido para diferentes *Grandes Rotas*, pretendendo, melhorar a solução final. O problema se estende ao caso onde pode ocorrer limites superiores no número de veículos necessários com determinada capacidade, usando um método *Branch-and-Bound*.

Devido à extensão do algoritmo e os propósitos de sua aplicação, procuraremos apenas detalhar as três primeiras partes. Deixamos para os que vierem, a continuidade do estudo deste magnífico trabalho.

Antes, porém, de iniciarmos a descrição do algoritmo, devemos colocar que neste trabalho o tamanho da frota e sua heterogeneidade são variáveis de decisão, quando nos estudos, até agora realizados, a frota é homogênea, GOLDEN & WONG (1981), CHAPLEAU ET AL (1983), ASSAD ET AL (1987), PEARN (1989). Para determinar o tamanho da frota e sua formação, devemos considerar o custo fixo dos veículos juntamente aos custos de operação destes. Os custos de operação, são tomados como uma função linear da distância percorrida. O objetivo é, pois, minimizar o custo total o qual é a soma dos custos fixos e variáveis.

Acerca das condições de operação impostas a este problema, temos:

1. Há um depósito único onde toda a frota está alocada inicialmente;
2. Os veículos retornam ao depósito após o término de seu trabalho;
3. Todas as atividades são realizadas nos arcos e no depósito. Em outras palavras, não há aterros ou locais de despejo como no caso da coleta do lixo urbano;
4. Cada veículo tem um limite de operação máximo, ou operam num máximo espaço de tempo;
5. Todas as demandas devem ser satisfeitas;
6. Não há divisão de serviço em arcos com demanda entre veículos. Nos casos em que a demanda dos arcos são maiores que a capacidade do

veículo de maior capacidade, então a demanda é feita por tantos quanto sejam os veículos totalmente carregados, apenas percorrendo estes arcos até que a demanda de serviço reminescente seja menor que a capacidade do maior veículo. O veículo mais carregado selecionado, é aquele que possui menor custo unitário;

7. Um veículo com a menor capacidade encontrando a demanda da rota de um outro veículo, será sempre capaz de ser designado àquela rota.

As fases do procedimento de solução podem ser assim descritas:

Passo 1: Encontre a *Grande Rota*

Passo 2: Particione a *Grande Rota* em sub-rotas, as quais correspondem a uma rota viável para o veículo

Passo 3: Selecione o conjunto de menor custo, satisfazendo as demandas do conjunto de veículos gerados

Passo 4: Repita os passos acima até que algum critério de parada seja satisfeito.

1. Criando a *Grande Rota*: Aplique o modelo apropriado de roteamento em arcos levando em consideração o grafo existente (não- direcionado, direcionado ou misto);
2. Particionando a *Grande Rota*: O particionamento da grande rota é de acordo com uma transformação na rede $G \rightarrow G^*$. G^* é definido por seu:

- (a) Conjunto de nós;
- (b) Conjunto de arcos;
- (c) Relações de incidência e
- (d) Estrutura de custo dos arcos.

Reduziremos inicialmente a rede $G_E \dots$

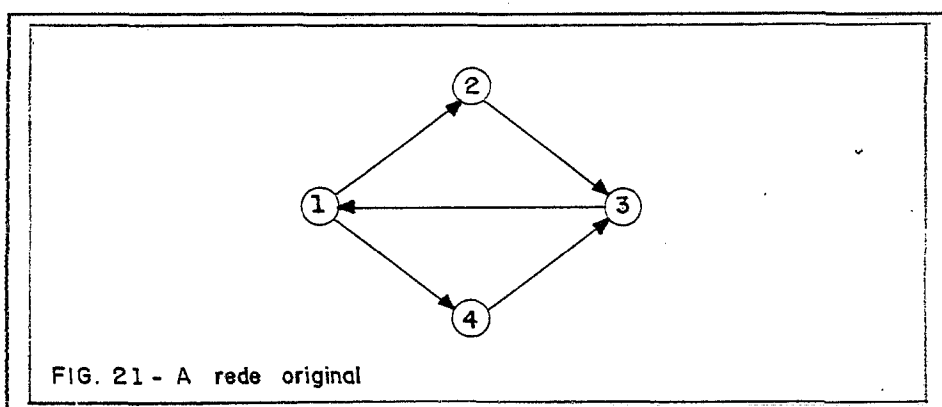
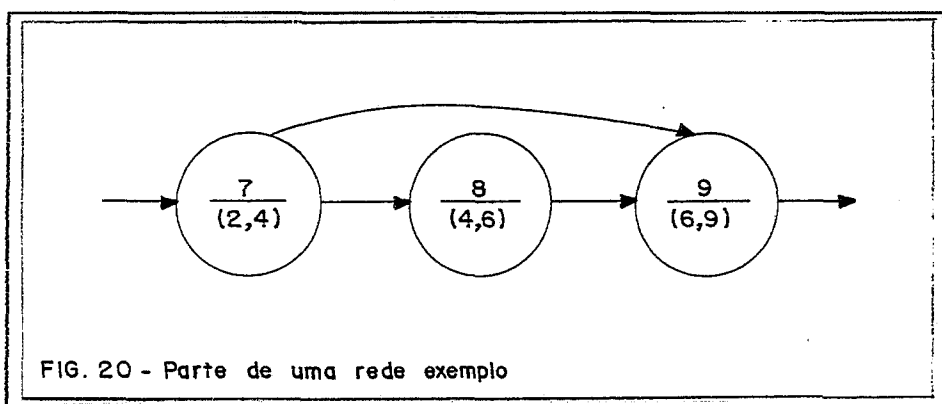
Regra 1: Se há alguma cadeia de arcos desnecessária ao serviço na rota de Euler, então ele é reduzido a um arco com nós iniciais e finais sendo aqueles da cadeia, e o seu custo será o custo total da cadeia.

A regra seguinte, determina o conjunto de nós da rede:

Regra 2: Cada nó de G^* corresponde a um caso na rota, exceto o primeiro nó. Este, em G^* , corresponde ao nó depósito (nó 1) de G . Os arcos na rota de Euler são numerados na mesma ordem que aparecem na rota e são mapeados

a nós de G^* na mesma ordem, com o primeiro arco correspondendo ao nó (2) em G^* . Se o último nó de G^* representa um arco não-servido, então ele é eliminado do conjunto de nós.

Cada arco de G^* corresponde a uma rota viável de um veículo em G^* . Por exemplo, um arco (k, h) em G^* corresponde à rota do veículo em G a qual inclui todos os arcos representados pelos nós $(k+1), \dots, h$. Na Figura 20, uma parte de uma rede exemplo é mostrada. O número do nó em G^* é dado na metade superior de cada nó e no arco correspondente na rota do veículo em G , é dada na metade inferior. De acordo com a convenção aqui adotada, o arco $(7,8)$ em G^* corresponde a uma rota de veículo em G servindo apenas o arco $(4,6)$. De maneira análoga, o arco $(7,9)$ em G^* corresponde a uma rota de um veículo em G , servindo apenas os arcos $(4,6)$ e $(6,9)$. Como uma rota de veículo servindo estes arcos é formada, será mostrado detalhadamente mais adiante.



Formação de conjuntos de arcos ...

Regra 3: Um arco é incidente a cada nó h dentro de cada nó k , com $h < k$, dado que a rota do veículo correspondente não viola quaisquer restrições laterais que devem estar presentes.

A presença de arcos sem demanda em G é explorada para reduzir o número de arcos em G^* , como determinado pela regra 3. Faça o nó i corresponder a um arco sem demanda em G .

Regra 4: Nenhum arco é incidente ao nó i exceto o arco $(i-1, i)$;

Regra 5: Nenhum arco é incidente ao do nó $(i - 1)$, exceto o arco $(i - 1, i)$.

Estas duas últimas regras, foram formuladas de modo a prevenir que arcos sem demanda estejam presentes nas rotas dos veículos sem propósito, e assim levando a rotas inferiores.

Quando particionamos uma rota de Euler, obtivemos a sequência de arcos da rota de Euler, as quais foram transformadas em rotas dos veículos. Para este propósito distinguimos dois casos:

Caso 1: Os arcos na sequência formam um ciclo;

Caso 2: Os arcos na sequência não formam um ciclo.

Caso 1: Se os arcos na sequência formam um ciclo, então:

(S_1) Se o nó 1 é um membro da sequência, então a sequência corresponde à rota do veículo;

(S_2) Se o nó 1 não é um membro da sequência, então conecte o nó 1 do ciclo, com a cadeia de menor custo.

Caso 2: Se os arcos na sequência não formam um ciclo, então:

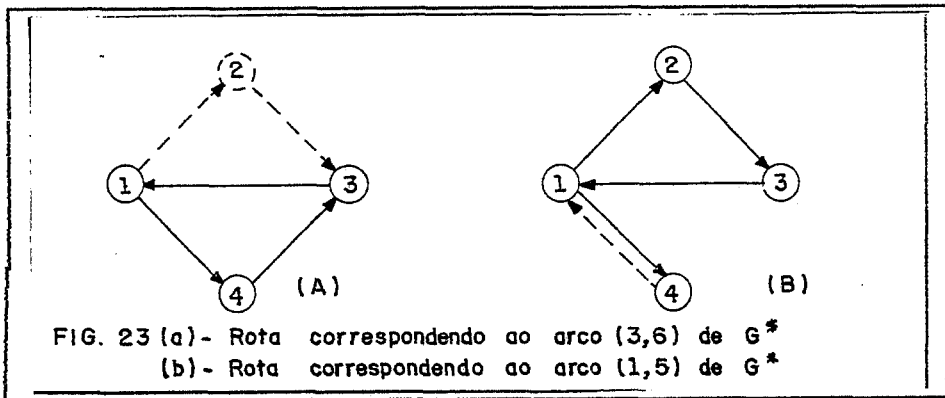
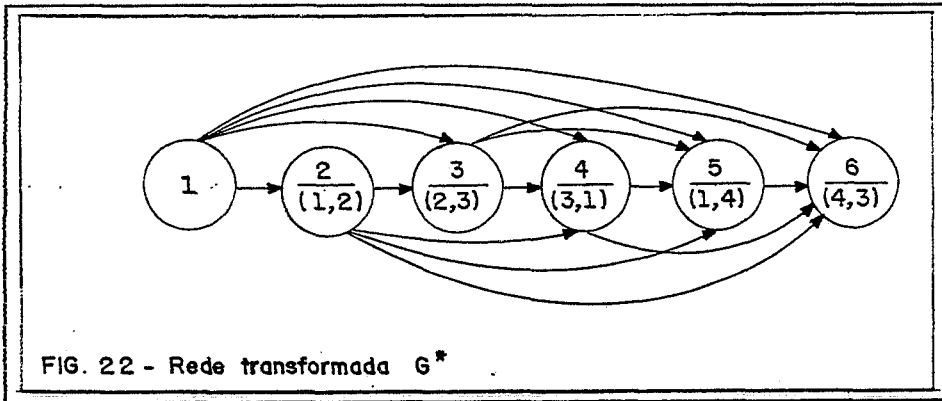
(S_3) Se o nó 1 é um membro da sequência, então conecte o nó 1 a dois nós de grau ímpar usando a cadeia de menor custo.

(S_4) Se o nó 1 não é um membro da sequência, então conecte o nó a dois nós de grau ímpar, usando a cadeia de menor custo.

Vejamos um exemplo da construção de G^* , e como um arco em G^* representa uma rota de veículo em $G \dots$

Considere a rede da Figura 21, na qual cada arco possui demanda positiva para serviço. Assim o PCC resolveria o problema da *Grande Rota*. Tal rota pode ser dada pela seguinte rota de Euler: $\{1,2,3,1,4,3,1\}$. Note que o arco $(3,1)$ é usado duas vezes, escolhendo o segundo como o arco sem demanda, reduzimos o tamanho de G^* de um nó, de acordo com a regra 2. Temos, pois, 6(seis) nós em G^* , mostrados na Figura 22. Todos os arcos resultantes da regra 5 estão incluídos em G^* , desde que nenhuma restrição de capacidade seja violada.

Verificando a rota do veículo, temos: considere o arco $(3,6)$ em G^* . A rota do veículo correspondente inclui os arcos representados pelos nós 4,5 e 6 em G^* . Usando a regra (S_1) temos a rota do veículo na Figura 23(a). A rota do veículo correspondente ao arco $(1,5)$ possui os arcos representados pelos nós 2, 3, 4 e 5 em G^* e é dada na figura 23(b). Isto é obtido usando a regra (S_3) como um resultado do qual o arco $(4,1)$ é incluído para completar a rota.



A construção de G^* estará completa, uma vez que atribuíamos o custo dos arcos \bar{c}_{ij} para cada arco (i,j) em G^* . O custo dos arcos são determinados de acordo com as seguintes regras:

- C1: Dado o nó $(k+1)$ em G^* , é um arco sem demanda (isto é, ele representa um arco sem demanda de *Grande Rota*), então $\bar{c}_{kk+1} = 0$.
- C2: \bar{c}_{kk} é a soma do custo dos arcos c_{ij} incluídos na rota do veículo, e os custos fixos do veículo de menor capacidade, juntando-se à demanda da rota do veículo correspondente.

Selecionando o conjunto de rotas de menor custo em G^* ...

Tendo construído G^* , estamos prontos para selecionar o conjunto de rotas de menor custo representadas em G^* . Isto pode ser acompanhado pela solução do caminho mais curto do nó (1) ao nó (m) , onde o nó (m) corresponde ao último arco com demanda de serviço positiva na *Grande Rota*. O caminho mínimo é obtido pela aplicação do algoritmo de DIJKSTRA (1959). A estrutura acíclica de G^* melhora a eficiência do algoritmo. As rotas correspondentes às rotas do caminho mínimo, formam o conjunto de rotas de menor custo em G^* .

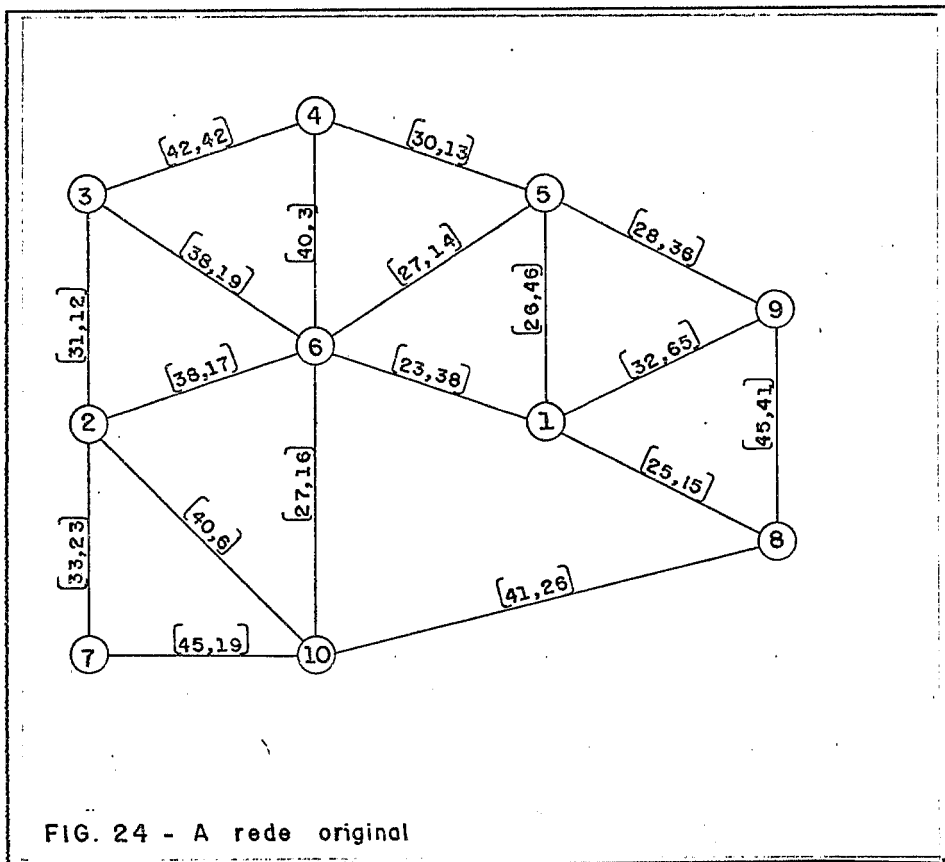
Critério de parada ...

Quando encontramos a menor rota G^* , teremos como resultado o conjunto de rotas de menor custo para aquele G^* particular, a qual é obtida para

uma dada rota euleriana em G_E . Caso construamos diferentes circuitos eulerianos de um mesmo grafo G_E , teremos, em geral, diferentes G^* . Para que não tenhamos que gerar uma infinidade de rotas para cada circuito euleriano possível, sugerimos utilizar o algoritmo de BODIN & KURSH (1979) na determinação de uma rota euleriana que minimiza percursos *desagradáveis*, como já foi discutido. Com isso geramos apenas um grafo G^* no algoritmo básico de Ulusoy.

Um problema exemplo ...

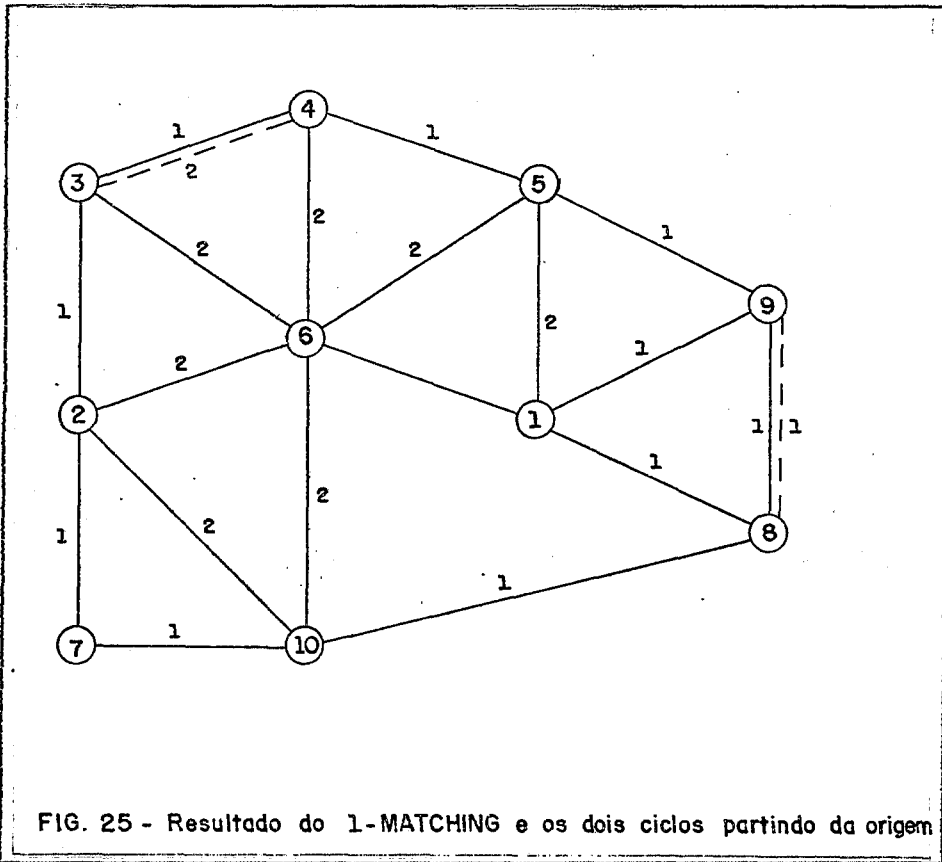
Considere a rede da Figura 24, onde os valores entre colchetes - $[a, b]$, representam: a , o custo do elo e b a demanda deste. Temos três tipos de veículos com capacidades 50, 80 e 100. Os seus custos fixos podem ser tomados de acordo com suas capacidades.



Resolvendo, a Fig. 25 mostra o grafo euleriano de G .

Adotando os critérios de BODIN & KURSH (1979),

- Segue em frente = 0
- Manobra à direita = 1
- Manobra à esquerda = 4
- Retorno em U = 8



$$\bullet \begin{cases} \text{Não Serve} - \text{Serve} \\ \text{Serve} - \text{Não-serve} \end{cases} = 5$$

encontramos, utilizando o algoritmo de BODIN & KURSH (1979), a seguinte rota:

1o. Ciclo: $1 \leftrightarrow 8 \leftrightarrow 10 \leftrightarrow 7 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5 \leftrightarrow 9 \leftrightarrow \underline{10} \leftrightarrow \underline{9} \leftrightarrow 1$

1 retorno em U
1 manobra à esquerda

2o. Ciclo $1 \leftrightarrow 6 \leftrightarrow \underline{3} \leftrightarrow \underline{4} \leftrightarrow 6 \leftrightarrow 10 \leftrightarrow 2 \leftrightarrow 6 \leftrightarrow 5 \leftrightarrow 1$

1 manobra à esquerda

O esforço total para a rota euleriana é de 30 unidades.

Os arcos sublinhados em cada um dos ciclos acima, representam arcos que não serão servidos, ou arcos de apenas passagem.

Matriz de Custos:

Exemplo: Desejamos achar o custo fixo dos arcos \bar{c}_{ij} , em especial do arco (12,14).

Arcos servindos: (1,6), (6,3)

Carga Servida: $38 + 19 = 57$ (veículo adequado: 80)

Veículo de menor carga que pode servi-lo: 80, com custo fixo 80

Rota do veículo: $\{1, 6, 3, 6, 1\} = c_{12,14} = (23 + 38 + 38 + 23) + 80 = 202$

MATRIZ DE CUSTOS DOS ARCOS
(CUSTO/PESO)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	-	100	166	285	305	336	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	15	-	166	255	285	336	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	41	26	-	239	239	336	393	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	60	45	19	-	238	269	336	356	-	-	-	-	-	-	-	-	-	-	-	-	-
5	83	68	42	23	-	180	247	247	-	-	-	-	-	-	-	-	-	-	-	-	-
6	95	80	54	35	12	-	209	239	293	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	96	77	54	42	-	162	196	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	90	67	55	13	-	136	204	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	91	49	36	-	152	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	77	41	-	-	114	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	65	-	-	96	202	-	246	300	371	371	-	-
13	-	-	-	-	-	-	-	-	-	-	-	38	-	172	-	216	270	321	351	401	-
14	-	-	-	-	-	-	-	-	-	-	-	57	19	-	-	169	223	274	274	354	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	60	22	3	-	-	150	201	201	261	281
17	-	-	-	-	-	-	-	-	-	-	-	76	58	19	-	16	-	201	201	231	281
18	-	-	-	-	-	-	-	-	-	-	-	82	44	25	-	22	6	-	172	202	232
19	-	-	-	-	-	-	-	-	-	-	-	99	61	42	-	39	23	17	-	126	156
20	-	-	-	-	-	-	-	-	-	-	-	-	75	56	-	53	37	31	14	-	102
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	99	83	77	60	46	-

Na matriz de custos acima, como vemos, omitimos arcos que geram rotas conjuntas de acordo com G^* . Este fato reduz o tamanho da rede, a memória de armazenamento e a velocidade de convergência do algoritmo de DIJKSTRA (1959). É bom observar, que esta medida foi tomada devido ao fato de não haver nenhum arco que exceda a capacidade do menor veículo, necessário na realização das tarefas.

Um exemplo de missão conjunta seria no caso da rota do arco (1, 21) \equiv $\{1, 5, 4, 6, 10, 2, 6, 5, 1\}$ com custo total 391 e cargas de 102 unidades, onde teriam que ser utilizados os veículos de 50 e 80 unidades de carga.

Da matriz de custos mostrada, podemos então observar que o caminho de 1 a 21 em G^* , possui a seguinte sequência de visitação:

$$\{(1) \rightarrow (5) \rightarrow (8) \rightarrow (10) \rightarrow (12) \rightarrow (16) \rightarrow (21)\}$$

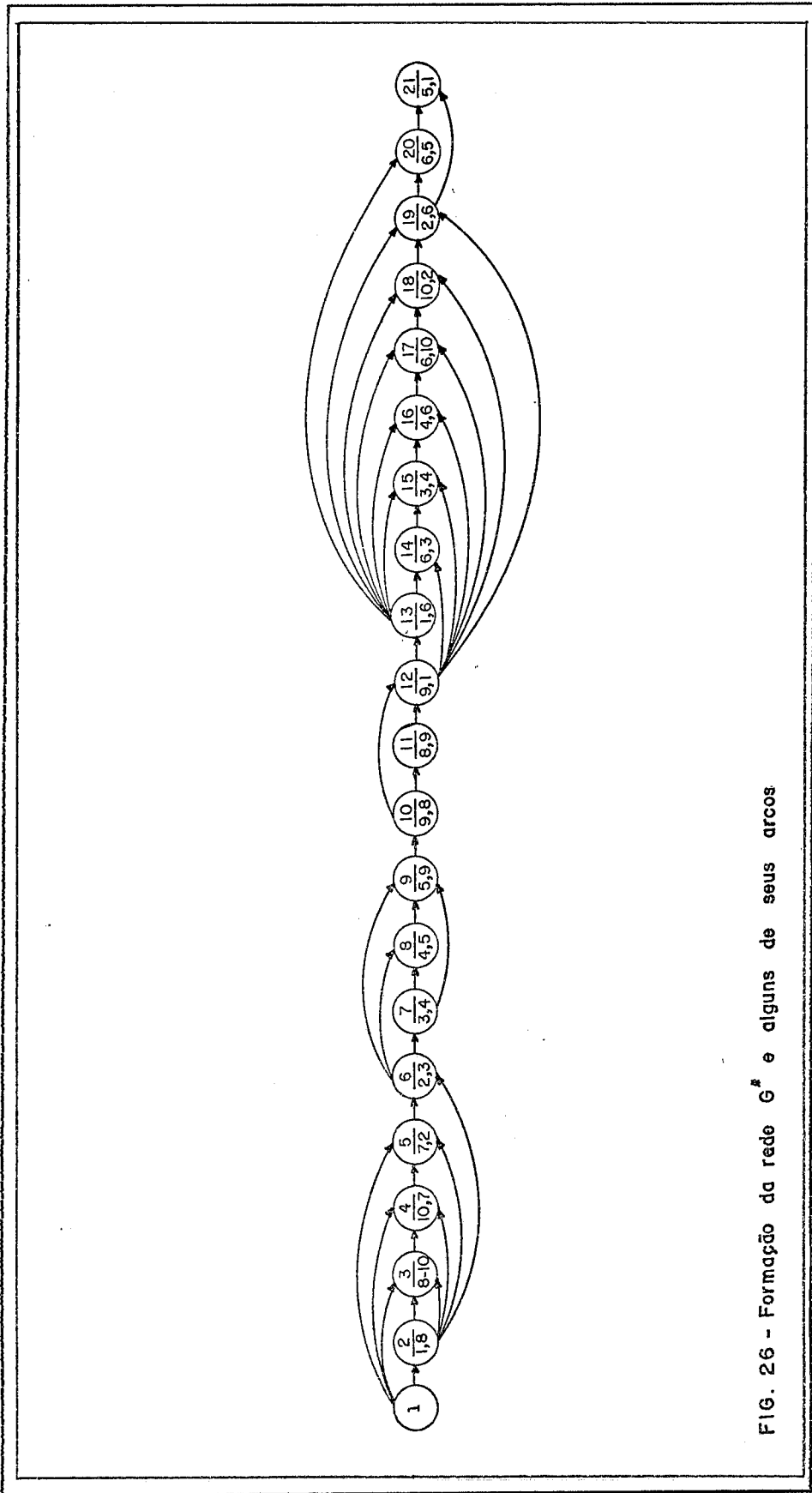


FIG. 26 - Formação da rede G^k e alguns de seus arcos

Rota	Custo	Carga	Veículo
{1,8,10,7,2,6,1}	305	83	100
{1,6,2,3,4,5,1}	247	67	80
{1,5,9,8,1}	204	77	80
{1,9,1}	114	65	80
{1,6,3,4,6,1}	246	60	80
{1,6,10,2,6,5,1}	281	99	100
Total:	1397	451	

II.7 Conclusões

Se me permitem o júbilo, todas as abordagens que apresentamos até aqui, são de extrema clarividência. Como não poderia deixar de ser, os trabalhos de FREDERICKSON (1979) e de ULUSOY (1985) coroam fortemente qualquer estudo de percurso em arcos.

Uma boa tarefa para um mestrando na área de roteamento, é realizar a implementação computacional destes dois algoritmos, pois, além da boa noção de programação inteira, poderia fortalecer bem as técnicas aqui usadas, tornando-as mais eficientes, caso seja possível.

Em cidades em que se deseja obter uma determinação apropriada do uso adequado de uma frota, dimensionando-se o seu tamanho e operacionalizando-se a sua funcionalidade, é evidente que o trabalho de Ulusoy é extremamente aplicável, apesar de inadequado à coleta do lixo, as idéias aqui apresentadas podem ser aplicáveis com sucesso na distribuição de gás butano, no roteamento dos leituristas de energia elétrica, nos removedores de neve, nos varredores rua, etc.

Nossa intenção, no entanto, é permitir que se tenha consciência da importância de uma análise global do ponto de vista prático. Quando transformamos um grafo qualquer em um grafo euleriano, temos a certeza de que poderemos gerar rotas ótimas ou próximas do ótimo em função das distâncias percorridas nesse grafo. Tal fato é relevante nas aplicações rotear-primeiro, agrupar-depois, pois amarram os limites inteiros em valores bastante atraentes.

Procuremos, por fim, mostrar de forma simplificada o problema de percurso em arcos, e acreditamos que atingimos nossos objetivos. No próximo capítulo, desenvolveremos um estudo detalhado de Planejamento e Roteamento, aplicado especificamente à coleta do lixo urbano domiciliar.

Capítulo III

O Problema de Planejamento da Frota (PPF)

III.1 Introdução

Na Coleta de Lixo Urbano, a situação dos veículos disponíveis na realização do trabalho é uma variável importante, tanto na manutenção do serviço, quanto na preocupação de manter uma maior longevidade da frota. Quando a Prefeitura contrata o serviço de uma empreiteira, ela já define a sua área de atuação (Sub-Zonas ou Trechos de coleta), e a frequência de serviço. Na tentativa de cumprir a promessa contratual, as empresas *se viram como podem* para realizar suas tarefas.

Como geralmente as empresas possuem suas próprias oficinas mecânicas, e normalmente as atividades possuem um controle pouco eficaz, em função das quebras ocasionais, faltas de peças no mercado e atraso de entregas, etc. nem sempre pode-se contar com a saída dos veículos para definir uma disponibilidade de operação. Pelo que constatamos em campo — NEGREIROS ET. AL (1990), o planejamento das atividades de coleta, atualmente, é realizado quase que *intuitivamente*. Ou seja, o gerente observa quais são os veículos que estão no pátio e, em seguida os atribui aleatoriamente aos trechos de operação, procurando fixar os motoristas aos trechos e veículos que já trabalham usualmente. Como podemos notar, esta prática pode ser eficiente apenas no controle do indivíduo que opera a máquina, apesar deste já conhecer bem o trecho que opera.

MAGNANTI (1981), definiu três abordagens para o Problema de Planejamento de Veículos. Dentre elas, a Operação de Planejamento vincula-se às condições da frota e ao número de veículos disponíveis para a realização do trabalho. No processo **Definição da Disponibilidade**, leva-se em conta o estado do veículo, o tempo corrido de serviço e um sequenciamento de atividades entre os diversos tipos existentes, haja vista a necessidade de manutenções preventivas de tempos em tempos especificados pelo fabricante do equipamento. Com esta idéia, então, partimos para a conceituação do Problema de Planejamento de Veículos, identificando um modelo que se adequasse à atividade, sem, no entanto, afetar o fato da designação

dos motoristas que conhecem as suas áreas de serviço, haja vista a manutenção da eficiência do processo hora realizado.

Em nossa pesquisa bibliográfica, não identificamos um modelo que se aproximasse à situação de operação e atribuição de veículos, conforme a sua utilização ideal.

Sendo de extrema importância a decisão de designação do veículo certo ao trecho certo, analisamos, juntamente com o setor de planejamento da Construtora Marquise Ltda. — empreiteira que detem atualmente mais de 50% da coleta de lixo em Fortaleza, um modelo adequado à satisfação da operação, e ao uso sistemático da empresa, na decisão do processo movimentação da frota e designação dos veículos aos trechos de serviço.

III.2 Descrição do (PPF)

O (PPF) pode ser formulado da seguinte forma: Considerando que a garagem possui uma frota de n_v veículos disponíveis, sendo estes considerados apenas por poderem sair para realizar a coleta, portanto, não consideramos os veículos em oficina realizando manutenção corretiva ou cumprindo um programa de preventiva. Desejamos encontrar a composição da frota de mínimo custo/tonelada, que satisfaça o seguinte conjunto de restrições:

- *Deve haver um conjunto de veículos em reserva técnica que jamais exceda a 5% do total de veículos disponíveis e nunca menos que o maior inteiro de 10% do total de veículos disponíveis — Estas restrições de reserva são impostas pela Prefeitura em contrato;*
- *Os veículos em serviço, ou seja, com menos de 300 hs ou com mais de 300 hs de serviço (Estes últimos não absorvidos pela oficina por superlotação ou incapacidade de fazer o serviço de preventiva), devem somar em carga nominal no mínimo a carga mínima estimada a ser coletada, e no máximo a máxima carga estimada para coleta;*
- *Os tempos de serviço dos veículos que estão em reserva técnica deve exceder àqueles que estão em serviço, e/ou suas cargas nominais devem ser iguais ou inferiores àqueles em serviço;*
- *Os veículos em serviço não podem estar operando com mais de dois turnos de trabalho seguidos;*
- *O número de veículos em serviço deve ser exatamente o número de trechos que devem ser percorridos para o turno determinado de trabalho.*

Como vemos, as restrições buscam comandar o procedimento de planejamento da coleta, haja vista uma rotação prioritária do equipamento disponível. Imaginemos que desejemos realizar esta operação de rotatividade em tempo real

para uma quantidade razoável (≥ 10) de veículos. Tal tarefa poderia ser extremamente exaustiva, e nosso gerente precisaria alocar máquina ao homem conhecedor do circuito. Isto incorreria numa predileção por veículo, e o cuidado ou rejeição pelo equipamento está vinculado à utilização daquele que opera constantemente. Com a rotatividade, todos os motoristas seriam responsáveis pelo equipamento que lhe é destinado, e, presumimos ser possível um aumento na vida útil dos equipamentos.

Felizmente, com o auxílio da programação matemática e de suas técnicas, podemos modelar e resolver o problema de definição das disponibilidades de serviço.

O modelo que apresentamos a seguir, foi desenvolvido de modo a fornecer soluções para o problema de definição de frota, e este passo pode ser tomado como o primeiro na otimização da atividade de coleta.

Consideremos, então, as seguintes variáveis:

C_i — Custo operacional por tonelada (gastos com o veículo i);

Ton_i — Carga nominal do veículo i ;

TS_i — Tempo de Serviço entre preventivas do veículo i ;

TSC_i — Tempo de Serviço Corrido do veículo i ;

X_i $\begin{cases} 1, & \text{se o veículo } i \text{ com até } 300 \text{ hs é designado ao serviço;} \\ 0, & \text{caso contrário} \end{cases}$

R_i $\begin{cases} 1, & \text{se o veículo } i \text{ ficará em reserva técnica;} \\ 0, & \text{caso contrário} \end{cases}$

E_i $\begin{cases} 1, & \text{se o veículo } i \text{ com mais de } 300 \text{ hs será designado ao} \\ & \text{serviço;} \\ 0, & \text{caso contrário} \end{cases}$

XSC_i $\begin{cases} 1, & \text{se o veículo } i \text{ com tempo de serviço corrido não} \\ & \text{excede o limite total;} \\ 0, & \text{caso contrário} \end{cases}$

TTC — Tempo total de serviço corrido por veículo;

MaxCC — Máxima carga a ser coletada;

MinCC — Mínima carga a ser coletada;

$[a]$ — Maior inteiro maior ou igual a a ;

NTS — Número de trechos de serviço.

O modelo matemático para o (PFV), que se adequa à solução do problema, pode ser assim descrito:

$$(PFV) \quad \text{Min} \sum_{i=1}^{nv} C_i (X_i + E_i) \quad (III.1)$$

sujeito a:

$$TS_i X_i \leq 300 \quad (i = 1, \dots, nv) \quad (III.2)$$

Todos os veículos em X_i têm tempo de serviço igual ou inferior a 300 hs;

$$TSC_i XSC_i \leq TTC \quad (i = 1, \dots, nv) \quad (III.3)$$

O tempo de serviço corrido de cada veículo deve ser sempre inferior ou igual ao tempo total corrido especificado;

$$X_i + R_i + E_i = 1 \quad (i = 1, \dots, nv) \quad (III.4)$$

O veículo possui apenas um estado de operação;

$$X_i + E_i \leq XSC_i \quad (i = 1, \dots, nv) \quad (III.5)$$

O veículo i somente estrará em serviço se o seu tempo de serviço corrido for igual ou inferior a TTC ;

$$\sum_{i=1}^{nv} R_i \geq [5\%nv] \quad (III.6)$$

A reserva técnica não deve ser inferior ao maior inteiro de 5% do número de veículos disponíveis;

$$\sum_{i=1}^{nv} R_i \leq [10\%nv] \quad (III.7)$$

A reserva técnica não deve exceder ao maior inteiro de 10% do número de veículos disponíveis;

$$\sum_{i=1}^{nv} Ton_i (X_i + E_i) \geq MinCC \quad (III.8)$$

A soma total das cargas dos veículos disponíveis ao serviço deve ser maior ou igual à carga mínima a ser coletada;

$$\sum_{i=1}^{nv} Ton_i (X_i + E_i) \leq MaxCC \quad (III.9)$$

A soma total das cargas dos veículos disponíveis ao serviço deve ser menor ou igual à carga máxima a ser coletada;

$$\sum_{i=1}^{nv} (X_i + E_i) \geq NTS \quad (III.10)$$

O número total de veículos disponíveis ao serviço deve ser maior ou igual ao número total de trechos de serviço solicitados;

$$X_i, R_i, E_i, XSC_i \in \{0, 1\} \quad (i = 1, \dots, nv) \quad (III.11)$$

Variáveis de decisão do modelo.

Os custos diretos podem ser tomados em relação aos gastos com combustível, pneus, manutenção, etc. Tais valores entram na função objetivo de modo a permitir que se escolha a composição de uma frota que possua menor custo de serviço por tonelada.

Após termos o resultado do problema, o conjunto dos veículos em reserva e o conjunto dos veículos em espera de preventiva são observados, de modo que se tenha preferencialmente aqueles em reserva com o número de horas superior aos em espera de preventiva, e o custo por tonelada superior ao primeiro. Isto acontece como uma pós-otimização no processo, haja vista a necessidade de se ter preferencialmente veículos em piores condições na reserva, favorecendo com que a oficina possa dá-los maior atenção logo que esteja liberada.

Para cada turno de trabalho deve-se determinar a frota que estará disponível ao serviço, devendo-se, pois, atualizar os dados de cada veículo que realizou operação e indicar aqueles que farão o serviço de coleta do próximo turno de trabalho.

O último passo seria, então, distribuir os veículos em cada trecho de serviço. Como estes são fixos, a tarefa é atribuir um veículo para cada trecho de modo que seja minimizado o número de viagens totais que serão dadas. Temos, pois, um problema de atribuição. Seja o seguinte conjunto de variáveis:

NTS — Número de trechos de serviço;

NVp — Número de veículos designados ao serviço pelo planejamento;

nvg_{ij} — Número de viagens do veículo i no trecho j ;

Q_j — Carga estimada de coleta no trecho de serviço j ;

Q_{vi} — Carga nominal do veículo i ;

CA_{ij} — Custo de utilização do veículo i no trecho j , relativo ao produto do seu custo operacional por tonelada pelo número de viagens nesse trecho, onde:

$$nvg_{ij} = \lceil Q_j / Q_{vi} \rceil \text{ e}$$

$$CA_{ij} = C_i nvg_{ij}$$

$$X_{ij} \begin{cases} 1, & \text{se o veículo } i \text{ for atribuído ao trecho } j; \\ 0, & \text{caso contrário} \end{cases}$$

$$X_{ji} \begin{cases} 1, & \text{se o trecho } j \text{ for ligado ao veículo } i; \\ 0, & \text{caso contrário} \end{cases}$$

O modelo de programação matemática para a Atribuição dos Veículos aos Trechos (AVT), é pois assim formulado:

$$(AVT) \quad \text{Min} \sum_{i=1}^{nvp} \sum_{j=1}^{nts} CA_{ij} X_{ij} \quad (III.12)$$

sujeito a:

$$\sum_{j=1}^{nts} X_{ij} = 1 \quad (i = 1, \dots, nvp) \quad (III.13)$$

Restrição que garante a ligação de apenas um veículo a um trecho;

$$\sum_{i=1}^{nvp} X_{ji} = 1 \quad (j = 1, \dots, nts) \quad (III.14)$$

Restrição que garante a ligação de apenas um trecho a um veículo;

$$X_{ij} \in \{0, 1\} \quad (i = 1, \dots, nvd), \quad (j = 1, \dots, nts) \quad (\text{III.15})$$

Variáveis de decisão do modelo.

Note que pode haver mais veículos do que trechos, de acordo com a restrição (III.11) do (PPF). Caso isto ocorra, na solução do (AVT), cria-se trechos artificiais com carga infinita, procedendo normalmente a resolução do problema proposto. E os veículos alocados a estes ficarão em reserva.

De posse dos resultados da atribuição, o gerente pode, então, comandar a operação de serviço, em tempo hábil e preocupar-se, doravante, apenas com outros serviços que seriam por exemplo o de comandar problemas emergentes na coleta. Tais problemas também podem ter o auxílio do processo decisório, desde que o gerente deseje alocar um veículo em um trecho onde houve quebra de um de seus veículos, sendo então necessário retirar equipamento da reserva técnica.

Para também atribuir a tripulação a determinado trecho, podemos notar que os motoristas já conhecem bem as suas áreas de trabalho. Esta predileção é comum e compreensível, pois normalmente eles contam com a ajuda de pessoas amigas que lhes oferecem água e até mesmo lanches. Um processo decisório que deve ser implementado em qualquer sistema que trate o problema da coleta, deve ser o de considerar o motorista ao trecho predileto de serviço, isto garantiria uma aceitação do plano de execução da coleta e uma maior garantia no retorno do investimento. Não discutiremos, porém, este enfoque, haja vista não estarmos abordando também o Problema de Planejamento de Pessoal.

III.3 Aplicação do Planejamento e Alocação de Frotas

Para o modelo (PPF), observamos que o algoritmo de Balas, MACULAN (1978), implementado por GEOFFRION (1972), destinado a problemas de Programação Inteira $\{0,1\}$, poderia ser adequado à sua solução. Apesar de ser um algoritmo enumerativo de complexidade exponencial $O(2^n)$ (onde n é número de variáveis de decisão do modelo), podemos dizer que esta técnica pode fornecer, em determinado momento, soluções viáveis bastante próximas do ótimo, haja vista a estrutura do problema. De posse da solução viável, utilizamos um refinamento heurístico, como já foi dito antes, procurando chegar na melhor solução gerencial. Este refinamento basicamente faz mudanças de estado em situações ocasionais em que a quantidade de veículos disponíveis é bem grande e a sua colocação poderá ser melhorada a partir da solução viável proposta pelo algoritmo.

A complexidade do refinamento é polinomial $O(nv)$, onde nv é o número de veículos disponíveis, o que garante o seu bom desempenho. Como a

aplicação foi desenvolvida em microcomputadores da linha PC, o limite de frota estabelecido no **S i s R o t** é de 300 veículos, e o tempo computacional é sempre consideravelmente alto, mas para a atividade é perfeitamente suportável.

A seguir, mostramos alguns exemplos de operação do planejamento, para casos testes aleatórios com doze veículos, realizados em um microcomputador da linha PC/AT-12MHz com Co-Processador Aritmético 80287.

1. Admita que temos a seguinte frota disponível de 12 veículos coletores, quando desejamos implantar o programa de planejamento de frotas, na seguinte situação:

<i>Código</i>	<i>TON</i>	<i>TS</i>	<i>TSC</i>	<i>Custo</i>
001	7,5	259	0	730
002	7,5	150	0	650
003	7,5	160	0	900
004	7,5	370	0	256
005	6,5	180	0	367
006	6,5	310	0	268
007	6,5	340	0	150
008	6,5	150	0	460
009	6,5	69	0	670
010	6,5	38	0	390
011	6,5	122	0	556
012	6,5	123	0	874

Seja, $\text{MaxCC} = 100 \text{ Ton}$, $\text{MinCC} = 20 \text{ Ton}$ e $\text{NTS} = 8$

Desejamos encontrar a disposição ótima dos veículos, nas seguintes situações:

$\text{TTC} = 12 \text{ hs}$, $\text{MaxRes} = 3$ e $\text{MinRes} = 1$;

Solução (1)

- *Custo Total da Frota Disponível (Cr\$/Ton) : 814.31*
- *Tempo de Trabalho Corrido (Máx. Permitido) : 12.0 hs.*
- *Carga Total Disponível: 61.50*
- *Tempo de Processamento: 0:00:24.55*

Distribuição da Frota:

- *Veículos Alocados a Serviço:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
<i>001</i>	<i>259.00</i>	<i>0.00</i>	<i>7.50</i>
<i>002</i>	<i>150.00</i>	<i>0.00</i>	<i>7.50</i>
<i>003</i>	<i>160.00</i>	<i>0.00</i>	<i>7.50</i>
<i>005</i>	<i>180.00</i>	<i>0.00</i>	<i>6.50</i>
<i>008</i>	<i>150.00</i>	<i>0.00</i>	<i>6.50</i>
<i>009</i>	<i>69.00</i>	<i>0.00</i>	<i>6.50</i>
<i>010</i>	<i>38.00</i>	<i>0.00</i>	<i>6.50</i>
<i>011</i>	<i>122.00</i>	<i>0.00</i>	<i>6.50</i>
<i>012</i>	<i>123.00</i>	<i>0.00</i>	<i>6.50</i>

- *Veículos em Reserva Técnica:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
<i>004</i>	<i>370.00</i>	<i>0.00</i>	<i>7.50</i>
<i>006</i>	<i>310.00</i>	<i>0.00</i>	<i>6.50</i>
<i>007</i>	<i>340.00</i>	<i>0.00</i>	<i>6.50</i>

- *Veículos em Espera de Preventiva:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
<i>004</i>	<i>370.00</i>	<i>0.00</i>	<i>7.50</i>
<i>006</i>	<i>310.00</i>	<i>0.00</i>	<i>6.50</i>
<i>007</i>	<i>340.00</i>	<i>0.00</i>	<i>6.50</i>

2. Após implantado o programa de planejamento, admita que em determinado dia e turno de operação desejamos realizar o planejamento dos veículos do exemplo anterior, agora na seguinte situação:

Cód.	TON	TS	TSC	Custo
001	7,5	259	0	730
002	7,5	150	13	650
003	7,5	160	7	900
004	7,5	370	8	256
005	6,5	180	10	367
006	6,5	310	14	268
007	6,5	340	9	150
008	6,5	150	5	460
009	6,5	69	4	670
010	6,5	38	7	390
011	6,5	122	10	556
012	6,5	123	12	874

Seja, $\text{MaxCC} = 100 \text{ Ton}$, $\text{MinCC} = 30 \text{ Ton}$ e $\text{NTS} = 8$

Desejamos encontrar a disposição ótima dos veículos, nas seguintes situações:

(2.a) $\text{TTC} = 12 \text{ hs}$, $\text{MaxRes} = 3$ e $\text{MinRes} = 1$;

(2.b) $\text{TTC} = 10 \text{ hs}$, $\text{MaxRes} = 4$ e $\text{MinRes} = 1$;

Solução (2.a)

- *Custo Total da Frota Disponível (Cr\$/Ton) : 750.72*
- *Tempo de Trabalho Corrido (Máx. Permitido) : 12.00 hs.*
- *Carga Total Disponível: 60.50*
- *Tempo de Processamento: 0:00:31.75*

Distribuição da Frota:• *Veículos Alocados a Serviço:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
001	259.00	0.00	7.50
003	160.00	7.00	7.50
005	180.00	10.00	6.50
007	340.00	9.00	6.50
008	150.00	5.00	6.50
009	69.00	4.00	6.50
010	38.00	7.00	6.50
011	122.00	10.00	6.50
012	123.00	12.00	6.50

• *Veículos em Reserva Técnica:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
002	150.00	13.00	7.50
004	370.00	8.00	7.50
006	310.00	14.00	6.50

• *Veículos em Espera de Preventiva:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
004	370.00	8.00	7.50
006	310.00	14.00	6.50
007	340.00	9.00	6.50

Solução (2.b)

- *Custo Total da Frota Disponível (Cr\$/Ton) : 611.26*
- *Tempo de Trabalho Corrido (Máx. Permitido) : 10.00 hs.*
- *Carga Total Disponível: 54.00*
- *Tempo de Processamento: 0:00:33.67*

Distribuição da Frota:● *Veículos Alocados a Serviço:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
001	259.00	0.00	7.50
003	160.00	7.00	7.50
005	180.00	10.00	6.50
007	340.00	9.00	6.50
008	150.00	5.00	6.50
009	69.00	4.00	6.50
010	38.00	7.00	6.50
011	122.00	10.00	6.50

● *Veículos em Reserva Técnica:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
002	150.00	13.00	7.50
004	370.00	8.00	7.50
006	310.00	14.00	6.50
012	123.00	12.00	6.50

● *Veículos em Espera de Preventiva:*

<i>Cód.</i>	<i>TS</i>	<i>TSC</i>	<i>Ton</i>
004	370.00	8.00	7.50
006	310.00	14.00	6.50
007	340.00	9.00	6.50

Propositadamente colocamos uma variação na reserva técnica, pois na verdade, ela varia conforme as necessidades do operador ou gerente. Quando colocamos as restrições (III.7) e (III.8), buscamos, em resumo, determinar a política da prefeitura em primeiro lugar. Na prática, as empresas normalmente trabalham com sua capacidade esgotada; sendo assim, tudo o que se pode fazer para resolver o problema é liberar o limite superior do número de veículos de modo que o gerente cumpra o seu papel de obter a reserva com menos ou mais veículos, conforme a possibilidade de encaixar no modelo os limites impostos.

Continuando com o exemplo anterior, desejamos agora alocar os veículos de modo que haja uma satisfação da demanda dos trechos com o menor custo por viagem. E considerando a seguinte situação da demanda dos trechos a serem servidos, temos:

Código	Trecho	Carga Média	Desvio
001	007-016	20470	4300
002	007-010	18470	2600
003	007-012	17300	4630
004	007-002	15820	3740
005	007-014	14990	3610
006	007-008	13130	3260
007	007-006	12120	5120
008	007-004	9500	3860

Resolvendo o problema (AVT), temos:

Solução (1)

• Solução no Limite Inferior de Carga a Coletar

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
010	6.5	390.00	007-004	2	5720
005	6.5	367.00	007-006	2	7000
008	6.5	460.00	007-008	2	4935
011	6.5	556.00	007-014	3	5690
009	6.5	670.00	007-002	3	6040
002	7.5	650.00	007-012	3	6335
001	7.5	730.00	007-010	3	7935
003	7.5	900.00	007-016	3	8085

Veículo(s) que Retorna(m) à Reserva Técnica:

012 \Rightarrow 6.5 Ton/Cr\$ 874.00

• Solução no Valor Médio de Carga a Coletar

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
011	6.5	556.00	007-004	2	4790
005	6.5	367.00	007-006	2	6060
010	6.5	390.00	007-008	2	6565
008	6.5	460.00	007-014	3	4996
009	6.5	670.00	007-002	3	5273
012	6.5	874.00	007-012	3	5766
002	7.5	650.00	007-010	3	6156
001	7.5	730.00	007-016	3	6823

Veículo(s) que Retorna(m) à Reserva Técnica:

003 \Rightarrow 7.5 Ton/Cr\$ 900.00

• Solução no Limite Superior de Carga a Coletar

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
008	6.5	460.00	007-004	2	6720
011	6.5	556.00	007-006	2	5746
005	6.5	367.00	007-008	2	5463
009	6.5	670.00	007-014	2	6200
010	6.5	390.00	007-002	3	6520
002	7.5	650.00	007-012	3	7310
001	7.5	730.00	007-010	3	7023
012	6.5	874.00	007-016	3	6192

Veículo(s) que Retorna(m) à Reserva Técnica:

008 ⇒ 7.5 Ton/Cr\$ 900.00

Solução (2.a)

• Solução no Limite Inferior de Carga a Coletar

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
010	6.5	390.00	007-004	2	5720
011	6.5	556.00	007-006	2	7000
008	6.5	460.00	007-008	2	4935
009	6.5	670.00	007-014	2	5690
007	6.5	150.00	007-002	3	6040
005	6.5	367.00	007-012	3	6335
001	7.5	730.00	007-010	3	7935
003	7.5	900.00	007-016	3	8085

Veículo(s) que Retorna(m) à Reserva Técnica:

012 ⇒ 6.5 Ton/Cr\$ 874.00

• Solução no Valor Médio de Carga a Coletar

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
007	6.5	150.00	007-004	2	4790
008	6.5	460.00	007-006	2	6060
010	6.5	390.00	007-008	2	6565
009	6.5	670.00	007-014	2	7495
005	6.5	367.00	007-002	3	5273
011	6.5	556.00	007-012	3	5766
001	7.5	730.00	007-010	3	6156
012	6.5	874.00	007-016	3	6823

Veículo(s) que Retorna(m) à Reserva Técnica:

003 ⇒ 7.5 Ton/Cr\$ 900.00

• Solução no Limite Superior de Carga a Coletar

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
005	6.5	367.00	007-004	2	6720
011	6.5	556.00	007-006	2	5746
010	6.5	390.00	007-008	2	5463
007	6.5	150.00	007-014	2	6200
008	6.5	460.00	007-002	3	6520
009	6.5	670.00	007-012	3	7310
001	7.5	730.00	007-010	3	7023
012	6.5	874.00	007-016	3	6192

Veículo(s) que Retorna(m) à Reserva Técnica:

003 ⇒ 7.5 Ton/Cr\$ 900.00

Solução (2.b)

• Solução no Limite Inferior de Carga a Coletar

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
008	6.5	460.00	007-004	2	5720
010	6.5	390.00	007-006	2	7000
011	6.5	556.00	007-008	2	4935
007	6.5	150.00	007-014	2	5690
005	6.5	367.00	007-002	3	6040
009	6.5	670.00	007-012	3	6335
001	7.5	730.00	007-010	3	7935
003	7.5	900.00	007-016	3	8085

• Solução no Valor Médio de Carga a Coletar

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
007	6.5	150.00	007-004	2	4790
010	6.5	390.00	007-006	2	6060
005	6.5	367.00	007-008	2	6565
008	6.5	460.00	007-014	2	7495
011	6.5	556.00	007-002	3	5273
009	7.5	670.00	007-012	3	5766
001	6.5	730.00	007-010	3	6156
003	7.5	900.00	007-016	3	6823

• *Solução no Limite Superior de Carga a Coletar*

Veículo	Ton	Cr\$	Trecho	NVG	Ton/Viagem
008	6.5	460.00	007-004	2	6720
005	6.5	367.00	007-006	2	5746
010	6.5	390.00	007-008	2	5463
007	6.5	150.00	007-014	2	6200
011	6.5	556.00	007-002	3	6520
009	6.5	670.00	007-012	3	7310
001	7.5	730.00	007-010	3	7023
003	7.5	900.00	007-016	3	8256

III.4 Observações e Conclusão

Quando comentamos acerca do tempo de operação, levamos em conta uma frota muito grande (≥ 100) veículos. Neste caso, seria necessário a implementação de um algoritmo enumerativo de melhor desempenho, de modo que a solução obtida possa ser avaliada rapidamente e em seguida haja a possibilidade imediata de execução do plano.

Os dados aqui apresentados, foram extraídos de NEGREIROS ET AL (1990), relativo ao trabalho de levantamento de dados em campo, na cidade de Fortaleza, no período de Maio e Junho de 1989, numa área de grande concentração de comércio, hotéis, clubes, escolas e residências.

Possivelmente algumas dúvidas poderão surgir quanto à pós-otimização, que em alguns casos pode aumentar o custo da função objetivo. Isto se deve ao fato de que o veículo que está em *Espera de Preventiva*, já possui sua tarefa de 300 hs esgotada, e sempre terá a predileção na *Reserva Técnica*.

O enfoque de planejamento com alocação, parece ser intuitivo nas operações de coleta. Na verdade, a decisão errada poderá aumentar sobremaneira os custos operacionais da atividade. Não resta dúvidas que esta metodologia serviria para orientar com bastante propriedade a atividade, não obstante, o papel do gerente é de extrema relevância no processo, pois as informações que constam no sistema às vezes podem não conter *certas nuances* da operação, apesar desse tentar abordar todo o processo. De qualquer modo, caso seja adotado um rigor no desenvolvimento da tarefa, acreditamos que a operação poderá ser cumprida sem problemas.

Capítulo IV

O Problema de Percurso na Coleta do Lixo Domiciliar

IV.1 Introdução

De acordo com a literatura — CLARK & GILLEAN (1975), CLARK & LEE (1976), BODIN ET AL (1989) e como vimos na introdução deste trabalho, as atividades de coleta representam cerca de 80% dos custos totais sobre toda a operação do lixo. Em virtude disso, é importante que se faça um planejamento nessa atividade de modo a reduzir os custos operacionais, que geralmente são bastante significativos. Além de se planejar os veículos que farão o serviço de coleta, deve-se ter em mente que a realização da tarefa de coleta, quando é feita com orientação de percurso, e este pensado de forma otimizada, trás resultados surpreendentemente animadores.

De acordo com CLARK & GILLEAN (1975), o gerenciamento do problema do lixo, pode ser dividido em duas áreas: COLETA, incluindo armazenagem (nos caminhões), transferência e transporte, e DESPEJO, incluindo qualquer atividade de tratamento. A operação de coleta pode ser subdividida em duas unidades de operação, COLETA e ARRASTO. A unidade de operação de COLETA, consiste na remoção do lixo do ponto de armazenamento no local de geração. Esta operação inicia quando os veículos coletores saem da garagem e incluem todo o tempo levado na rota, se ou não ele for produtivo. A unidade de operação de ARRASTO começa quando o veículo de coleta sai do local onde a última carga foi recolhida, e inclui o tempo que leva no local de remoção e o tempo após a saída do local e retorno ao primeiro ponto de coleta da próxima rota, ou para o depósito temporário quando o dia de trabalho for concluído. A unidade de ARRASTO inclui, portanto, o tempo total da viagem da rota de coleta para o local de remoção.

Três alternativas são consideradas na remoção do lixo: Carregamento direto do município para o aterro sanitário; Carregamento direto para uma estação de transbordo onde o lixo é transferido para veículos maiores e embarcados para uma última remoção, e Carregamento direto do município para um incinerador onde o lixo é queimado e o resíduo é embarcado para uma última remoção.

Embora muitos estudiosos tenham explorado a aplicação de modelos de simulação determinísticos, analíticos, e estocásticos para o gerenciamento do lixo urbano, suas aplicações têm sido baseadas tanto em um banco de dados sem base de observações ou num com base em observações. Estas observações têm sido usadas para demonstrar o potencial dos sistemas ou técnicas de pesquisa operacional, no caso do roteamento e planejamento de frotas, e a aquisição de dados obtidas por séries históricas bem controladas.

Nossa preocupação, no entanto, foi criar um sistema planejador de frotas e gerador de rotas de veículos, que pudesse ser ágil na formação de rotas, e se utilizasse de estimativas coerentes de modo que fosse possível a realização de serviços eficientes de coleta, considerando o início e fim do trabalho dentro das especificações impostas pela prefeitura.

Um sistema chamado **S i s R o t** foi desenvolvido, integrando um banco de dados geográfico, e um acompanhamento de frotas, com a capacidade de gerenciar a atividade de coleta urbana para o caso específico aqui estudado. (Ver Apêndice 4)

Levar em consideração todos os itens que compõem a sistematização da coleta do lixo, é uma tarefa um pouco difícil, por se tratar de um sistema que manipula um sem número de situações ocasionais. Acreditar no funcionamento deste, é outro problema, pois os resultados estão sempre de acordo com informações aproximadas. Como qualquer atividade séria, a coleta do lixo necessita de um sistema que analise o serviço e libere um pouco o indivíduo que trabalha diretamente com coleta, pois o trabalho é de extremo desgaste físico, o que favorece uma grande rotatividade de pessoal — **NEGREIROS ET AL (1990)**. Como já não bastasse o serviço difícil, as informações de controle necessitam ser de fácil acesso e extremamente flexíveis à tomada de decisão, conforme os problemas fora de controle surjam no decorrer do processo de coleta.

IV.2 Determinando o Problema da Coleta do Lixo Urbano

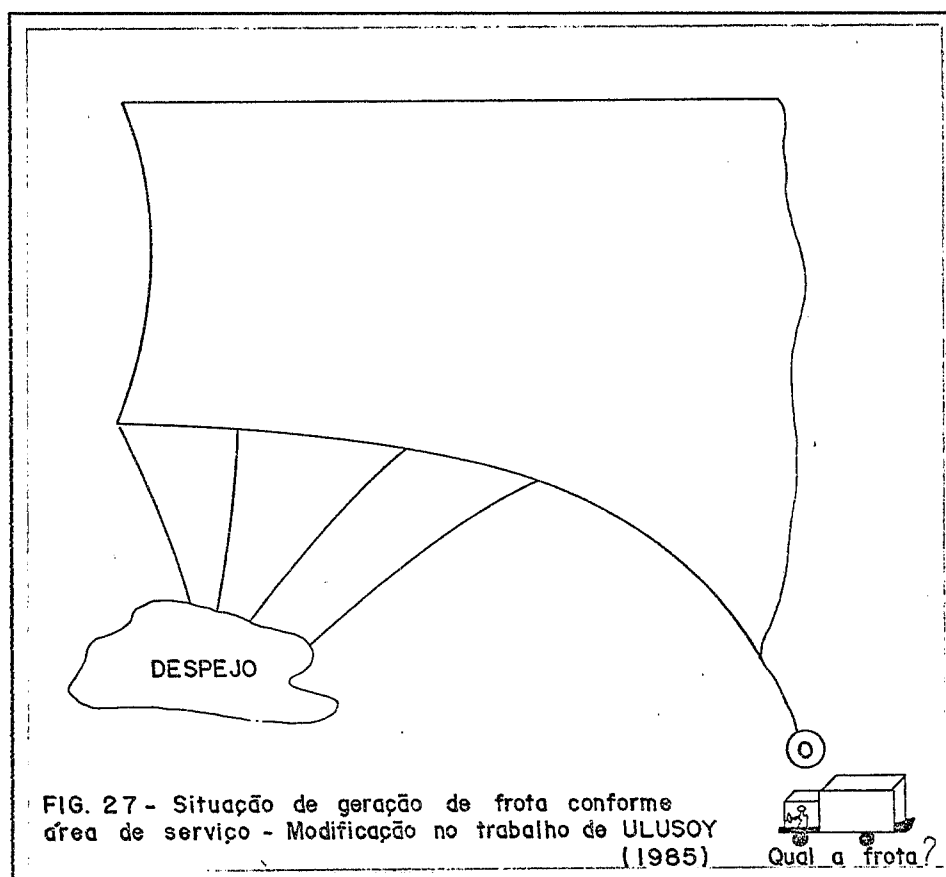
O estudo do Problema da Coleta de Lixo Urbano que desenvolvemos, está vinculado exclusivamente na determinação de alocação de frota e percurso de veículos no cumprimento das atividades de coleta do lixo. Como vimos anteriormente, o processo de coleta é executado tanto pelos veículos quanto por garis, que levam aos veículos o lixo de sacos ou camburões. Assim tanto o serviço braçal influencia no mecânico quanto vice-versa.

Durante nosso trabalho de levantamento em campo, observamos a atividade mecânica tanto do ponto de vista contratual (Prefeitura) quanto do operacional. O compromisso da limpeza urbana é passar por toda via em condições de tráfego, servindo-a. Todo o trabalho de remoção, aparentemente é mecânico. Na verdade, em determinadas vias sem saída onde os veículos não podem entrar,

o serviço é realizado com o estacionamento do caminhão coletor e o trabalho de evacuação do lixo é feito totalmente pelos garis. Do ponto de vista da empresa que presta o serviço de coleta, tentar-se-á cumprir a tarefa, minimizando o custo deste trabalho com equipamento em atividade no serviço, respeitando não somente o critério de satisfação da demanda, como também os horários de serviço da tripulação, as imposições de tráfego (mão única, contra-fluxo), e a carga dos veículos à disposição.

Aparentemente, o estudo começaria de seguinte forma: Dada uma área de serviço, e um conjunto de tipos diferentes de veículos, disponíveis no mercado, deseja-se formar uma frota e os seus respectivos percursos, que possam servir toda a área para ele definida, respeitando as restrições de tempo da tripulação e sua carga total, com custo associado minimizado.

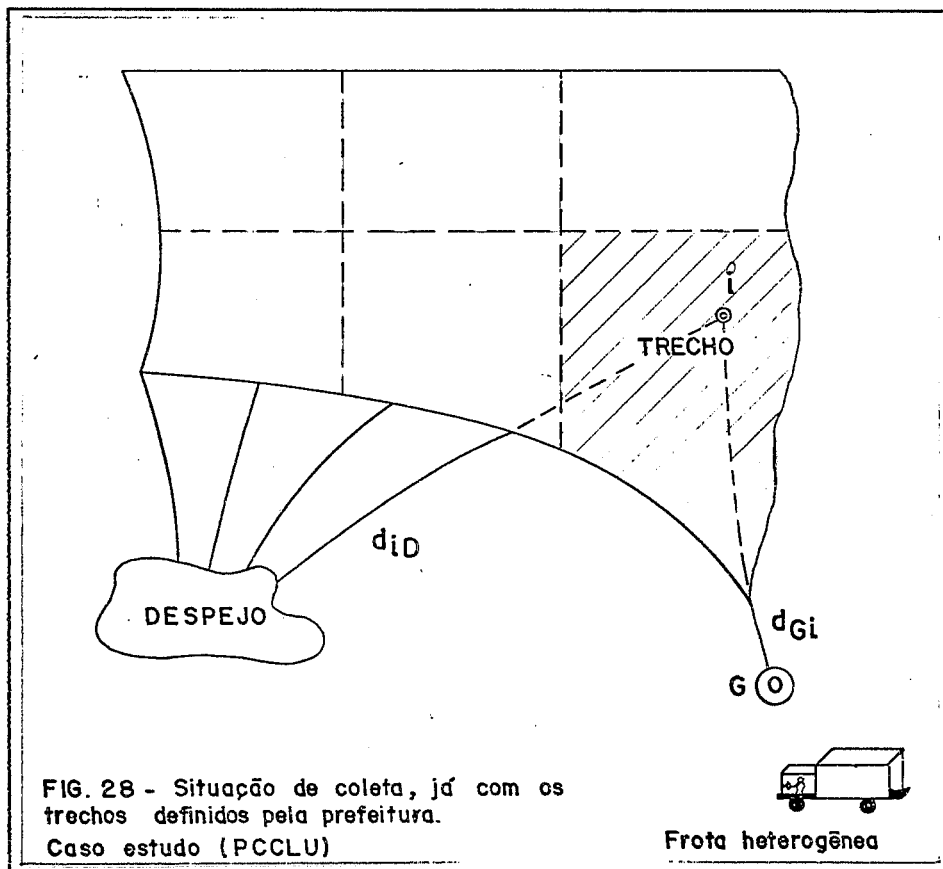
O problema acima parece grosseiramente com o problema proposto por ULUSOY (1985), (ver Fig. 27). A diferença é que os veículos podem retornar ao serviço mesmo depois de terem completado a sua carga e descarregado, em um ponto senão o de origem, ou garagem – no nosso caso, o local de despejo. Como o tempo de serviço, durante o carregamento, implicaria no cumprimento ou não de determinado trecho por determinado veículo, tal fato é relevante no critério de decisão e na execução de uma tarefa que necessite ser planejada em termos de horário e carregamento por viagem, todos dependentes e equilibrados, também visando garantir um aumento no tempo de vida médio dos veículos coletores.



A abordagem que apresentamos, pode ser feita a partir de uma rede-

finição operacional feita pela prefeitura. Onde a ela caberia administrar e fiscalizar os trechos de coleta, resultado do extenso e elaborado plano de operação. Tal fato, indicaria uma definição apropriada da frota de serviço, custos associados, e uma política mais justa na divisão das tarefas entre empresas privadas interessadas na coleta do lixo.

O que temos normalmente, porém, é um procedimento um pouco diferente daquele mencionado anteriormente. Para a maioria das cidades existe uma definição das áreas de coleta e estas subdivididas em trechos que estão dimensionados conforme a carga de um veículo padrão e um tempo estimado de realizar o serviço não muito superior à carga de trabalho de uma tripulação (normalmente de 7hs e 20 min) – Ver Fig. 28. A frota das empresas já é *a priori* conhecida, e o trabalho precisa ser realizado *com o que se tem*. Desta maneira, a coleta do lixo se processa com uma filosofia operacional bem diferente e voltada exclusivamente para as condições de serviço das empresas. A administração e execução dos serviços de coleta do lixo, quando feitas exclusivamente pela prefeitura, pode ser extremamente onerosa e perigosa, do ponto de vista político.



Em Fortaleza, nosso local de estudo, como em outras capitais brasileiras, a coleta do lixo se processa através do comando administrativo e fiscal realizado pela prefeitura, e a execução do serviço pelas empresas empreitadas. Apesar da prefeitura ainda ter o total controle das atividades em determinados trechos importantes, como no centro da cidade, e outros pólos comerciais, o maior volume de coleta é feito exclusivamente pela iniciativa privada. Isto reforça o fato de que as empresas prestadoras de serviço procuram administrar a execução da coleta, mantendo

ao máximo a operação e o serviço, de acordo com as normas contratuais, visando reduzir os seus custos operacionais ou garantir eficiência com toda a limitação de investimentos que possuem.

Do ponto de vista da empresa prestadora do serviço, o problema é, então, cumprir os trechos contratados com mínimo custo operacional, que procure desenvolver uma rotatividade na frota e respeitando as reservas impostas pela prefeitura e os horários por ela determinados.

O problema acima já possui uma área pré-definida para serviço. O que temos então é: Dada uma frota disponível na garagem, desejamos distribuí-la entre os diversos trechos de coleta, de modo que seja reduzido o número de viagens ao local de despejo, que seja alocado o veículo certo minimizando sua ociosidade e aumentando o seu tempo de vida, como já foi mostrado no capítulo anterior. Já para o percurso, o trabalho deverá ser realizado totalmente em quarteirões onde os veículos possam trafegar, e, nos locais onde os veículos não possam entrar, como vilas, travessas, etc. - Pontos de Coleta ou Pontos Especiais de Coleta, a limpeza será feita com o veículo estacionado e os garis removendo o lixo do local até o caminhão. Deve-se considerar que a rede viária é composta por vias de mão-única, vias com contra-fluxo e mão-dupla.

Haja vista termos dois tipos de demanda, ARCOS e NÓS, numa rede mista, com horários de serviço da tripulação, e pontos específicos de início e fim de trabalho, chamá-lo-emos de Problema do Caminhão Coletor de Lixo Urbano (PCCLU). Apesar de ser muito semelhante ao PCA, o PCCLU abre um pouco o conceito deste impondo uma rede mista e o início e fim de serviço podendo ser diferente. É bom observar que para o PCCLU, toda a demanda de um arco ou nó pode ser suprida pelo veículo, pois este tem capacidade maior ou igual àqueles.

Considerando as seguintes variáveis como:

nvg — Número de viagens do veículo coletor;

Q — Carga nominal associada ao veículo;

T — Tempo limite de serviço;

t_{ij} — Tempo de atravessar o arco/elo (i,j) sem servi-lo;

t_{ij}^o — Tempo de servir o arco/elo (i,j) ;

c_{ij}^s — Custo de atravessar o arco/elo (i,j) na s -ésima viagem;

$(i, D)/iD$ — Arco/Elo direcionado que sai de um nó qualquer i , e chega em D , o qual consideramos como o nó Despejo;

$(i, G)/iG$ — Arco/Elo direcionado que sai de um nó qualquer i , e chega em G , designado como o nó Garagem;

Ψ — Conjunto dos arcos/elos que formam o percurso mais rápido entre o local de Despejo e a Garagem;

$NADG$ — Número de arcos no caminho mais rápido entre os vértices Despejo e Garagem;

$$x_{ij}^s \begin{cases} 1, & \text{se o veículo atravessa o arco/elo } (i,j) \text{ na } s\text{-ésima viagem;} \\ 0, & \text{caso contrário} \end{cases}$$

$$l_{ij}^s \begin{cases} 1, & \text{se o veículo atravessa coletando o arco/elo } (i,j) \text{ na} \\ & s\text{-ésima viagem;} \\ 0, & \text{caso contrário} \end{cases}$$

$TP(s)$ — Tempo de parada para descando ou despejo, na s -ésima viagem;

f_{ij}^s — Variáveis de fluxo que evitam percursos ilegais;

n — Número de vértices da Rede Viária;

E — Conjunto dos elos da rede viária (mão-dupla);

A — Conjunto dos arcos da rede viária (mão-única);

$S_1 \subseteq E \cup A$ — Conjunto de arcos/elos com demanda;

O modelo em programação matemática para o PCCLU pode ser assim descrito:

$$(PCCLU) \quad \text{Min} \sum_{i=1}^n \sum_{j=1}^n \sum_{s=1}^{nvg} c_{ij}^s x_{ij}^s \quad (IV.1)$$

sujeito a:

$$\sum_{i=1}^n x_{ij}^s - \sum_{k=1}^n x_{jk}^s = 0 \quad (j = 1, 2, \dots, n), \quad (s = 1, \dots, nvg) \quad (IV.2)$$

Condição que garante que o número de passagens por determinado nó é par;

$$\sum_{s=1}^{nvg} (l_{ij}^s - l_{ji}^s) = 1 \quad (i, j) \in S_1 \quad (IV.3)$$

Condição que garante o serviço em uma única direção, em uma única viagem, num determinado arco/elo de S_1 ;

$$x_{ij}^s \geq l_{ij}^s \quad (i, j = 1, 2, \dots, n), \quad (s = 1, \dots, nvsg) \quad (IV.4)$$

Condição que garante a passagem do veículo na s -ésima viagem coletando ou não no arco/elo da rede viária;

$$\sum_{i=1}^n \sum_{j=1}^n q_{ij} l_{ij}^s \leq Q \quad (s = 1, \dots, nvsg) \quad (IV.5)$$

Condição que garante a possibilidade de carregamento do veículo, servindo o arco/elo, de i para j , respeitando o seu limite de carga;

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{s=1}^{nvsg} [(x_{ij}^s - l_{ij}^s) t_{ij} + l_{ij}^s t_{ij}^o + TP(s)] \leq T \quad (IV.6)$$

Condição que limita o tempo de serviço total de acordo com o estabelecido por veículo;

$$\sum_{i=1}^n \sum_{s=1}^{nvsg} x_{iD}^s \geq 1 \quad (IV.7)$$

Condição que garante no mínimo um despejo realizado pelo veículo;

$$\sum_{i=1}^n \sum_{s=1}^{nvsg} \sum_{s=1}^{nvsg} x_{iG}^s = 1 \quad (IV.8)$$

Condição que garante um único retorno à garagem;

$$u_{ij} \begin{cases} 1, & \text{se } \sum_{s=1}^{nvsg} x_{ij}^s \geq 1 \quad \forall (i, j) \in \Psi \\ 0, & \text{caso contrário} \end{cases} \quad (IV.9)$$

$$\sum_{(i, j) \in \Psi} u_{ij} = NADG \quad (IV.10)$$

Condições que garantem a passagem pelo caminho mais rápido entre Despejo e Garagem apenas uma vez.

Note que estas restrições permitem que sejam usados os arcos intermediários do caminho mínimo (D,G), entre o início e fim da coleta. Como somente há a possibilidade de chegar à garagem apenas uma vez, então estas restrições garantem o retorno à garagem do local de Despejo;

$$\sum_{k=1}^n f_{ik}^s - \sum_{k=1}^n f_{ki}^s = \sum_{j=1}^n l_{ij}^s \quad (i = 2, \dots, n), \quad (s = 1, \dots, nvq) \quad (\text{IV.11})$$

$$f_{ij}^s \leq n^2 x_{ij}^s \quad (i, j = 1, \dots, n) \quad (s = 1, \dots, nvq) \quad (\text{IV.12})$$

$$f_{ij}^s \geq 0 \quad (i, j = 1, \dots, n) \quad (s = 1, \dots, nvq) \quad (\text{IV.13})$$

Condições para a não formação de rotas ilegais – GOLDEN & WONG (1981);

$$x_{ij}^s, l_{ij}^s \in \{0, 1\} \quad (i, j = 1, \dots, n) \quad (\text{IV.14})$$

Variáveis de decisão do modelo.

Como notamos acima, o modelo procura resolver o problema da coleta de lixo, minimizando as passagens ociosas em arcos/elos da rede viária. Tal fato, no geral, minimiza os custos operacionais quando tratamos apenas o problema do percurso do veículo no trecho pré-determinado. Notamos também que toda informação necessária parece determinística, apesar de, em se tratando de lixo não o ser. De acordo com nossa experiência em campo, séries históricas de produção foram levantadas e nos levou a considerar o pior caso de geração, como aquele em que sugeriríamos ao processo de planejamento da coleta. Tal consideração não causa grande discrepância com a realidade, mesmo porque os motoristas precisam estar habituados a fazer determinado percurso, haja vista ser este um modo eficiente de tornar executável a coleta. Nos trabalhos de MCBRIDGE (1985), BODIN & KURSH (1981), CLARKE & GILLEAN (1975), dentre outros, comenta-se que é pouco interessante uma modificação rotineira nos circuitos, em função da adaptação dos motoristas à execução de suas tarefas. Isto sugere, então, que acreditemos ser o procedimento atualmente utilizado como o ideal na resolução do problema da coleta de lixo urbano.

IV.3 O Algoritmo MARCOLIX

Em nossa pesquisa bibliográfica, várias aplicações que tratavam do problema da coleta do lixo, todas elas, no entanto, enfocando apenas a situação em que cada veículo partia de um depósito central, e o final do trabalho, estava sempre vinculado ao *enchimento* do veículo, não importando a que distância estivesse do local de despejo mais próximo — BODIN ET AL (1989), MCBRIDGE (1982), TURNER ET AL (1976), WYSKIDA & GUPTA (1972), BOAVENTURA ET AL (1985), MATION (1982), CHRISTOFIDES (1973), BODIN & BELTRAMI (1974).

Em todas as idéias, o enfoque ao PCA pareceu evidente, mas, mesmo considerando que as soluções foram bem elaboradas e bastante recursivas, havia sempre um tratamento diferenciado entre o caso direcionado e o não-direcionado, e pouco ou nada se viu para o caso misto considerando o tempo de trabalho da tripulação.

Como mostramos anteriormente, o trabalho de FREDERICKSON (1979) pareceu esquecido em se tratando de aplicações mais realistas, e infelizmente não compreendemos o por quê de não se ter desenvolvido em estudo mais cuidadoso utilizando esta heurística. Talvez o fato do problema misto não poder obter soluções exatas para muitos casos, por ser \mathcal{NP} -HARD, seja o motivo suficiente para abandoná-lo.

Achamos, porém, que do ponto de vista prático, estamos a todo momento querendo simular a realidade com informações aproximadas e técnicas que nos dão geralmente uma *noção do que fazer* — ficando a decisão de ir mais adiante por conta do especialista, e principalmente do investidor. A idéia da solução mais apropriada, está longe, em muitos aspectos, do ótimo formal que nos acostumamos a conhecer da Programação Matemática. Técnicas de Inteligência Artificial na solução de problemas de *Scheduling* aplicado ao controle de tráfego de trens, FONSECA NETO (1990), são uma demonstração clara da readaptação de um modelo matemático em função de um momento decisório. A dinâmica do processo, parece, como vemos, dar sempre uma nova face ao problema focado.

Em virtude do que expomos acima, acreditamos que seria justo elaborar uma técnica que tratasse o PCA do ponto de vista do lixo, especificamente resolvendo o modelo proposto como PCCLU, já colocado e discutido.

A idéia do algoritmo MARCOLIX surgiu ao praticarmos, experimentalmente o roteiramento em trechos de coleta em Fortaleza, NEGREIROS ET AL (1990). Neste trabalho, por não termos recursos computacionais que nos permitissem resolver por algum método conhecido o problema da coleta de lixo; procuramos fazê-lo praticamente à mão, adotando critérios de escolha de visitação, determinando áreas prioritárias, e mantendo ao máximo uma atividade de serviço que cumprisse as restrições impostas pelo modelo. Os resultados obtidos podem ser tomados como expressivos em relação ao praticado, Tabela IV.1. Isto foi-nos suficiente para tomarmos o impulso necessário e em pouco tempo escrever o MARCOLIX.

O algoritmo MARCOLIX conta, então, com as idéias de ORLOFF

1a. Viagem	Praticado	Plano	$\Delta\%$
Km	15.000	9.400	37.3 [-]
Ton	6.900	7.800	13.0 [+]
Tempo (H)	4.070	3.120	23.3 [-]
Ton/Km	0.460	0.830	80.4 [+]
Ton/Km-H	0.113	0.266	135.4 [+]

2a. Viagem	Praticado	Plano	$\Delta\%$
Km	10.600	9.800	7.6 [-]
Ton	6.100	7.900	29.5 [+]
Tempo (H)	3.510	3.270	6.8 [-]
Ton/Km	0.570	0.810	40.9 [+]
Ton/Km-H	0.166	0.310	86.8 [+]

Global	Praticado	Plano	$\Delta\%$
Viagens	> 2	2	
Km	25.500	19.200	24.7 [-]
Ton	13.200	15.700	18.9 [+]
Tempo (H)	7.580	6.390	18.6 [-]
Ton/Km	0.518	0.818	57.9 [+]
Ton/Km-H	0.068	0.128	88.2 [+]
Balanceamento	1.130	0.990	

Tabela IV.1: Comparação entre os índices médios praticados e os índices médios do plano de roteiramento no trecho 007-004 – Fortaleza, Abril/1990.

(1973] e os critérios de roteamento de CHAPLEAU ET AL (1981), baseado nas distâncias entre o ponto de partida e o ponto de chegada. Aliado ao algoritmo de caminho mínimo de DIJKSTRA (1959). No algoritmo MARCOLIX se admite que todo elemento de demanda solicita carga e menos tempo de serviço que os limites impostos aos veículos e tripulação. O algoritmo MARCOLIX pode ser considerado como um algoritmo do tipo *guloso - gridy*, pois enquanto vai caminhando em busca de novos elemento a servir ele já vai construindo a rota definitiva. Passaremos, a seguir, a detalhá-lo.

Considere o seguinte conjunto de variáveis:

G — Rede original, $G = (V, E, A)$

\mathcal{G} — Rede modificada;

$Pi1$ — Conjunto de vértices a visitar;

PHp — Conjunto ordenado pela distância euclidiana de referência dos vértices a visitar;

T — Conjunto de vértices do trecho a servir;

NVg — Número de viagens a realizar;

NVr — Número de viagens cumpridas;

NNs — Número de vértices servidos;

NNp — Número de vértices de $Pi1$ e PHp ;

NNr — Número de vértices na rota;

NNt — Número de vértices totais;

Qv — Carga nominal do veículo;

Qac — Carga acumulada na viagem;

Qc — Carga do vértice candidato;

TSc — Tempo de serviço do vértice candidato;

TSt — Tempo da carga de trabalho da tripulação;

Tp — Tempo de parada para Pesagem/Descarrego

Tap — Tempo de aproveitamento de serviço;

TR — Tempo de retorno ao ponto de despejo pelo caminho mais rápido.

T_0 — Tempo no vértice 0 — Relativo ao rótulo do caminho tomado na distância entre dois pontos de \mathcal{G} ;

T_x — Tempo no vértice x — Relativo ao rótulo do caminho tomado na distância entre dois pontos de \mathcal{G} , sucessor de 0;

T_c — Tempo para sair de c ao local de despejo pelo caminho mais rápido;

d_{G_i} — Distância mais rápida entre a garagem e o i -ésimo elemento a servir;

d_{D_i} — Distância mais rápida entre local de despejo e o i -ésimo elemento a servir;

d_{iD} — Distância mais rápida entre i -ésimo elemento a servir e o local de despejo;

PTOBS — Ponto de observação — vértice da rede original;

NoAt — Índice do vértice candidato ao serviço;

NoAt — Irmão de NoAt (Caso mão dupla);

Início — Vértice correspondente à garagem;

Despejo — Vértice correspondente ao local de despejo;

$Rota_v$ — Seqüência de visitação dos vértices;

$Rota_A$ — Atividade de serviço da seqüência $Rota_v$;

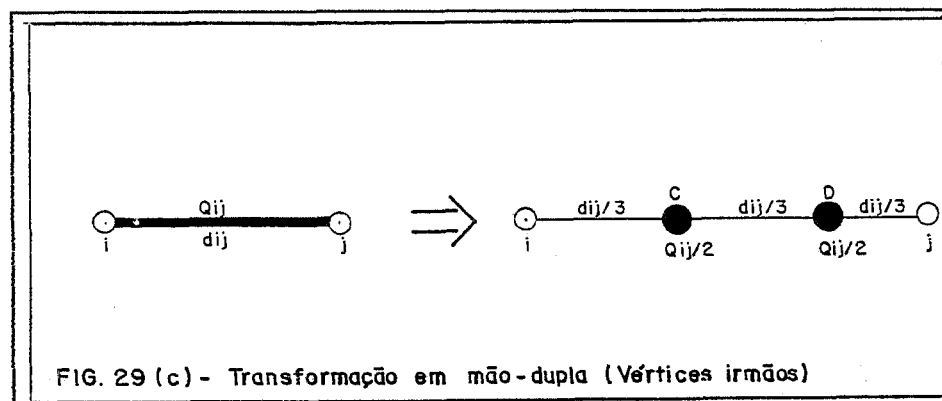
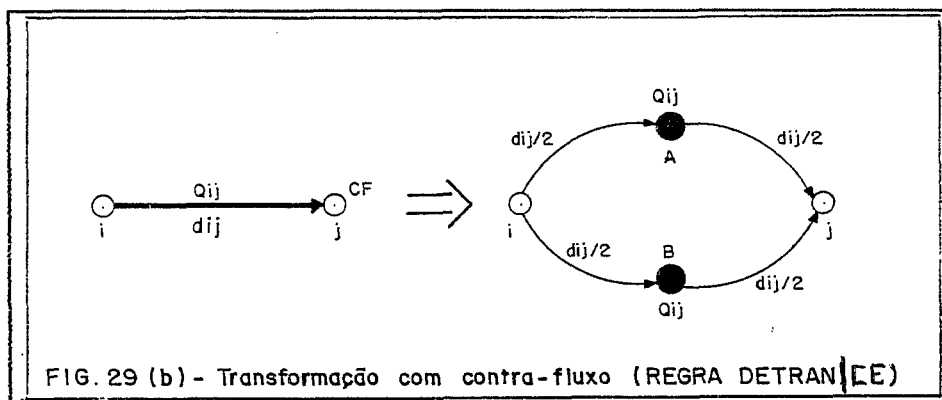
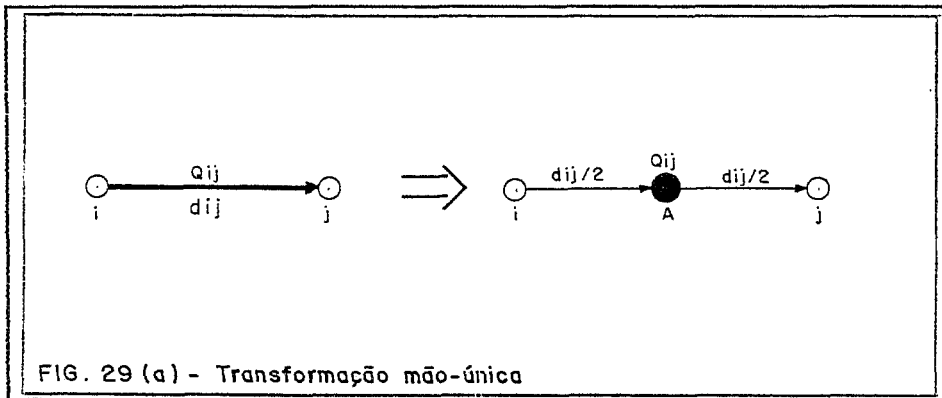
Exemplo:

- $Rota_v = \{ 1 \ 3 \ 5 \ 7 \ 9 \ 10 \}$
- $Rota_A = \{ F \ V \ V \ F \ V \ F \}$

Passo 0:

Dada uma rede viária \mathcal{G} , identifique os arcos/elos com demanda e quebre-os utilizando o seguinte critério, Fig. 29(a, b, c):

- *Ruas de Mão-Única:* Defina um nó de \mathcal{G} no centro de cada quarteirão;
- *Ruas com Contra-Fluxo:* Crie dois lados de serviço conforme as regras de trânsito em cada lado, e para cada lado mão-única, defina um nó de \mathcal{G} no centro destes arcos;
- *Ruas de Mão-Dupla:* Defina dois nós de \mathcal{G} a um terço e dois terços de cada quarteirão;



- Dada uma rede viária G , identifique todos os elementos de serviço associados, acoplado ao trecho de trabalho requerido.

Passo 1:

Determinar as distâncias dos elementos a visitar em T , da seguinte forma:

$$\begin{aligned} d'_i &\leftarrow d_{Gi} + d_{iD} \\ d''_i &\leftarrow d_{Di} + d_{iD} \end{aligned}$$

Passo 2:

Determinar o número de viagens e a carga por viagem do veículo alocado ao serviço, do seguinte modo:

Caso Balanceado:

$$NVg \leftarrow \lceil \sum_{c \in T} Qc/Qv \rceil \quad (IV.15)$$

$$Qvg \leftarrow \sum_{c \in T} Qc/NVg \quad (IV.16)$$

Caso Não-Balanceado:

$$Qvg \leftarrow Qv \quad (IV.17)$$

$$NVg \leftarrow \sum_{c \in T} Qc/Qv \quad (IV.18)$$

Passo 3:

Determine a pilha $Pi1$, e gere $PiHp$ conforme a posição relativa de cada vértice a servir da origem e despejo.

Passo 4:

Enquanto $(NVr \neq NVg)$ ou $(NNs \neq NNt)$ faça

- $NoAt \leftarrow PiHp[1];$
- Enquanto Fim = F. faça
- Se $NoAt = PiHp[1]$ então
- Execute $APROVEITA(Tap);$
- Senão $Tap = 0;$
- Verifique o tipo do Candidato que solicita serviço
- 1. Cliente ou Ponto de Coleta;
- 2. Mão-única ou contra-fluxo;
- 3. Mão-dupla.
- Se Candidato = 1 ou 2 então

Se $(T_{Sa} + TR + T_{ap} + T_p + T_{Sc} \leq T_{St})$ e $(Q_{ac} + Q_c \leq Q_{ug})$

Então

$T_{Sa} \leftarrow T_{Sa} + T_{ap} + T_{Sc};$

$Q_{ac} \leftarrow Q_{ac} + Q_c;$

$P_{i1} \leftarrow P_{i1} - \{NoAt\};$

$P_{iHp} \leftarrow P_{iHp} - \{NoAt\};$

$NN_s \leftarrow NN_s + 1;$

$NN_r \leftarrow NN_r + 1;$

$Rota_v[NN_r] \leftarrow NoAt$

$Rota_A[NN_r] \leftarrow .V.$

$NN_p \leftarrow NN_p - 1;$

Se $NN_p \neq 0$ então

Execute NOVOPTOBS

Senão $Fim \leftarrow .V.$

Senão $Fim \leftarrow .V.$

Se Candidato = 3 então

Se $(T_{Sa} + TR + T_{ap} + 2 T_{Sc} + T_p \leq T_{St})$ e $(Q_{ac} + 2 Q_c \leq Q_{ug})$

Então

$T_{Sa} \leftarrow (T_{Sa} + 2 T_{Sc});$

$Q_{ac} \leftarrow (Q_{ac} + 2 Q_c);$

Verifique o vértice mais distante de PTOBS, se

$NoAt$ ou \underline{NoAt} ;

Caso \underline{NoAt} seja o mais distante,

$Ind1 \leftarrow \underline{NoAt}$ e $Ind2 \leftarrow NoAt$

Senão $Ind1 \leftarrow NoAt$ e $Ind2 \leftarrow \underline{NoAt}$;

$P_{i1} \leftarrow P_{i1} - \{NoAt, \underline{NoAt}\};$

$P_{iHp} \leftarrow P_{iHp} - \{NoAt, \underline{NoAt}\};$

$NN_s \leftarrow NN_s + 2;$

$NN_p \leftarrow NN_p - 2;$

$NN_r \leftarrow NN_r + 1;$

$Rota_v[NN_r] \leftarrow Ind1$

$Rota_A[NN_r] \leftarrow .V.$

$NN_r \leftarrow NN_r + 1;$

$Rota_v[NN_r] \leftarrow Ind2$

$Rota_A[NN_r] \leftarrow .V.$

Se $NN_p \neq 0$ então

Execute NOVOPTOBS

Senão $Fim \leftarrow .V.$

Senão $Fim \leftarrow .V.$

$NN_r \leftarrow NN_r + 1;$

$Rota_v[NN_r] \leftarrow Despejo$

$Rota_A[NN_r] \leftarrow .F.$

Determine o caminho mais rápido de $Rota_v[NN_r - 1]$ até $Rota_v[NN_r]$, e contabilize o tempo de viagem em T_{Sa} ;

$NV_g \leftarrow NV_g + 1;$

$Fim \leftarrow .F.$

Se *Inicio* \neq *Despejo* então

. Reordene *PiHp* conforme a posição relativa ao despejo

PTOBS \leftarrow *Despejo*

Inicio \leftarrow *Despejo*

TSa \leftarrow *TSa* + *Tp*

Rotina APROVEITA ;

Esta rotina tem o objetivo de aproveitar os vértices que estão entre o *PTOBS* e *NoAt*, pelo caminho mais rápido. Nesse procedimento, também se calcula o tempo de passagem sem serviço.

Determine o caminho mais rápido do *PTOBS* ao *NoAt*;

X \leftarrow *Suces(PTOBS)*;

Tap \leftarrow 0; *T₀* \leftarrow 0;

Tz \leftarrow *T(x)*;

Nv \leftarrow *.F.*;

Enquanto (*X* \neq *NoAt*) ou (*Fim* = *.F.*) faça

. Se *X* \in *Pi1* então

. Determine qual o tipo de *X* - (1,2 ou 3);

. *Nv* \leftarrow *.V.*

. Se tipo de *X* = 1 ou 2 então

. Se (*TSa* + *Tp* + *TR* + *Tap* + *TSc* \leq *TSt*) e (*Qac* + *Qc* \leq *Qvg*)

. Então

. *TSa* \leftarrow *TSa* + *Tap* + *TSc*;

. *Tap* \leftarrow 0;

. *Qac* \leftarrow *Qac* + *Qc*;

. *Pi1* \leftarrow *Pi1* - {*X*};

. *PiHp* \leftarrow *PiHp* - {*X*};

. *NNs* \leftarrow *NNs* + 1;

. *NNr* \leftarrow *NNr* + 1;

. *Rota_v*[*NNr*] \leftarrow *X*

. *Rota_A*[*NNr*] \leftarrow *.V.*

. *NNp* \leftarrow *NNp* - 1;

. Se *NNp* = 0 então

. *Fim* \leftarrow *.V.*

. Se tipo de *X* = 3 então

. Se (*TSa* + *TR* + *Tap* + *TSc* + *Tp* \leq *TSt*) e (*Qac* + 2 *Qc* \leq *Qvg*)

. Então

. *TSa* \leftarrow (*TSa* + 2 *TSc*);

. *Tap* \leftarrow 0;

. *Qac* \leftarrow (*Qac* + 2 *Qc*);

. *Pi1* \leftarrow *Pi1* - {*X*, *Suces(X)*};

. *PiHp* \leftarrow *PiHp* - {*X*, *Suces(X)*};

. *X* \leftarrow *Suces(X)*

. *NNs* \leftarrow *NNs* + 2;

. *NNr* \leftarrow *NNr* + 1;

. *Rota_v*[*NNr*] \leftarrow *X*

. *Rota_A*[*NNr*] \leftarrow *.V.*

```

.       $NNr \leftarrow NNr + 1;$ 
.       $Rota_v[NNr] \leftarrow Suces(X);$ 
.       $Rota_A[NNr] \leftarrow .V.$ 
.      Se  $NNp = 0$  então
.           $Fim \leftarrow .V.$ 
.
Se  $Nv \langle \rangle .V.$  então  $Tap \leftarrow Tap + Tx - T_0;$ 
 $T_0 \leftarrow Tx;$ 
 $X \leftarrow Suces(X)$ 
 $Nv \leftarrow .F.$ 

```

Rotina NOVOPTOBS

Esta rotina procura manter o veículo em serviço contínuo ao máximo, e caso não consiga ou dá espaço para um novo NoAt ou termina.

Passo 0:

Determinar o elemento adjacente ao NoAt que é cruzamento ou vértice da rede primária. Chame-o de PTOBS e coloque-o na rota, caso este não pertença em Pi1.

Caso PTOBS \in Pi1, então

```

.      Se  $(TSa + Tp + TSc \leq TSt)$  e  $(Qac + Qc \leq Qvg)$ 
.      Então
.           $TSa \leftarrow TSa + TSc;$ 
.           $Qac \leftarrow Qac + Qc;$ 
.           $Pi1 \leftarrow Pi1 - \{PTOBS\};$ 
.           $PiHp \leftarrow PiHp - \{PTOBS\};$ 
.           $NNs \leftarrow NNs + 1;$ 
.           $NNr \leftarrow NNr + 1;$ 
.           $Rota_v \leftarrow NoAt$ 
.           $Rota_A \leftarrow .V.$ 
.           $NNp \leftarrow NNp - 1;$ 
.          Se  $NNp = 0$  então
.               $Fim \leftarrow .V.$ 
.          Senão  $Fim \leftarrow .V.$ 

```

Passo 1:

De PTOBS, verifique qual dos vértices adjacentes está em Pi1 e também está mais próximo (em termos de distância) do ponto de partida e chegada. Chame-o de NoAt caso exista, e caso contrário $NoAt \leftarrow PiHp[1]$.

Caso NoAt \in G então Execute NOVOPTOBS;

Teorema: A heurística MARCOLIX possui complexidade computacional de ordem $O(n^2 \log 2k)$, onde n é o número de vértices de G e k é o número de candidatos de G a serem servidos.

Prova

Admitindo que o candidato de Pi1[1] a ser servido é o elemento de G

mais distante do ponto de partida e que em seu caminho estão todos os elementos a servir. Tomando o algoritmo de DIJKSTRA (1959), como o esforço computacional deste algoritmo de caminho mínimo entre dois pontos de uma rede é $O(n^2)$, e para cada elemento retirado de $P_i l$ e $P_i H p$ é necessário, no pior caso, um esforço de $O(\log 2k)$ operações. Então o algoritmo MARCOLIX é $O(n^2 \log 2k)$.
c.q.d

IV.3.1 Observações sobre o algoritmo MARCOLIX

A heurística MARCOLIX, foi elaborada para resolver tanto um problema que considera uma coleta balanceada, quanto uma coleta não balanceada. Tais considerações permitem, na prática, que o gerente possa decidir pelo feitio de rotas que mantenham ou não uma grande quantidade de serviço contínuo durante a coleta. Na verdade, quando as zonas de trabalho são bastante distantes dos pontos de partida e despejo, notamos que a influência de um balanceamento pode não ser tão relevante do ponto de vista da coleta. Por outro lado, o balanceamento garantiria sempre uma menor depreciação dos veículos, pois os desgastes serão bem reduzidos.

Como podemos notar, também, o veículo procura realizar o seu serviço andando sempre pelo percurso mais rápido e não o mais curto, e o motivo disto é relativo ao período de trabalho da tripulação que não pode ser excedido. Nota-se, na verdade, que se deseja realizar a tarefa de coleta no menor tempo possível, de modo a reduzir o desgaste da tripulação no trabalho.

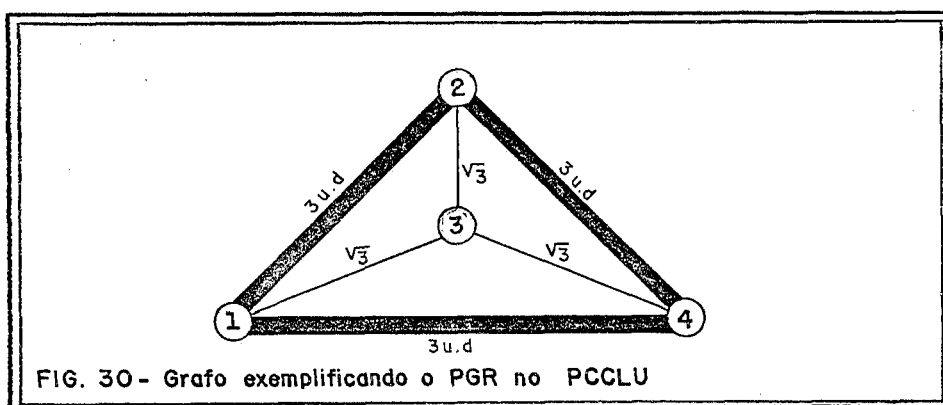
É importante visualizar que o MARCOLIX procura manter-se em serviço o máximo de tempo possível, e somente realiza sua tarefa caso seja possível fazê-lo. Durante o processo de geração da pilha de elementos a servir - $P_i l$, adotamos critérios de distância euclidiana que não levam em consideração a carga de serviço. Os critérios de distância podem ser visualizados conforme a posição relativa do local de despejo e garagem.

A configuração das rotas pode mudar substancialmente quando ajustamos preferencialmente um parâmetro ou critério de seleção de um novo candidato na rotina NOVPTOBS, onde podemos piorar incontrolavelmente os valores da rota. Os ajustes finais podem ser feitos pelo especialista, caso seja esse o problema, e o MARCOLIX poderá servir como uma poderosa ferramenta na solução criteriosa do modelo.

IV.3.2 Exemplos de Roteamento Utilizando o Algoritmo MARCOLIX

Várias situações podem ser vistas na aplicação da resolução do problema proposto. Seleccionamos alguns exemplos para a melhor compreensão do MARCOLIX, que a seguir desenvolvemos:

Exemplo (a): Admita o grafo da Fig. 30, onde cada elo em linhas grossas demanda 2 u.c e cada vértice *escurecido*, demanda 1 u.c. Admita também que o veículo possa servir toda a demanda com tempo de trabalho ilimitado.

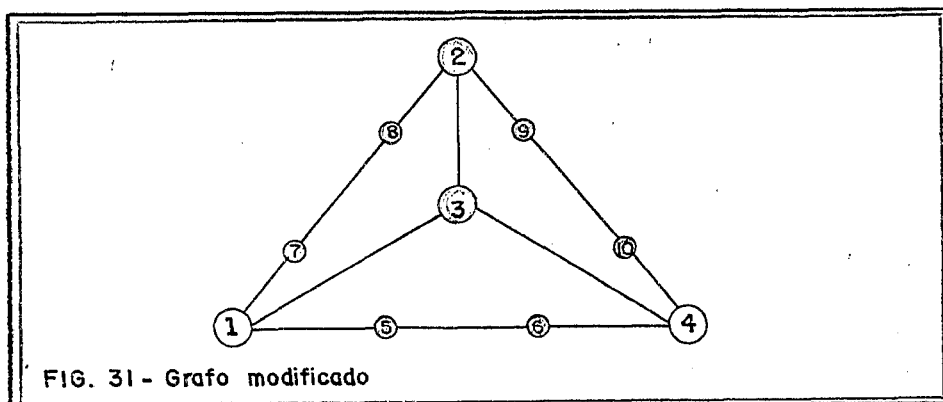


Determinaremos:

1. A rota de serviço partindo de (1) e chegando em (4)

$P_i1 - \{ (2) \ 8 \ 9 \ 10 \ 7 \ (3) \ 5 \ 6 \}$

$P_iH_p - \{ 6 \ 6 \ 6 \ 6 \ 5 \ 2\sqrt{3} \ 3 \ 3 \}$



Rota final:

$\{ (1) \leftrightarrow 7 \leftrightarrow 8 \leftrightarrow (2) \leftrightarrow 9 \leftrightarrow 10 \leftrightarrow (4) \leftrightarrow (3) \leftrightarrow (1) \leftrightarrow 5 \leftrightarrow 6 \leftrightarrow (4) \}$

Comprimento: $9 + 2\sqrt{3}$

2. Rota de serviço saindo de (1) e retornando a (1)

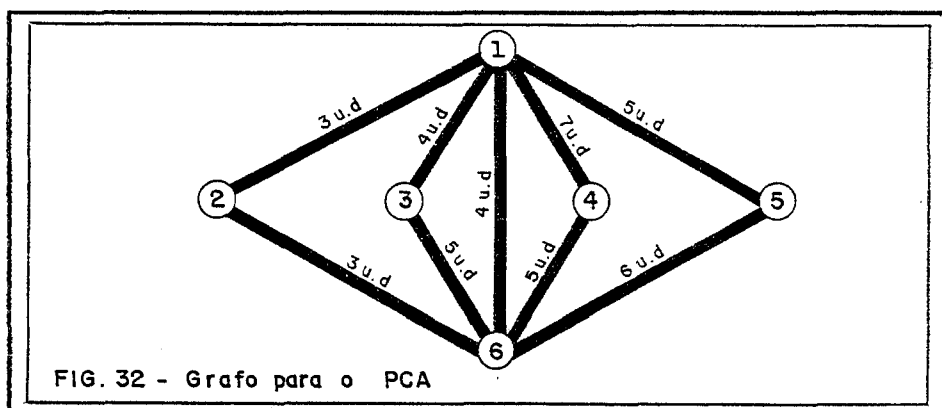
$Pi1 - \{ 9 \ 10 \ (2) \ 6 \ 8 \ (3) \ 5 \ 7 \}$
 $PiHp - \{ 8 \ 8 \ 6 \ 4 \ 4 \ 2\sqrt{3} \ 2 \ 2 \}$

Rota final:

$\{ (1) \leftrightarrow 7 \leftrightarrow 8 \leftrightarrow (2) \leftrightarrow 9 \leftrightarrow 10 \leftrightarrow (4) \leftrightarrow (3) \leftrightarrow (1) \leftrightarrow 5 \leftrightarrow 6 \leftrightarrow$
 $(4) \leftrightarrow (1) \}$

Comprimento: $12 + 2\sqrt{3}$

Exemplo (b): Tome o grafo da Fig. 32, modificando o MARCOLIX para que uma viagem seja completada a cada vez que o veículo retornar a (1). Considere que cada arco possui 1 u.c, e que o veículo possua carga máxima de 4 u.c.
 - Problema proposto por GOLDEN & WONG (1981).



$Pi1 - \{ 22 \ 24 \ 23 \ 19 \ 20 \ 21 \ 10 \ 13 \ 16 \ 12 \ 15 \ 9 \ 18 \ 11 \ 14 \ 17 \ 7 \ 8 \}$

$PiHp - \{ 44/3 \ 42/3 \ 36/3 \ 34/3 \ 34/3 \ 34/3 \ 20/3 \ 16/3 \ 16/3 \ 14/3 \ 14/3 \ 10/3 \ 10/3 \ 8/3 \ 8/3 \ 8/3 \ 4/3 \ 2/3 \}$

1a. Viagem:

Rota final:

$\{(1) \leftrightarrow 14 \leftrightarrow 16 \leftrightarrow (6) \leftrightarrow 21 \leftrightarrow 22 \leftrightarrow (4) \leftrightarrow 15 \leftrightarrow 12 \leftrightarrow (1)\}$

Comprimento: 16 u.d

Carga: 3 u.c

2a. Viagem:

Rota final:

$\{(1) \leftrightarrow 9 \leftrightarrow 10 \leftrightarrow (5) \leftrightarrow 24 \leftrightarrow 23 \leftrightarrow (6) \leftrightarrow 20 \leftrightarrow 19 \leftrightarrow (3) \leftrightarrow 12 \leftrightarrow$
 $11 \leftrightarrow (1)\}$

Comprimento: 10 u.d

Carga: 4 u.c

3a. Viagem:

Rota final:

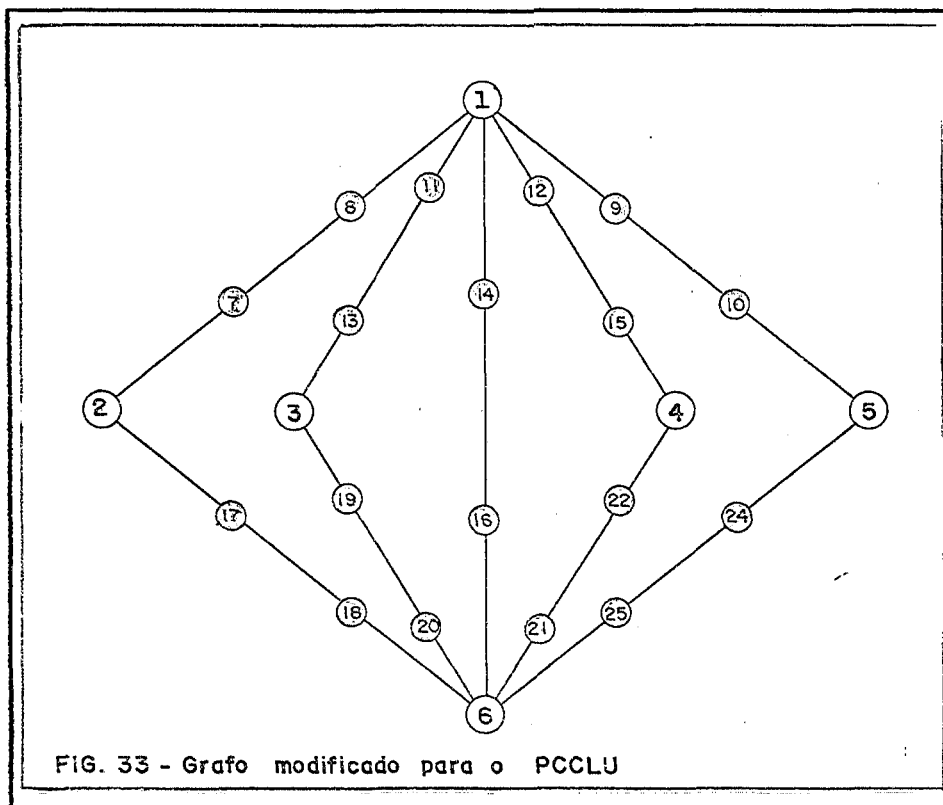
$\{(1) \leftrightarrow 8 \leftrightarrow 7 \leftrightarrow (2) \leftrightarrow 17 \leftrightarrow 18 \leftrightarrow (6) \leftrightarrow (1)\}$

Comprimento: 10 u.d

Carga: 2 u.c

Comprimento Total: 46 u.d

Mostramos na Fig. 34, um exemplo típico de situação de coleta em uma grande cidade, onde procuramos reproduzir todas as situações de coleta do lixo



domiciliar.

Na operação do S e R o t , cabe ao usuário determinar qual medida deve ser tomada em caso de quebra de serviço por extrapolação no balanceamento de carga. Também deixamos a critério do gerente a opção ou não pelo balanceamento, e mostramos o que isso acarretaria no caso comparativo (Ver Tabela IV.2). Não obstante, também procuramos dotar a heurística de critérios de distância para os candidatos de forma diferenciada.

Na Fig. 34, notamos a existência de ruas de mão-única, com contra-fluxo, pontos especiais de coleta e ruas de mão-dupla. Em cada quadra, podemos encontrar os valores de $[a, b)$, onde a é a distância ou comprimento do quarteirão e b é a carga estimada de coleta. Já para os Pontos de Coleta e Pontos Especiais de Coleta, temos o tempo de serviço e a sua carga estimada. Na Tabela IV.3, podemos encontrar as velocidades médias de operação para cada tipo de via e a classificação das vias da Fig. 34.

Quatro formas de abordar o algoritmo MARCOLIX foram usadas para resolver o mesmo problema que parte da garagem - DLU, e o local de despejo é o ATERRO (JANGURUSSU). Na primeira e terceira, não há mudança nos critérios entre viagens, sendo observado os candidatos mais distantes e mais próximos dos locais de partida e chegada, respectivamente. Na segunda e quarta, optamos em abordar os critérios variando conforme a posição relativa do local de partida. Nas Tabelas IV.3 e IV.4, vemos os resultados obtidos para o problema exemplo proposto.

Nas páginas a seguir, mostramos o relatório de saída para o motorista que deverá cumprir o roteiro de serviço para a região de estudo especificada, relativo

Via / Vel. (Km/h)	Primária	Secundária
Coleta	2.9	2.9
Vazio	30.0	20.0
Cheio	22.0	17.0

Tabela IV.2: Dados de movimentação contínua dos veículos coletores para o exemplo teste.

- Vias Primárias: Av. Aguanambi, BR-116, Av. Santos Dumont, Av. Anto. Sales (CF) e Av. Barão de Studart;
- Vias Secundárias: Silva Jataí, R. Idelfonso Albano e R. Antonele Bezerra;
- Coleta Especial: Shopping Center Riviera (Silva Jataí com Anto. Sales);
- Coleta Pontual: Vila São Paulo.

BALANCEADO	VDD	VDP	VPP	VPD
Qv (Ton)	2.0	2.0	2.0	2.0
TSt (Hs)	2.0	2.0	2.0	2.0
NVg	2	2	2	2
Km	5.31	5.26	5.59	5.28
TSa	1h 51min	1h 50min	1h 45min	1h 50min
% Servido	100.00	100.00	100.00	100.00
Ton/Km-Hr (Min)	0.27	0.28	0.27	0.28
Ton/Km-Hr (Max)	0.36	0.36	0.35	0.36
TPr (Seg)	6.04	3.01	7.98	6.62

Tabela IV.3: Exemplos executados para o caso balanceado, na rede exemplo de situação.

NAO BALANCEADO	VDD	VDP	VPP	VPD
Qv (Ton)	2.0	2.0	2.0	2.0
TSt (Hs)	2.0	2.0	2.0	2.0
NVg	2	2	2	2
Km	5.55	5.65	6.66	6.11
TSa	1h 52min	1h 51min	1h 54min	1h 53min
% Servido	100.00	100.00	100.00	100.00
Ton/Km-h (Min)	0.26	0.26	0.21	0.23
Ton/Km-h (Max)	0.34	0.33	0.28	0.30
TPr (Seg)	6.02	7.17	8.05	7.04

Tabela IV.4: Exemplos executados para o caso não balanceado, na rede exemplo de situação.

Considere TPr o tempo de processamento do algoritmo em um microcomputador do tipo PC/AT-286, 12 MHz com co-processador aritmético 80286.

ao serviço de menor custo encontrado nas abordagens.

COPE/UFRJ - OTIMIZACAO
DEPARTAMENTO DE LIMPEZA URBANA (DLU)

Hora: 14:17.44
Data: 18/9/1990

Veículo: T0001 Carga Nominal (Ton): 2.00 Pag.: 2
Início da Rota: Aterro Sanitario Fim da Rota: Aterro Sanitario
No. de Garis: 3 Tempo Maximo de Service: 2hs 10min.

ROTEIRO DE SERVIÇO NO CIRCUITO 001-000/ 2a. Viagem

... Siga na BR. 116 ate R. ANTONEL BEZERRA
(*) Coleta na R. ANTONEL BEZERRA ate R. IDELFONSO ALBANO
(**) Passe sem coletar na R. IDELFONSO ALBANO ate VILA SAO PAULO
***** ESTACIONE O VEICULO E COLETE NA VILA SAO PAULO
(**) Passe sem coletar na R. IDELFONSO ALBANO ate AV. SANTOS DUMONT
(*) Coleta na AV. SANTOS DUMONT ate R. BARAO DE ARACATI
(*) Coleta na R. BARAO DE ARACATI ate R. ANTONEL BEZERRA
(**) Passe sem coletar na R. ANTONEL BEZERRA ate R. IDELFONSO ALBANO
(**) Passe sem coletar na R. ANTONEL BEZERRA ate D.L.U. (GARAGEM)
(**) Passe sem coletar na R. SILVA JATAI ate AV. SANTOS DUMONT
(*) Coleta na AV. SANTOS DUMONT ate R. IDELFONSO ALBANO
(*) Coleta na R. IDELFONSO ALBANO ate AV. ANTO. SALES
(*) Coleta na AV. ANTO. SALES ate R. BARAO DE ARACATI
(**) Passe sem coletar na R. BARAO DE ARACATI ate AV. SANTOS DUMONT
(**) Passe sem coletar na R. BARAO DE ARACATI ate R. ANTONEL BEZERRA
(**) Passe sem coletar na R. ANTONEL BEZERRA ate R. IDELFONSO ALBANO
(**) Passe sem coletar na R. ANTONEL BEZERRA ate D.L.U. (GARAGEM)
(**) Passe sem coletar na R. SILVA JATAI ate AV. SANTOS DUMONT
(**) Passe sem coletar na R. SILVA JATAI ate AV. ANTO. SALES
(*) Coleta na AV. ANTO. SALES ate R. IDELFONSO ALBANO
... Siga na AV. ANTO. SALES ate R. BARAO DE ARACATI
... Siga na R. BARAO DE ARACATI ate AV. SANTOS DUMONT
... Siga na AV. SANTOS DUMONT ate ATERRO (JANGURUSSU)
(*) Pesagem/Descarrego.
Percezridos -> 3.795 Km
Completados -> 100.00% de total a coletar.
Carga (Min) -> 1.19 Ton (Max) -> 1.99 Ton
Tempo Estimado -> 1hs 50min.

DADOS GLOBAIS

(*) Numero de Viagens: 2
(*) Distancia Total percorrida: 5.255 Km
(*) Tempo de Service Estimado: 1hs 50min.
(*) Lixo Coletado (X): 100.00

** Previsao de Carga para Coleta **
(Min) -> 2.30 Ton (Max) -> 3.90 Ton

** Rendimento da Coleta **
(Min) -> 0.24 Ton/Km-hr (Max) -> 0.40 Ton/Km-hr

(*) Tempo de Processamento: 0h 0min 21.120

Como podemos notar nas Tabelas IV.3 e IV.4, o exemplo nos mostra que é bem mais sensato a utilização de processo balanceado, mesmo que não levemos em conta que os veículos tenham uma queda de depreciação por utilização irracional de suas potencialidades.

A estrutura da malha viária que mostramos no exemplo, coloca o local de início, próximo ao trecho de serviço, por que não dizer exatamente no local de serviço. Aparentemente, esta forma de caminhar pode não ser a melhor, pois não leva em conta um posicionamento da garagem em um local longe do trecho de

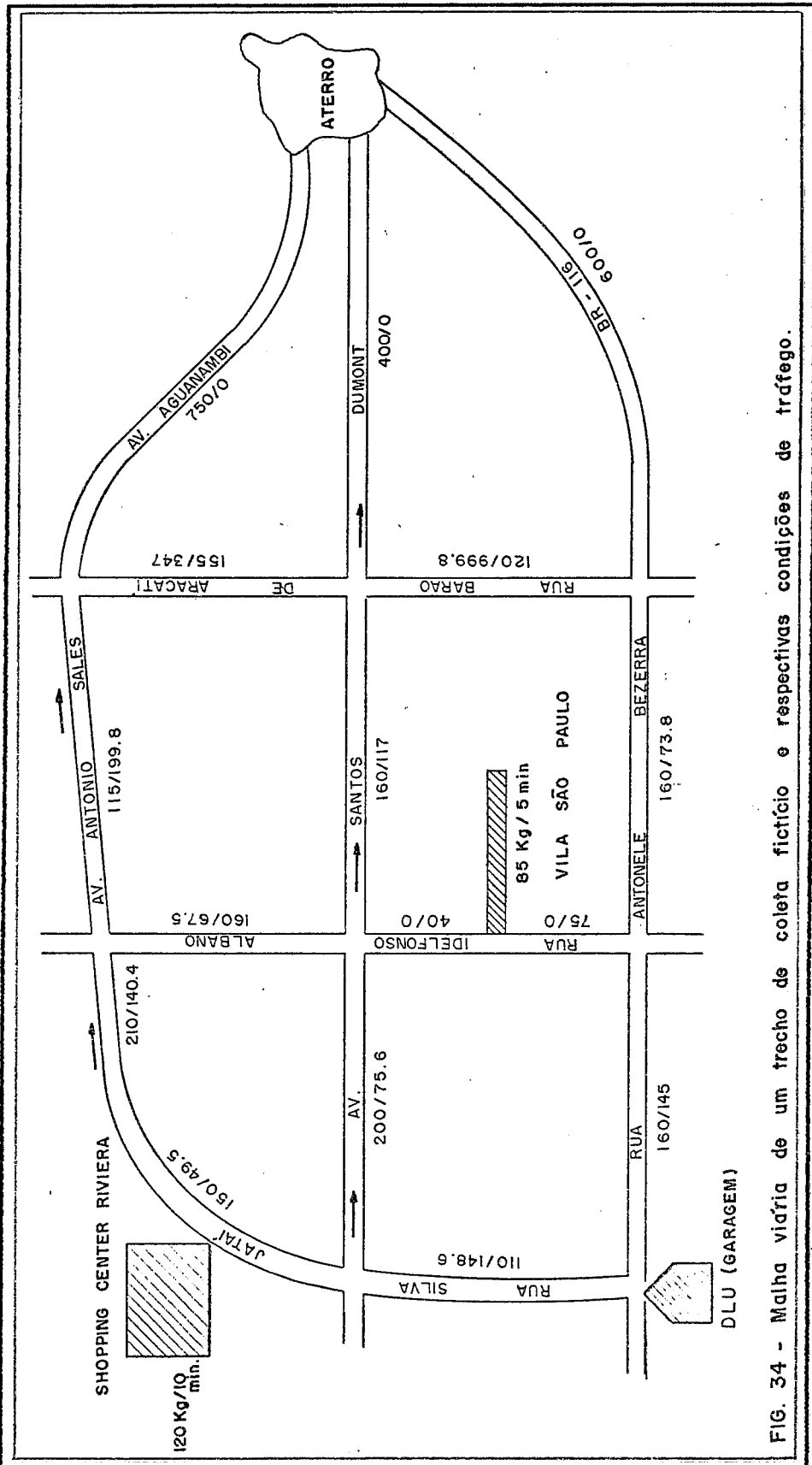
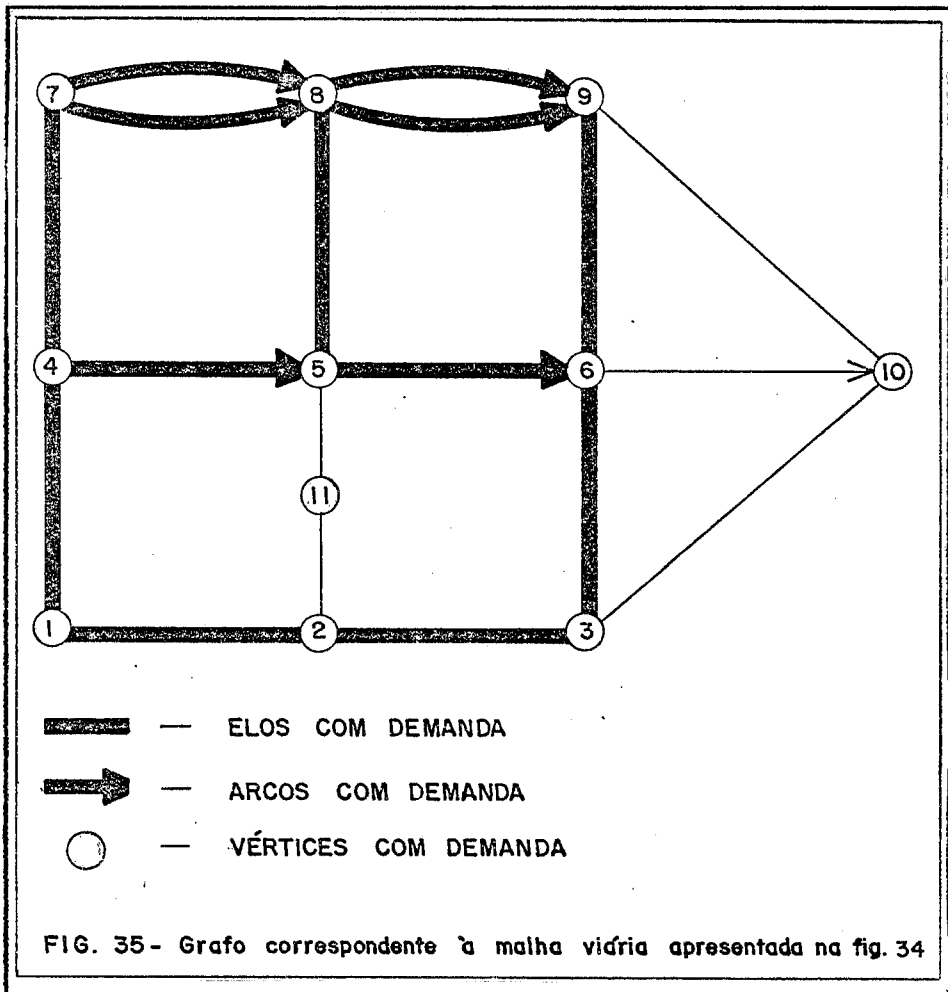


FIG. 34 - Malha viária de um trecho de coleta fictício e respectivas condições de tráfego.

coleta. No trabalho de campo realizado em Fortaleza, a garagem é bem distante do local de despejo e dos trechos de serviço como na Fig. 28. No APÊNDICE 4, podemos verificar nos diversos exemplos, a contribuição da distância relativa e os tempos de processamento em vários casos testes, tanto para o exemplo da Fig. 34 quanto para o caso de Fortaleza.



IV.4 Conclusão

No trabalho que aqui realizamos, procuramos desenvolver um sistema computacional capaz de resolver o Problema da Coleta do Lixo Urbano, como foi formulado, levando-se em conta as atividades de coleta e remoção.

Apesar de computacionalmente difícil de ser implementado, e do pouco recurso que dispusemos na complementação do nosso trabalho, acreditamos que o algoritmo MARCOLIX pode dar boas economias se tomarmos como base a prática realizada atualmente, além do que, tem a facilidade de permitir que o usuário se liberte um pouco da impropriedade de ter que fazer o seu planejamento sem um critério de otimização razoável.

Ainda muito há por fazer em termos de melhoria de *desempenho*. No algoritmo MARCOLIX, por exemplo, podemos modificar alguns de seus parâmetros para definir uma frota de acordo com os tipos de veículos especificados pelo comércio, como no trabalho de ULUSOY (1985). Ainda mais, podemos expandí-lo para trechos que possuam a mesma área das cidades, com o uso apropriado de algoritmos de caminho mínimo, como o algoritmo de GLOVER (1989), atualmente o *estado-da-arte* dos algoritmos de caminho mínimo. Também podemos reduzir o número de vértices da rede G , e conseqüentemente melhorar o tempo de execução do algoritmo MARCOLIX.

Tais trabalhos poderiam ser implementados em um computador de maior porte, que não os microcomputadores da linha PC que utilizamos, haja vista uma melhoria considerável no tempo de resposta do algoritmo.

Acreditamos que atingimos nossos objetivos e que procuramos ser o mais sucintos possível na elaboração de um trabalho que tivesse uma direção mais realista na solução do problema da coleta do lixo. Temos, pois a certeza de que contribuimos com uma solução a um dos graves problemas da sociedade brasileira, e que de imediato seus benefícios podem ser utilizados.

Capítulo V

Conclusão

Este trabalho constou de duas partes, a primeira procurando abordar o problema de roteiramento de veículos, desde a sua formalização até os modos de solução para os casos específicos em estudo, o problema de roteiramento na coleta de lixo domiciliar. Já na segunda, colocamos o aspecto real da coleta do lixo e propomos uma solução para o Problema de Planejamento de Frotas e Percurso de Veículos na Coleta do Lixo Urbano Domiciliar, utilizando técnicas adequadas.

Discutimos apropriadamente, a maioria das abordagens do problema do Carteiro Chinês, e procuramos contribuir com um estudo detalhado do Problema do Carteiro Chinês Misto, onde analisamos pormenores do trabalho de FREDERICKSON (1979), o qual consideramos o melhor dentre os dois pesquisados.

Haja vista a realidade do problema do lixo estar intimamente relacionada à demanda de lixo em quarteirões (arcos e elos) de uma rede viária, foi importante enunciar o Problema de Roteamento Capacitado em Arcos (PRCA), proposto por GOLDEN & WONG (1981), e discutir um pouco a sua generalidade em relação aos problemas de roteamento em arcos e nós, como também sua complexidade e heurísticas mais conhecidas para a sua solução. Dentre todos os trabalhos que estudamos, selecionamos o de ULUSOY (1985) como aquele mais próximo da realidade dos problemas gerais em redes capacitadas, e o de maior acordo com os estudos de circuitos eulerianos realizados preliminarmente no capítulo II. Propusemos uma modificação no algoritmo de Ulusoy, utilizando o algoritmo proposto por BODIN & KURSH (1979), que minimiza o esforço computacional do algoritmo consideravelmente, haja vista a redução de esforços desnecessários na elaboração da grande rota e criação de um grafo modificado, G^* . O resultado com esta modificação, foi o melhor encontrado por Ulusoy para o seu problema exemplo, também aqui discutido.

Apesar de apenas termos mostrado com exemplos, como se processa os algoritmos estudados, a sua implementação computacional pode ser de extrema utilidade para a sociedade, haja vista a grande quantidade de benefícios que o conhecimento de roteiros mais favoráveis podem causar, por exemplo: nos serviços da distribuição de gás butano, na definição de roteiros e número de leituristas de energia, na pulverização de ruas para erradicar mosquitos transmissores de doenças,

dentre outros.

No capítulo III iniciamos com uma pequena demonstração das atividades de operação de uma frota de coleta de lixo, e as necessidades de se fazer uma rotatividade entre os veículos que realizam o serviço para garantir uma redução na depreciação destes. Propusemos, então, um modelo para o Problema de Planejamento da Frota e mostramos a resolução deste aproveitando as suas características, usando o algoritmo enumerativo de BALAS, MACULAN (1979), o qual provou ser bastante eficiente. Mostramos também a necessidade de se resolver um problema de alocação de veículos aos circuitos de trabalho, de modo que possa ser cumprida a tarefa do gerente de operações das atividades de coleta..

Em seguida, discutimos o Problema do Percorso do Caminhão Coletor de Lixo Urbano, de modo específico o problema da coleta do lixo na cidade de Fortaleza, e propusemos um algoritmo heurístico original, denominado de Algoritmo MARCOLIX, de alto desempenho, o qual consegue determinar rotas viáveis com boa qualidade. Neste algoritmo, é possível fazer melhorias para aumentar o seu desempenho, como por exemplo, reduzindo o número de vértices na quebra dos elos de mão-dupla, além de um pós-processamento após ser encontrada a rota do Caminhão Coletor, de modo a verificar a possibilidade da redução da utilização repetida e indesejável de arcos - Ver Apêndice 4. Uma outra abordagem, poderia ser a adaptação do algoritmo MARCOLIX para a solução de problemas mais gerais, como por exemplo, o problema de definição da melhor frota com ou sem restrição do número de veículos em uma dada área de serviço e problemas com mais de uma garagem (multi-depósito) e/ou mais de um despejo (multi-despejo).

Apesar de sabermos que os métodos atualmente empregados para execução dos serviços de coleta domiciliar, não obedecem a um critério científico, na elaboração de roteiros e planejamento da frota dos veículos, acreditamos que todo o esforço que se faz no sentido de dotar os usuários da coleta, de mais informação e de melhores soluções para o desempenho das tarefas, é de extrema relevância, pois não somente reduz os custos de operação da atividade, como também humaniza o serviço a ser prestado. Acreditamos, portanto, que este trabalho poderá contribuir, de forma relevante, na solução de um dos problemas da coleta do lixo, e assim reduzir o alto desgaste daqueles que realizam diariamente este serviço tão necessário e vital à sociedade atual.

Referências Bibliográficas

- ASSAD, A., PEARN, W. & GOLDEN, B. (1987) - "The Capacited Chinese Postman Problem: Lower Bounds and Solvable Cases", *Am. J. Mathematics and Management Science*, Vol. 7 pgs. 63-88
- AUMEF (1984) - "Plano Metropolitano de Limpeza Pública - Relatório No. 2: Concepção e Análise de Alternativas" - *Governo do Estado do Ceará - Secretaria de Planejamento e Coordenação - SEPLAN/CE - Fortaleza*
- BAZARAA, M. S. & JARVIS, J. J. (1977) - *Linear Programming and Network Flow* - New York, John Wiley & Sons.
- BEASLEY, J. (1983) - "Route first-cluster second methods for vehicle routing" - *Omega*, Vol. 11(4) pgs. 403-408
- BEASLEY, J. E. (1984) - "Fixed routes" - *Computers & Op. Res.*, Vol. 35(1) pgs. 49-55
- BOAVENTURA, P., MASSOTE, A., COSTA, L., MATTOS, N. & COSTA, R. (1985) - "Um sistema de roteamento de percursos pré-eulerianos" - *Anais do XIX SBPO/IV CLAIO - Rio de Janeiro*
- BOAVENTURA, P. (1985) - "Análise da melhoria máxima admissível em -um percurso pré-euleriano sub-ótimo" - *Anais do XVIII SBPO, São José dos Campos*
- BODIN, L. & BELTRAMI, E.J. (1974)- "Networks and vehicle routing for municipal waste collection" - *Networks*, Vol. 4 pgs. 65-94
- BODIN, L. & KURSH, S. (1979) - "A detailed description of a computer system for the routing and scheduling of street sweepers" - *Computers & Op. Res.*, Vol. 6 pgs. 181-198
- BODIN, L. & GOLDEN, B. (1981) - "Classification in vehicle routing and scheduling" - *Networks*, Vol. 11 pgs. 97-108
- BODIN, L., FAGIN, G., WELEBNEY, R. & GREENBERG, J. (1989) - "The design of a computerized sanitation vehicle routing and scheduling system for the town of Oyster Bay, New York", *Computers & Op. Res.*, Vol. 16(1) pgs. 45-54
- BUENO FILHO, R. (1989) - "Metodos de resolucao para o problema de percursos de veiculos" - *COPPE/UFRJ - Tese de Mestrado, Eng. Sistemas*
- BURKARD, R. & DERIGS, U. (1980) - *Assignment and Matching Problems: Solution Methods with FORTRAN programs* - Lecture Notes in Economics and Mathematical Systems, No. 184, Springer, Berlin.
- CHAPLEAU, L., FERLAND, J., LAPALME, G. & ROUSSEAU (1984) - "A parallel insert method for the capacited arc routing Problem" - *Operations Reserch Letters*, Vol. 3 pgs. 95-99

- CHAPLEAU, L., FERLAND, J. & ROUSSEAU, J. M. (1985) - "Clustering for routing in densely populated areas" - *European Journal of Op. Res.*, Vol. 20 pgs. 48-57
- CHRISTOFIDES, N. & EILON, S. (1969) - "An algorithm for the vehicle dispatch problem" - *Operations Res. Quarterly*, Vol. 20 pgs. 309- 318
- CHRISTOFIDES, N. (1973) - "The optimal traversal of a graph" - *Omega*, Vol. 1 pgs. 719-732
- CHRISTOFIDES, N. (1975) - *Graph Theory - An Algorithm Approach* - New York, Academic Press.
- CHRISTOFIDES, N. (1976) - "The vehicle routing problem" - *RAIRO Recherche Operationnelle*, Vol. 10 pgs. 55-70
- CHRISTOFIDES, N., MINGOZZI, A. & TOTH, P. (1981) - "Exact algorithm for the vehicle routing problem" - *Mathematical Programming*, Vol. 20 pgs. 255-282
- CHRISTOFIDES, N., CAMPOS, V. CORBERÁN, A. & MOTA, E. (1986) - "An algorithm for the rural postman problem on a directed graph." - *Mathematical Programming Study*, Vol. 26 pgs. 155-166.
- CLARK, R. M. & GILLEAN, J. I. (1975) - "Analyses of solid waste management operations in Cleveland, Ohio: A case study", *Interfaces*, Vol. 8(1) pgs. 82-42
- CLARK, R. M. & LEE JR., C. H. (1976) - "Systems planning for solid waste collection", *Computers & Op. Res.*, Vol. 3 pgs. 157-173
- CLARKE, G. & WRITE, J. (1964) - "Scheduling of vehicles from a central depot to a number of delivery points" - *Operation Research*, Vol.12 pgs. 568-581
- DIJKSTRA, E. (1959) - "A note on two problems in connection with graphs" - *Numer. Math.*, Vol. 1 pgs. 269-271
- DANTZIG, G. & RAMSER, J. (1959) - "The truck dispatching problem" - *Management Science*, Vol. 6 pgs. 80-91
- EDMONDS, J. & JOHNSON, E. (1973) - "Matching, Euler tours and the Chinese Postman Problem" - *Mathematical Programming*, Vol. 5 pgs. 88-124
- FISHER, M. & JAIKUMAR, R. (1981) - "A generalized assignment heuristic for vehicle routing" - *Networks*, Vol. 11 pgs. 109-124
- FONSECA NETO, R. (1989) - "Relacionamento entre inteligência artificial e programação matemática: Aplicação a um problema de "Scheduling" - Parte I" - *Resumos XXII SBPO* - pgs. 511 a 517.
- FREDERICKSON, G. (1979) - "Approximation algorithms for some postman problems" - *Journal of the ACM* , Vol. 26(3) pgs. 538-554

- FREDERICKSON, G., HECHT, M. & KIM, C. (1978) - "Approximation algorithms for some routing problems" - *SIAM Journal of Computers*, Vol. 7(2) pgs. 178-193
- GASKELL, T. (1967) - "Bases for vehicle fleet scheduling" - *Op. Res. Quart.*, Vol. 18 pgs. 281-295
- GEOFFRION, A. (1968) - "Surrogate constraints" - *Operations Research*, Vol. 16(4)
- GEOFFRION, A. (1969) - "An improved enumeration approach for integer programming" - *Operations Research*, Vol. 16(4)
- GILLET, B. & MILLER, L. (1974) - "A heuristic algorithm for the vehicle dispatch problem" - *Operation Research*, Vol. 22 pgs. 340- 349
- GLOVER, F., KLINGMAN, D. & PHILLIPS, N. (1985a) - "A new polynomially bounded shortest path algorithm" - *Operations Res.*, Vol. 33(1) pgs. 65-73
- GLOVER, F., KLINGMAN, D. & PHILLIPS, N. (1985b) - "New polynomially shortest path algorithm and their computational attributes" - *Management Science*, Vol. 31 pgs. 1106-1128
- GLOVER, F. & KLINGMAN, D. (1989) - "New sharpness properties, algorithms and complexity bounds for partitioning shortest path procedures" - *Operations Res.*, Vol. 37 pgs. 542-546
- GOLDEN, B., MAGNANTI, T. & NGUYEN, H. (1977) - "Implementing vehicle routing algorithms" - *Networks*, Vol. 7 pgs. 113-148
- GOLDEN, B. & WONG, R. (1981) - "Capacited arc routing problems" - *Networks*, Vol. 11 pgs. 305-315
- GOLDEN, B., DEARMON, J., BAKER, E. (1983a) - "Computational experiments with algorithms for a class of routing problems" - *Computers & Op. Res.*, Vol. 10 pgs. 47-59
- GOLDEN, B., BODIN, L., ASSAD, A. & BALL, M. (1983b) - "Routing and Scheduling of vehicle and crews - The state of art" - *Computers & Op. Res.*, Vol. 10(2) pgs. 63-211
- GOLDEN, B., ASSAD, A., LEVY, L., GHEYSENS, F. (1984) - "The fleet size and mix vehicle routing problem", *Computers & Op. Res.*, Vol. 11(1) pgs. 49-66
- GOLDEN, B. & BODIN, L. (1986) - "Microcomputer based vehicle routing and scheduling software", *Computers & Op. Res.*, Vol. 13(2/3) pgs. 277-285
- GOLDEN, B. & ASSAD, A. (1988) - *Vehicle Routing - Methods and Studies* - North Holland, Ed. Vol. 16 - Studies in Management Science and Systems
- GONDRAM, M. & MINOUX, M. (1979) - *Graphs and Algorithms* - John Willey & Sons, Ed. Willey - Interscience Series in Discrete Mathematics

- GUAN, M. (1984) - "On the windy postman problem" - *Discrete Applied Mathematics*, Vol. 9 pgs. 41-46
- HARTINGAN, J. A. (1974) - *Clustering Algorithms* - New York, John Wiley and Sons
- LAWLER, E. (1976) - *Combinatorial Optimization - Networks and Matroids* - Holt, Rinehart and Winston, NY
- LENSTRA, J. & RINNOOY KAN, A. (1976) - "On general routing problems" - *Networks*, Vol. 6 pgs. 273-280
- LENSTRA, J. & RINNOOY KAN, A. (1981) - "Complexity of vehicle routing and scheduling problems" - *Networks*, Vol. 11 pgs. 221-227
- LIN, S. & KERNINGHAM, B. (1973) - "An effective heuristic algorithm for the traveling salesman problem" - *Operations Research*, Vol. 21 pgs. 498-516
- KWAN, MEI-KO (1962) - "Graphic programming using odd or even points" - *Chinese Mathematics*, Vol. 1 pgs. 273-277
- MACULAN, N. (1978) - *Programação Linear Inteira* - Publicações COPPE/UFRJ.
- MAGNANTI, T. (1981) - "Combinatorial optimization and vehicle fleet planning perspectives and prospects." - *Networks*, Vol. 11 pgs. 179-213
- MATION, H. (1982) - "Caminhos Eulerianos de mínimo custo em grafos Eulerianos" - *Anais do XV Simposio Brasileiro de Pesquisa Operacional / I Congresso Latino-Americano de Pesquisa Operacional e Eng. de Sistemas*, pgs. 483-492
- MATION, H. (1983) - "Otimização de sistemas de coleta de lixo - Notas preliminares" - *COPPE/UFRJ - Relatório Técnico, Eng. de Produção*
- MATION, H. (1984) - "Um modelo de distribuição aplicado ao problema do transporte de resíduos gerados em grandes cidades" - *COPPE/UFRJ, Tese Mestrado, Eng. Produção*
- MCBRIDGE, R. (1982) - "Controlling left and U-turns in the routing of refuse collection vehicles", *Computers & Op. Res.*, Vol. 9(2) pgs. 145-152
- MINIEKA, E. (1978) - *Optimization Algorithms for Networks and Graphs* - Industrial Engineering/1, Marcel Dekker, Inc.
- MINIEKA, E. (1979) - "The chinese postman problem for mixed networks" - *Management Science*, Vol. 23(7) pgs. 643-658
- MOLE, R. & JAMESON, S. (1976) - "A sequential routing-building algorithm employing a generalised savings criterion" - *Op. Res. Quater.*, Vol. 27 pgs. 503-511
- MOLE, R., JOHNSON, J. & WELLS, K. (1983) - "Combinatorial analysis for route-first cluster second vehicle routing" - *Omega*, Vol. 11(5) pgs. 507-512

- NEGREIROS, M. G., RIOS, V., FONTELLAS, A. C. , CRAVEIRO, R., FELIX, E. , ALENCAR , J., BRUNO, P. (1990) - "Roteamento de Veículos e Planejamento de Frotas para Recolhimento do Lixo Urbano Domiciliar na Região Metropolitana de Fortaleza (Projeto Piloto)" - *Relatório final à Construtora Marquise LTDA.*
- NELSON, M. D., NYGARD, K. E., GRIFFIN, J. H. & SHREVE, W. E. (1985) - "Implementation techniques for the vehicle routing problem" - *Computers & Op. Res.*, Vol. 12(3) pgs. 273-283
- ORLOFF, C. (1974a) - "A fundamental problem on vehicle routing" - *Networks*, Vol. 4 pgs. 34-64
- ORLOFF, C. (1974b) - "Routing a fleet of m vehicles to/from a central facility" - *Networks*, Vol. 4(B) pgs. 147-162
- PAPADIMITRIOU, C. (1976) - "On the complexity of edge traversing" - *Journal of the ACM*, Vol. 23(3) pgs. 544-554
- PEARN, W. ,ASSAD, A. & GOLDEN, B. (1987) - "Transforming arc routing into node routing problems", *Computers & Op. Res.*, Vol. 14(4) pgs. 285-288
- PEARN, W. L. (1989) - "Approximate solutions for the capacited arc routing problem", *Computers & Op. Res.*, Vol. 16(6) pgs. 589-600
- PUETZ, W. (1979) - "The Chinese Postman Problem" - Numerische Untersuchungen Chinesischen, Postboten-Problem, Diploma thesis, *Mathematics Institut der Universität zu Köln*
- ROY, S. & ROUSSEAU, J. M. (1989) - "The capacited canadian postman problem" - *IFORS*, Vol. 27(1)
- STERN, H. & DROR, M. (1979) - "Routing electric meter readers", *Computers & Op. Res.*, Vol. 6 pgs. 209-223
- SYSLO, M. ,DEO, N. & KOWALIK, J. (1983) - *Discrete Optimization Algorithms with Pascal Programs* - Prentice Hall, Ed.
- TARJAN, R. E. (1972) - "Depth-First Search and Linear Graph Algorithms" - *SIAM J. of Computing*, Vol. 1(2), 146-160
- THANH, V. ,TURNER, W. & ESVALANTE, O. (1976) - "Systems design for rural refuse collection" - *AIIE Transactions*, Vol. 8(1) pgs. 84- 95
- TURNER, W. & HOUGLAND, E. (1975) - "The optimal routing of solid waste collection" - *AIIE Transactions*, Vol. 7 pgs. 427-431
- ULUSOY, G. (1985) - "The fleet size and mix problem for capacited arc routing" - *European Journal of Op. Res.*, Vol 22 pgs 329-337
- WYSKIDA, R. & GUPTA, J. (1973) - "IE's improve city's solid waste collection" - *Industrial Engineering*, Vol. 4(6) pgs. 12-15

- YELLOW, P. (1970) - "A computational modification to the savings method of vehicle scheduling" - *Op. Res. Quart.*, Vol. 21 pgs. 281-283**
- YAXIONG LIN & YONGCHANG, Z. (1988) - "A new algorithm for the direct chinese postman problem" - *Comp. & Oper. Res.*, Vol. 15 pgs. 577-584 (1988)**
- ZAW WIN (1989) - "On the windy postman problem on eulerian graphs" - *Mathematical Programming*, Vol. 44 pgs. 97-112**

Apêndice 1

Complexidade Computacional dos Algoritmos de Roteamento

Durante muito tempo a literatura se ocupou em formular problemas de roteamento, outros os agruparam em classes bastante definidas – GOLDEN ET AL (1981), mas ficou em aberto um estudo mais evidente da solução de problemas para cada subdivisão conhecida. LENSTRA & RINNOOY KAN (1981) publicaram o único artigo completo que procura determinar a complexidade dos problemas, ou simplesmente classificá-los em sua ordem. Neste estudo determina-se a complexidade das heurísticas mais populares conhecidas. Mostraremos, a seguir, as tabelas de classificação e complexidade apresentadas naquele artigo, e o fazemos por acharmos de grande valor o seu conhecimento e no auxílio da leitura deste texto. Consideramos, então:

V – Conjunto de vértices de um grafo fortemente conexo;

$\mathcal{V} \subseteq V$ – Conjunto de vértices para serviço;

E – Conjunto de elos de um arco fortemente conexo;

$\mathcal{E} \subseteq E$ – Conjunto de elos para serviço;

A – Conjunto de arcos de um grafo fortemente conexo;

$\mathcal{A} \subseteq A$ – Conjunto de arcos para serviço;

$[t_i, u_i]$ – Intervalo de tempo de serviço de um determinado elemento de demanda;

m -PPV – Problema de Percurso (Roteamento) de Veículos com mais de um veículo;

ℓ -PPV – Problema de Percurso (Roteamento) de Veículos com mais de um depósito central;

n – Número de vértices de um grafo fortemente conexo;

a – Número de arcos de um grafo fortemente conexo;

Definição: Um problema é dito \mathcal{NP} - Hard quando não pode ser resolvido em tempo polinomial, a menos que $\mathcal{P} = \mathcal{NP}$.

Nome	Código	E	A	V	\mathcal{E}	\mathcal{A}
Problema do Caixeiro Viajante	PCV	- ^a	\emptyset	V	\emptyset	\emptyset
Problema do Caixeiro Viajante Direcionado	PCVd	\emptyset	-	V	\emptyset	\emptyset
Problema do Carteiro Chinês	PCC	-	\emptyset	\emptyset	E	\emptyset
Problema do Carteiro Chinês Direcionado	PCCd	-	-	\emptyset	\emptyset	A
Problema do Carteiro Chinês Misto	PCCm	-	-	\emptyset	E	A
Problema do Carteiro Rural	PCR	-	\emptyset	\emptyset	-	\emptyset
Problema do Carteiro Rural Direcionado	PCRd	\emptyset	-	\emptyset	\emptyset	-
Problema do Geral de Roteamento	PGR	-	\emptyset	-	-	\emptyset

-^a: Arbitrário

- : Conjunto a servir

\emptyset : Vazio

Tabela V.1: Problemas de Roteamento Simples ou Básicos.

Problema de Roteamento	Complexidade
PPV	\mathcal{NP} -Hard
PCV	\mathcal{NP} -Hard
PCVd	\mathcal{NP} -Hard
PCC	$O(n^3)$
m-PCC	\mathcal{NP} -Hard
PCCd	\mathcal{NP} -Hard
m-PCCd	\mathcal{NP} -Hard
PCCm	\mathcal{NP} -Hard
PCR	\mathcal{NP} -Hard
PCRd	\mathcal{NP} -Hard
PGR	\mathcal{NP} -Hard
PPV ($\forall t_i = u_i$)	$O(n^3)$
PPV ($\forall t_i = 0, \forall u_i = u$)	\mathcal{NP} -Hard
ℓ - PPV ($\forall t_i = u_i$)	Aberto

Tabela V.2: Complexidade dos Problemas de Roteamento mais conhecidos.

Apêndice 2

O Algoritmo Primal-Dual OUT-OF-KILTER

Consideremos o problema de encontrar um fluxo de um dado valor ν de s a t em um grafo direcionado, tal que o custo do fluxo é minimizado. Neste problema, cada arco (i,j) tem dois números associados a ele, a capacidade q_{ij} e o custo por unidade de fluxo ao longo de cada arco C_{ij} .

Obviamente se ν possui um valor maior que o fluxo máximo de s a t , nenhuma solução existiria. Entretanto, se ν é um valor menor ou igual ao fluxo máximo, um número de diferentes padrões de fluxo irá, em geral, ser possível, e portanto aquele fluxo de menor custo. O procedimento mais adequado para o fluxo de custo mínimo é chamado algoritmo OUT-OF-KILTER de FORD & FULKERSON (1960).

Este algoritmo é idêntico ao Primal-Dual Simplex, o qual inicia com uma viabilidade dual, mas não necessariamente primal viável, e interage entre os problemas primal e dual até que sua otimalidade seja encontrada. Entretanto, difere do Primal-Dual Simplex pois o OUT-OF-KILTER, nem sempre mantém folgas complementares. Assim, ele pode ser visto como uma generalização de algoritmo Primal-Dual para Problemas de Fluxo em Redes.

A forma em modelo de programação matemática para o Problema de Fluxo em Redes de Mínimo custo, é:

$$(PFRM) \quad \text{Min} \sum_{i=1}^m \sum_{j=1}^m C_{ij} X_{ij} \quad (V.1)$$

sujeito a:

$$\sum_{j=1}^m X_{ij} - \sum_{k=1}^m X_{ki} = 0 \quad (i = 1, \dots, m) \quad (V.2)$$

$$l_{ij} \leq X_{ij} \leq u_{ij} \quad (i, j = 1, \dots, m) \quad (V.3)$$

onde entende-se que as somas e desigualdades são admitidas apenas sobre arcos existentes. Chamamos de *Conservação de Fluxo*, todo fluxo (escolha dos X_{ij}) satisfazendo a restrição 2. Uma conservação de fluxo que satisfaz as restrições reminiscentes $l_{ij} \leq X_{ij} \leq u_{ij}$ é um *fluxo viável*. Admitiremos que C_{ij} , l_{ij} e u_{ij} são inteiros e que $0 \leq l_{ij} \leq u_{ij}$.

Descrição do Algoritmo

Passo Inicial

Selecione uma conservação de fluxo, por exemplo, cada X_{ij} , e coloque as variáveis duais, tal que cada $w_i = 0$.

Passo Principal

1. Se todos os arcos estão IN-KILTER, pare \Rightarrow A solução é ótima. Senão, selecione (ou continue com o previamente selecionado) um arco OUT-OF-KILTER, chame-o (p, q) . Apague todos os rótulos existentes.
Se (p, q) está em um dos estados onde o fluxo aumenta, Δ_{pq} é necessário. Faça $s \leftarrow q$, $t \leftarrow p$, e $L(s) \leftarrow (+t, \Delta_{pq})$.
2. Se o vértice i tem um rótulo, o vértice j não tem rótulo, o fluxo deve ser aumentado de uma quantidade Δ_{ij} ao longo do arco (i, j) . Então, atribua ao vértice j o rótulo $L(j) \leftarrow (+i, \Delta_j)$ onde $\Delta_j \leftarrow \min\{\Delta_j, \Delta_{ij}\}$. Se o vértice i tem um rótulo, então o vértice j não tem rótulo, e o fluxo deve diminuir de Δ_{ji} ao longo do arco (j, i) . Então, dado o vértice j temos o rótulo $L(j) \leftarrow (-i, \Delta_j)$ onde $\Delta_j \leftarrow \min\{\Delta_j, \Delta_{ji}\}$. Repita o passo 2 até que tanto o vértice t seja rotulado, ou até que não haja vértices que possam ser rotulados. Se o vértice t está rotulado, vá para o passo 3 (um *Breakthrough* foi encontrado), senão vá para o passo 4 (*NonBreakthrough* ocorreu).
3. Faça $\Delta \leftarrow \Delta_j$. Mude o fluxo ao longo dos ciclos identificados como segue. Inicie no vértice t , se o primeiro elemento em $L(t)$ está $+k$, então adicione Δ a X_{kt} . Senão, se o primeiro elemento em $L(t)$ é $-k$, então subtraia Δ de X_{tk} . Volte ao vértice k e repita o processo até que o vértice t seja atingido novamente no processo de retorno (Backtrack). Vá para o passo 1.
4. Faça X ser o conjunto de vértices rotulados e faça F ser o conjunto de vértices rotulados e faça $\bar{F} \leftarrow N - F$. Defina, $S_1 \leftarrow \{(i, j) \mid i \in F, j \in \bar{F}, Z_{ij} - C_{ij} < 0, X_{ij} \leq u_{ij}\}$ e $S_2 \leftarrow \{(i, j) \mid i \in \bar{F}, j \in F, Z_{ij} - C_{ij} > 0, X_{ij} \geq l_{ij}\}$.

Faça $\theta \leftarrow \{\min\{|Z_{ij} - C_{ij}|, \infty\} \mid (i, j) \in S_1 \cup S_2\}$

Se $\theta = \infty$, pare \Rightarrow Não há solução viável, senão

Faça: $w_i' \leftarrow \begin{cases} w_i + \theta, & \text{se } i \in F \\ w_i, & \text{se } i \in \bar{F} \end{cases}$

e

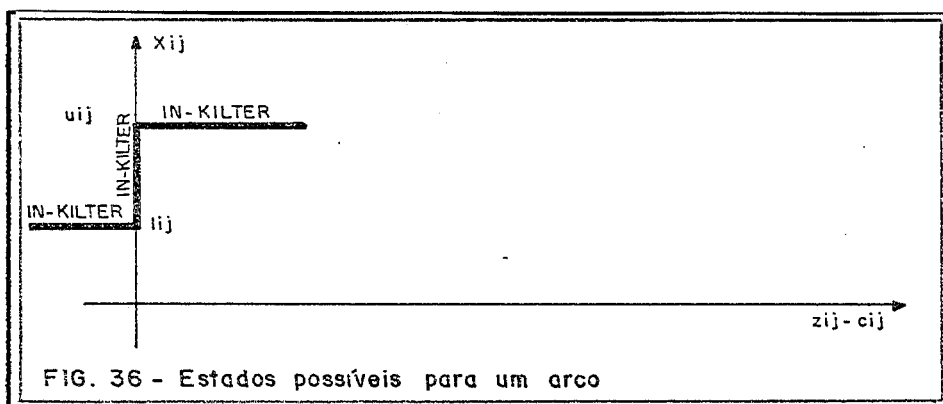
$$(Z_{ij} - C_{ij})' \leftarrow \begin{cases} (Z_{ij} - C_{ij}), & \text{se } (i, j) \in (F, F) \cup (\bar{F}, \bar{F}) \\ (Z_{ij} - C_{ij}) + \theta, & \text{se } (i, j) \in (F, \bar{F}) \\ (Z_{ij} - C_{ij}) - \theta, & \text{se } (i, j) \in (\bar{F}, F) \end{cases}$$

Vá para o passo 1.

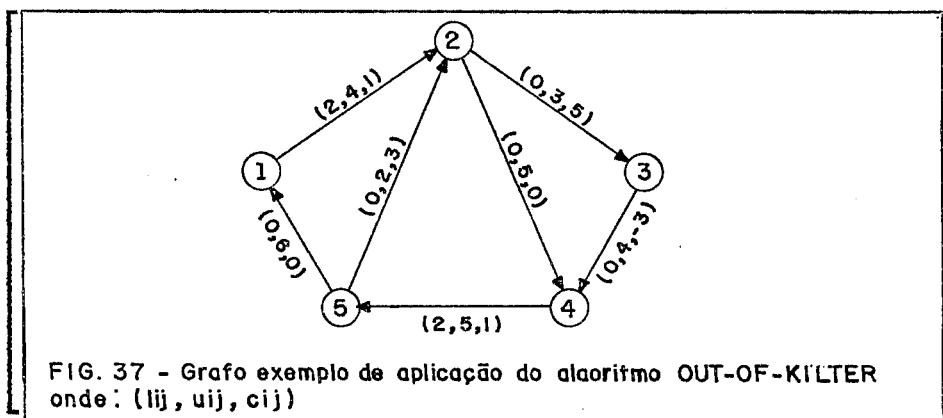
Exemplo

	$(Z_{ij} - C_{ij}) < 0$	$(Z_{ij} - C_{ij}) = 0$	$(Z_{ij} - C_{ij}) > 0$
$X_{ij} > u_{ij}$	OUT	OUT	OUT
$X_{ij} = u_{ij}$	OUT	IN	IN
$l_{ij} < X_{ij} < u_{ij}$	OUT	IN	OUT
$X_{ij} = l_{ij}$	IN	IN	OUT
$X_{ij} < l_{ij}$	OUT	OUT	OUT

Tabela V.3: Estados KILTER do Problema de Fluxo em Redes de Custo Mínimo



Considere a rede da Fig. 37. Inicializando o algoritmo OUT-OF-KILTER com o qual $X_{ij} = 0$ e cada $w_i = 0$, temos a sequência primal e dual com o procedimento de rotulação:



Passo de Inicialização

Arcos OUT-OF-KILTER: $\{ (1,2), (4,5) \}$
 $(p, q) = (1, 2), s = 2, t = 1, L(2) = (1, 2)$
 $L(4) = (+2, 2)$
 $L(5) = (+4, 2)$
 $L(1) = (+5, 2)$

Breakthrough: $\Delta = 2 \quad L_1 = +5 \Rightarrow X'_{5,1} = X_{5,1} + \Delta = 2$
 $L_5 = +4 \Rightarrow X'_{4,5} = X_{4,5} + \Delta = 2$

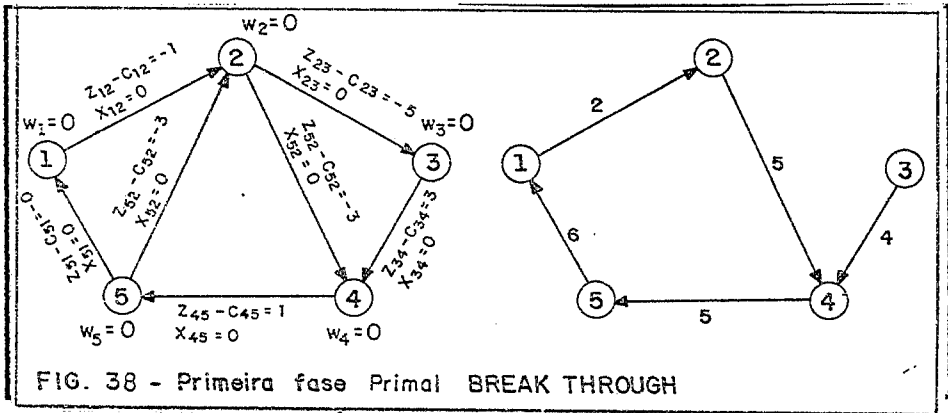


FIG. 38 - Primeira fase Primal BREAK THROUGH

$$L_4 = +2 \Rightarrow X'_{2,4} = X_{2,4} + \Delta = 2$$

$$L_2 = +1 \Rightarrow X'_{1,2} = X_{1,2} + \Delta = 2$$

... evoluindo o algoritmo, temos:

$$X^*_{1,2} = X^*_{2,4} = X^*_{4,5} = 2$$

$$X^*_{2,3} = X^*_{3,4} = X^*_{5,1} = X^*_{5,2} = 0$$

$$Z^* = -2$$

Apêndice 3

Telas do S i s R o t

A seguir, mostramos as telas principais do sistema S i s R o t. Nelas, podemos observar a parte de arquivos (Banco de Dados Geográfico) e informações referentes à frota, como também a execução do Roteamento na área de estudo do exemplo da Fig. 34. Nossa preocupação é de mostrar um pouco do nosso trabalho como um todo, e o que tivemos que fazer para desenvolver um estudo apropriado e já aplicar à coleta do lixo. Infelizmente até aqui, o S i s R o t não está todo completo, ainda nos falta desenvolver uma parte gráfica, acoplar o problema da coleta especial (Clientes - Roteamento em Nós), e concluir o acoplamento da parte de Planejamento da Frota, usando melhorias consideráveis no algoritmo de BALAS.

Dentre as facilidades do S i s R o t, como pode ser visto nas telas a seguir, temos um editor de texto para as conclusões do Roteamento e Planejamento da Frota de veículos. Em todos, é possível colocar mensagens ou alterar os resultados, conforme as necessidades do operador ou gerente.

S i s R o t - Versão 1.00
(C) Marcos José Negreiros Gomes. 1989
(C) Miguel Arcanjo Negreiros Gomes. 1989
Pressione Qualquer <Tecla>

VERSAD 1.00

COPPE/UFRJ - OTIMIZACAO
S i s t e m a

SRV
Zona Trab. : 007

Arquivos	Planos & Rotas	Informes	Rotas	Zona	Fim
----------	----------------	----------	-------	------	-----

Rede Viária
Clientes
Frota

M & M Negreiros 17:08:04 Sexta 05/10/90

* - Cadastra toda Rede Viária, como Ruas, Bairros, etc. - *

VERSAD 1.00

COPPE/UFRJ - OTIMIZACAO
 S i s t e m a

SRVAD113
 Zona Trab. : 007

Arquivos	Planos & Rotas	Informes	Rotas	Zona	Fim
----------	----------------	----------	-------	------	-----

- Rede Viária
- Bairros
 - V
 - R Inclusão
 - Z Alteração
 - S **Consulta**
 - Exclusão
 - Relatorio

CODIGO	NOME DO BAIRRO
001	BARRA DO CEARA
002	CJ. N. S. ASSUNCAO
003	JARDIM IRACEMA
004	ALVARO WEYNE
005	CARLITO PAMPLONA
006	VILA ELLERY
007	CRISTO REDENTOR
008	N. S. DAS GRACAS
009	PIRAMBU

< PgUp > Tela Anterior < PgDn > Próxima Tela.

M & M Negreiros

17:33:42 Sexta 05/10/90

* - Consulta os Bairros da Cidade. - *

VERSAD 1.00

COPPE/UFRJ -- OTIMIZACAO

S i s t e m a

SRVAG123

Zona Trab. : 007

Arquivos

Planos & Rotas Informes Rotas Zona Fim

CODIGO	NOME DA RUA	SEN	C.F	B-1	B-2	B-3	B-4	B-5	B-6
07043	R. JOSE NAPOLEAO	U	0	013					
07044	R. PROF. DIAS DA ROCHA	D	0	013	015	023			
07045	R. CEL. JUDA	D	0	013	015	023			
07045	R. CEL. JUDA	U	0	024					
07046	R. MONSENHOR CATAD	D	0	013	015	023			
07047	R. FONSECA LOBO	D	0	015	023				
07048	R. MARECHAL RONDON	D	0	023					
07049	R. SEN. TOMAZ RODRIGUES	D	0	023					
07050	R. FREDERICO BORGES	D	0	015					

< PgUp > Tela Anterior < PgDn > Próxima Tela.

M & M Negreiros

17:36:10

Sexta

05/10/90

VERSAD 1.00

COPPE/UFRJ - OTIMIZACAO

S i s t e m a

SRVAD13R

Zona Trab. : 007

Arquivos

Planos & Rotas

Informes

Rotas

Zona

Fim

Rede Viária

Bairros
Vias

Rede

Inserção
Alteração
Consulta
Exclusão
Relatório

Gerar
RUA
NO

Código da Rua : 07009 Sentido : U

Nome : R. BARAD DE ARACATI

Cód. Bairro : 024 Zona : 007

Sub-Z: 002 Nº do N6 : 0093 Contra Fluxo: 0

Tipo de Via. (S/F). : S Cód. Mapa. : 005

Cód. Cruz. : 07100 R. DEP. MOREIRA DA ROCHA

Distância (mts)..... : 120.00

Nº de Casas..... : 10

Nº de Apartamentos... : 0

Nº de Comércio..... : 20

Nº de Clínicas..... : 0

Nº Esc., Clubes, Hotéis : 0

Carga (Kg)..... : 532.50 REC =70

M & M Negreiros

17:11:07

Sexta

05/10/90

< ↑ > Reg. Anterior < ↑ > Próximo Reg.

VERSÃO 1.00

COPPE/UFRJ - OTIMIZACAO
 S i s t e m a

SRVAD143
 Zona Trab. : 001

Arquivos	Planos & Rotas	Informes	Rotas	Zona	Fim
----------	----------------	----------	-------	------	-----

- Rede Viária
 - Bairros
 - Vias
 - Rede
 - Zonas
- Inclusão
- Alteração
- Consulta
- Exclusão
- Relatório

Código da Zona..... : 001
 ** Produção (Kg) por ou Ne de **
 Pessoa... : 0.90 Ne Ps/Res: 5.00
 Ne Ps/Apt: 4.50 Comércios: 20.00
 Clínicas.: 0.00 Extras... : 20.00
 Estimada.: 3.99 Ton Turno.....: D
 ** Velocidades de Coleta (em Km/Hr) **
 Vel. Vazio (P)..... : 30.00
 Vel. Vazio (S)..... : 20.00
 Vel. Carregado (P). : 22.00
 Vel. Carregado (S). : 17.00
 Vel. de Coleta..... : 2.90

REG = 1

M & M Negreiros 17:42:30 Sexta 05/10/90

< ↑ > Reg. Anterior < ↓ > Próximo Reg.

VERSÃO 1.00

COPPE/UFRRJ - OTIMIZACAO
 S i s t e m a

SRVAG0143
 Zona Trab. : 007

Arquivos	Pianos & Rotas	Informes	Rotas	Zona	Fim
----------	----------------	----------	-------	------	-----

- Rede Viária
 - Bairros
 - Vias
 - Rede
 - Zonas
- Inclusão
- Alteração
- Consulta
- Exclusão
- Relatório

Código da Zona..... : 007

** Produção (Kg) por ou No de **

Pessoa... : 0.65 No Ps/Res: 5.00

No Ps/Apt: 4.50 Comércios: 25.00

Clínicas.. : 15.00 Extras... : 20.00

Estimada.. : 133.82 Ton Turno.... : N

** Velocidades de Coleta (em Km/Hr) **

Vel. Vazio (P)..... : 27.00

Vel. Vazio (S)..... : 22.00

Vel. Carregado (P).. : 21.00

Vel. Carregado (S).. : 18.00

Vel. de Coleta..... : 3.00

REG = 10

M & M Negreiros	17:38:43	Sexta	05/10/90
-----------------	----------	-------	----------

< ↑ > Reg. Anterior < ↓ > Próximo Reg.

VERSAD 1.00

COPPE/UF RJ - OTIMIZACAO
 S i s t e m a

SRVIN F2
 Zona Trab. : 007

Arquivos Planos & Rotas **Informes** Rotas Zona Fim

Zona
Sub-Zona

```

**** ESTIMATIVA DE SUB-ZONA ****
Perimetro Total (mts) : 15,550.00
Total de Residências.. : 705
Total de Apartamentos. : 960
Total de Comércio.... : 395
Total de Clínicas..... : 0
Total de Extras..... : 40
Carga Estimada ... (kg) : 15,774.29
População Estimada.... : 7,845
  
```

Entre com a Sub-Zona : 002

M & M Negreiros 17:21:11 Sexta 05/10/90

* - Pressione uma tecla para encerrar. - *

VERSAD 1.00

COPPE/UFRJ - OTIMIZACAO
 S i s t e m a s

SRVAC223
 Zona Trab. : 001

Planos & Rotas	Informes	Rotas	Zona	Fim
----------------	----------	-------	------	-----

Cód Rua : 07250 VILA SAO PAULO
 Bairro..: 023
 Zona : 001 Sub-Zona : 000 NE do Nú : 0011
 Código do Mapa... : 000
 Cód. Cruzamento.. : 07007 R. IDELFONSO ALBANO
 Nº de Residências : 12
 Nº de Aptos..... : 0
 Nº de Comércio.. : 0
 Nº de Clínicas... : 0
 Nº Esc., Club, Hotel: 0
 Carga.....(Kg). : 85.00
 Tp. Serv. (min.). : 5.00

Rede Viária
 Planos & Rotas

 Especiais
 Planos & Rotas

 Inclusão
 Alteração
 Planos & Rotas
 Exclusão
 Relatorio

M & M Negreiros

18:41:30

Sexta

14/12/90

< ↑ > Reg. Anterior < ↓ > Próximo Reg.

VERSAD 1.00

COPPE/UFRRJ - OTIMIZACAO
S i s t e m a

SRVAD033
Zona Trab. : 001

Planos & Rotas

Informes

Rotas

Zona

Fim

Cód. Veículo : 002 Placa : A03434 Ano : 1989
Modelo : 1314 Carga (Ton) : 7.50 Cr#/km : 200.00

Ne Atualiz:	Horímetro	Km (gem)	Coleta	Abastec.
Atual.....	141.0	1,258.0	115.00	111
Data.....	12/12/90	12/12/90	12/12/90	12/12/90
Anterior.....	122.0	1,234.0	122.00	123
Data.....	17/11/89	17/11/89	17/11/89	17/11/89
Acumulado.....	19.0	24.0	237.00	234
Plano... (ROT)...	0.0	0.0	0.00	
Consumo Médio...	0.79 1/h	11.33 Km/l		

Tempo Serv. Corrido : 19.00 Turnos Corridos : 0

Cond. Veículo : N Data : Hora :
Acumulado..Hr : Acumulado..Hr :

M

Pressione [4] para continuar ou < F > para finalizar.

VERSAD 1.00

COPPE/UFERJ - OTIMIZACAO
S i s R o t

SRV
Zona Trab. : 001

Arquivos	Planos & Rotas	Informes	Rotas	Zona	Fim
----------	----------------	----------	-------	------	-----

Planejamento
Rotas

Demilitar

Prepara Arquivo

Para Rede Viaria
Situacao da Rede
Relat. Situacao da Rede

No. Arcos → Mão Unica: 5 Mão Dupla: 11 Reg.: 24

M & M Negreiros	18:46:31	Sexta	14/12/90
-----------------	----------	-------	----------

... Pressione uma Tecla.

VERSÃO 1.00

COPPE/UFRRJ - OTIMIZACAO

S i s t e m a

Rotas por Sub-Zona

Veiculo...: 00001 Ton.....: 3.00

Zona.....: 001 Sub-Zona...: 000

Garagem...: 0010000001 Despejo...: 00100000010

R\$0

US\$0

Tempo (Máx): 2:10 (hs)

Carga.....: 4.20 Ton Desvio (σ): 0.40 Ton

Viag. (Máx): 3 Ton/Viagem: 3.00

Resultados da Rota

Viagem	Tsa	% Coletado	Km
1.a	0:57	54.97	1.540
2.a	1:51	45.03	3.210
3.a	1:51	100.00	4.750

Deseja Gerar Rotas em Outra Zona? <S/N>

M

Sexta

14/12/90

COPPE/UF RJ - OTIMIZACAO S i s t e m a		VERSAO 1.00 SRVRLD1 Zona Trab. : 001	
Arquivos	Planos & Rotas	Informes	Zona Fim
COPPE/UF RJ - OTIMIZACAO DEPARTAMENTO DE LIMPEZA URBANA (DLU)			
---> Editor de Conclusao			
Veiculo: 00001	Carga Nominal (Ton): 3.00		
Inicio da Rota: DLU	Fim da Rota: Aterro Sanitario		
No. de Garis:	Tempo Maximo de Servico: 2hs 10min.		
ROTEIRO DE SERVICIO NO CIRCUITO 001-000/ 1a. Viagem			
(*) Colete na R. SILVA JATAI	ate AV. SANTOS DUMONT		
(*) Colete na R. SILVA JATAI	ate AV. ANTO. SALES		
***** ESTACIONE O VEICULO E COLETE NA AV. ANTO. SALES			
F3 + Imprime ^M + Grava			
Use: PgUp/PgDn (Paginação), Home/End (Inicio/Fim de linha) e Setas.			
M & M Negreiros	19:58:21	Sexta	14/12/90

Apêndice 4

Desempenho Computacional do Algoritmo MARCOLIX

Após termos desenvolvido o algoritmo MARCOLIX, realizamos vários testes práticos com a heurística. Tomamos como base, principalmente, o exemplo proposto neste texto, de modo que pudéssemos observar aspectos de operação das heurísticas construídas a partir do modelo original. Também nos preocupamos em desenvolver pesquisas em áreas maiores, grafos com mais de 5500 arcos e 2500 nós. Esta rede foi construída a partir de nosso trabalho em campo em Fortaleza, NEGREIROS ET AL (1990).

No final, mostramos o formato das rotas geradas com os melhores resultados obtidos para os problemas 1 e 2, e comparamos os melhores resultados obtidos na execução dos problemas 5 e 6 com o ótimo conhecido. Em tais problemas, observamos que o algoritmo MARCOLIX está sempre dentro do limite de 1,5 proposto por GOLDEN & WONG (1981), para o PCA.

Na verdade, é bom lembrar que nada se pode garantir sobre o limite superior do algoritmo, pois o MARCOLIX é um problema \mathcal{NP} -HARD, então uma grande variedade de contra exemplos podem surgir em virtude de um pior caso estudado.

BALANCEADO	VDD	VDP	VPP	VPD
Qv (Ton)	2.0	2.0	2.0	2.0
TSt (Hs)	2.0	2.0	2.0	2.0
NVg	2	2	2	2
Km	5.31	5.26	5.59	5.28
TSa	1h 51min	1h 50min	1h 45min	1h 50min
% Servido	100.00	100.00	100.00	100.00
Ton/Km-Hr (Min)	0.27	0.28	0.27	0.28
Ton/Km-Hr (Max)	0.36	0.36	0.35	0.36
TPr (Seg)	6.04	3.01	7.98	6.62

Tabela V.4: Problema 1, ($n = 32$, $m = 63$), Origem=Garagem, Destino=Aterro (Jangurussu), Fig. 34

BALANCEADO	VDD	VDP	VPP	VPD
Qv (Ton)	2.0	2.0	2.0	2.0
TSt (Hs)	2.0	2.0	2.0	2.0
NVg	2	2	2	2
Km	7.64	5.50	5.23	5.23
TSa	1h 59min	1h 43min	1h 38min	1h 38min
% Servido	100.00	93.79	92.25	92.25
Ton/Km-Hr (Min)	0.18	0.26	0.29	0.28
Ton/Km-Hr (Max)	0.23	0.35	0.38	0.36
TPr (Seg)	3.007	2.212	2.090	2.270

Tabela V.5: Problema 2, ($n = 32$, $m = 63$), Origem=Aterro (Jangurussu), Destino=Aterro (Jangurussu), Fig. 34

BALANCEADO	VDD	VDP	VPP	VPD
Qv (Ton)	1.5	1.5	1.5	1.5
TSt (Hs)	1.0	1.0	1.0	1.0
NVg	1	2	1	1
Km	1.46	3.60	1.41	2.35
TSa	0h 51min	1h 14min	0h 52min	1h 06min
% Servido	48.73	48.73	46.86	56.30
Ton/Km-Hr (Min)	0.83	0.25	0.85	0.52
Ton/Km-Hr (Max)	1.52	0.43	1.50	0.82
TPr (Seg)	6.084	7.144	7.680	7.01

Tabela V.6: Problema 3, ($n = 32$, $m = 63$), Origem=Garagem, Destino=Aterro (Jangurussu), Fig. 34

BALANCEADO	VDD	VDP	VPP	VPD
Qv (Ton)	1.5	1.5	1.5	1.5
TSt (Hs)	1.0	1.0	1.0	1.0
NVg	1	2	2	2
Km	3.71	3.65	3.74	3.74
TSa	1h 03min	1h 10min	1h 08min	1h 08min
% Servido	53.83	78.53	77.37	77.37
Ton/Km-Hr (Min)	0.32	0.48	0.47	0.47
Ton/Km-Hr (Max)	0.53	0.66	0.66	0.66
TPr (Seg)	7.056	6.194	4.590	5.14

Tabela V.7: Problema 4, ($n = 32$, $m = 63$), Origem=Aterro (Jangurussu) Destino=Aterro (Jangurussu), Fig. 34

BALANCEADO	VDD	VDP	VPP	VPD
Qv (Ton)	4.0	4.0	4.0	4.0
TSt (Hs)	2.0	2.0	2.0	2.0
NVg	1	1	1	1
Km	4.765	4.765	5.78	5.225
TSa	1h 30min	1h 30min	1h 34min	1h 32min
% Servido	100.00	100.00	100.00	100.00
Ton/Km-Hr (Min)	0.38	0.38	0.30	0.34
Ton/Km-Hr (Max)	0.49	0.49	0.39	0.43
TPr (Seg)	5.033	3.191	3.380	2.88

Tabela V.8: Problema 5, ($n = 32$, $m = 63$), Origem=Garagem, Destino=Aterro (Jangurussu), Fig. 34

BALANCEADO	VDD	VDP	VPP	VPD
Qv (Ton)	4.0	4.0	4.0	4.0
TSt (Hs)	2.0	2.0	2.0	2.0
NVg	1	1	1	1
Km	5.58	5.58	5.29	5.290
TSa	1h 33min	1h 32min	1h 30min	1h 30min
% Servido	100.00	100.00	100.00	100.00
Ton/Km-Hr (Min)	0.31	0.31	0.34	0.34
Ton/Km-Hr (Max)	0.40	0.41	0.44	0.44
TPr (Seg)	7.002	8.168	5.980	6.43

Tabela V.9: Problema 6, ($n = 32$, $m = 63$), Origem=Aterro (Jangurussu), Destino=Aterro (Jangurussu), Fig. 34

BALANCEADO	VDP	VDP	VDP	VDP
Problema	7	8	9	10
Circuito	007-002	007-002	007-002	007-002
Qv (Ton)	6.5	7.5	6.5	7.5
TSt (Hs)	8h 10min	8h 10min	9h 10min	9h 10min
NVg	3	2	3	2
Km	97.90	80.46	100.04	91.98
TSa	8h 10min	8h 10min	9h 06min	9h 10min
% Servido	73.31	85.87	100	92.16
Ton/Km-Hr (Min)	0.01	0.01	0.01	0.01
Ton/Km-Hr (Max)	0.02	0.02	0.02	0.02
TPr (Seg)	1h 32min	1h 34min	1h 40min	1h 37min

Tabela V.10: Problemas com rede real ($n = 2900$, $m = 5900$), Origem=Garagem, Destino=Aterro (Jangurussú) na malha viária real da Zona 007 em Fortaleza/CE

Os valores de TPr foram calculados juntamente com a impressão da rota de cada veículo. O propósito deste exemplo é apenas de avaliar o desempenho do processo global na obtenção de respostas para efeito de análise.

