

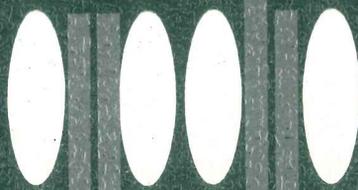
# INVESTIGAÇÃO OPERACIONAL

Junho 1992

Numero 1

Volume 12

Publicação Científica da



Associação Portuguesa para o Desenvolvimento  
da Investigação Operacional

# INVESTIGAÇÃO OPERACIONAL

Propriedade:

APDIO — Associação Portuguesa para o Desenvolvimento  
da Investigação Operacional

## ESTATUTO EDITORIAL

*«Investigação Operacional», órgão oficial da APDIO cobre uma larga gama de assuntos reflectindo assim a grande diversidade de profissões e interesses dos sócios da Associação, bem como as muitas áreas de aplicação da I. O. O seu objectivo primordial é promover a aplicação do método e técnicas da I. O. aos problemas da Sociedade Portuguesa.*

*A publicação acolhe contribuições nos campos da metodologia, técnicas, e áreas de aplicação e software de I. O. sendo no entanto dada prioridade a bons casos de estudo de carácter eminentemente prático.*

---

Distribuição gratuita aos sócios da APDIO

# INVESTIGAÇÃO OPERACIONAL

Volume 12 - nº 1 - Junho 1992

Publicação semestral

Editor Principal: Joaquim J. Júdice  
Universidade de Coimbra

## Comissão Editorial

M. Teresa Almeida  
Inst. Sup. Economia e Gestão

Jaime Barceló  
Univ. de Barcelona

Paulo Barcia  
Univ. Nova de Lisboa

Isabel Branco  
Univ. de Lisboa

António Câmara  
Univ. Nova de Lisboa

C. Bana e Costa  
Inst. Superior Técnico

M. Eugénia Captivo  
Univ. de Lisboa

Jorge O. Cerdeira  
Inst. Sup. de Agronomia

João Clímaco  
Univ. de Coimbra

J. Dias Coelho  
Univ. Nova de Lisboa

J. Rodrigues Dias  
Univ. de Évora

Laureano Escudero  
IBM, Espanha

J. Soeiro Ferreira  
Univ. do Porto

J. Fernando Gonçalves  
Univ. do Porto

Clóvis Gonzaga  
Univ. Fed., Rio Janeiro

Luís Gouveia  
Univ. de Lisboa

Rui C. Guimarães  
Univ. do Porto

J. Assis Lopes  
Inst. Superior Técnico

N. Maculan  
Univ. Fed., Rio Janeiro

Ernesto Q. Martins  
Univ. de Coimbra

Vladimiro Miranda  
Univ. do Porto

J. Pinto Paixão  
Univ. de Lisboa

M. Vaz Pato  
Inst. Sup. Economia e Gestão

Celso Ribeiro  
Univ. Católica, Rio Janeiro

A. Guimarães Rodrigues  
Univ. do Minho

Mário S. Rosa  
Univ. de Coimbra

J. Pinho de Sousa  
Univ. do Porto

L. Valadares Tavares  
Inst. Superior Técnico

Isabel H. Themido  
Inst. Superior Técnico

B. Calafate Vasconcelos  
Univ. do Porto

José M. Viegas  
Inst. Superior Técnico

A Revista "INVESTIGAÇÃO OPERACIONAL" está registada na Secretaria de Estado da Comunicação Social sob o nº 108335.

Esta Revista é distribuída gratuitamente aos sócios da APDIO. As informações sobre inscrições na Associação, assim como a correspondência para a Revista devem ser enviadas para a sede da APDIO - Associação Portuguesa para o Desenvolvimento da Investigação Operacional - CESUR, Instituto Superior Técnico, Av. Rovisco Pais, 1000 Lisboa.

Este Volume foi subsidiado por :

**Instituto Nacional de Investigação Científica (INIC)**

**Junta Nacional de Investigação Científica e Tecnológica (JNICT)**

**Fundação Calouste Gulbenkian**

Para efeitos de dactilografia e composição, foram utilizados equipamentos gentilmente postos à disposição pelo CEAUL (DEIO - Faculdade de Ciências de Lisboa).

Assinatura: 5.000\$00

# ESTUDO DA EVOLUÇÃO PREVISÍVEL DO PROCESSO DE SUBSTITUIÇÃO DE VEÍCULOS LIGEIRO NO MERCADO PORTUGUÊS

**Jorge Freire de Sousa**  
**Rui Campos Guimarães**  
GEIN - Secção de Gestão e Engenharia Industrial  
Faculdade de Engenharia da Universidade do Porto  
Rua do Bragas, 4099 Porto

## Abstract

This article presents a vehicle replacement study based upon operating and replacement cost data from the Portuguese and the French car markets. The French market is taken to be indicative of what is likely to happen in Portugal in the foreseeable future.

The analysis was based both on deterministic and stochastic models and the results derived from these models are compared and used to show the effect of several factors on the replacement policies, namely:

- (i) the characteristics of different models within the same range,
- (ii) the car range, and
- (iii) the structure of operating and replacement costs, which is a function of the stage of development of each car market.

A global perspective of how replacement policies are likely to evolve in the Portuguese market is presented on the basis of the results derived in this study, in particular those mentioned in (iii).

## Resumo

Neste artigo descreve-se um estudo relativo ao processo de substituição de veículos ligeiros. O estudo foi efectuado com base nos custos de operação e de substituição de tais veículos nos mercados português e francês, sendo este último utilizado como cenário indicativo da evolução do mercado nacional.

Os resultados apresentados, que derivam da aplicação de modelos determinísticos e estocásticos, são obtidos tendo em conta o efeito de diversos factores, designadamente:

- (i) o tipo de veículo, dentro da mesma gama;
- (ii) o tipo de gama dos veículos; e
- (iii) a composição da estrutura de custos, que depende do grau de desenvolvimento do mercado automóvel.

Com base nestes resultados, particularmente os respeitantes ao ponto (iii), obtém-se uma perspectiva da evolução do processo de substituição de veículos ligeiros em Portugal.

## Keywords

Vehicle Replacement, Deterministic Models, Stochastic Models, Market Dynamics.

## 1 - Introdução

O estudo descrito neste artigo insere-se num projecto mais geral de concepção de um Sistema de Apoio à Decisão (SAD) para a rede de concessionários da Renault e para gestores de frotas de veículos ligeiros. Tal sistema tem como objectivo o aconselhamento aos clientes dos concessionários e aos gestores de frotas no que diz respeito ao andamento dos custos de operação, em particular os de manutenção e reparação, e às decisões de substituição dos seus veículos; simultaneamente, e a título complementar, pretende-se que o sistema possa auxiliar os concessionários no estabelecimento de preços

de retoma e de revenda de veículos usados e na definição das condições dos contratos de assistência.

Neste artigo apresenta-se uma análise do processo de substituição de veículos, que é fundamentada nos resultados obtidos com base em modelos determinísticos e estocásticos. A análise efectuada põe em evidência a forma como aquele processo é afectado pelos seguintes factores: o tipo de veículo dentro da mesma gama, o nível de gama do veículo e a composição da estrutura de custos, que depende claramente do grau de desenvolvimento do mercado automóvel. Com base no mercado francês, que representa um estado mais evoluído do mercado nacional, obtêm-se indicadores quantificados da transformação previsível do processo de substituição no nosso País.

## 2 - Os factores envolvidos no processo de decisão

A definição da eficiência económica de uma política de substituição de veículos assenta, naturalmente, nos custos de operação e de substituição destes ao longo do tempo.

São considerados custos de operação todos aqueles que decorrem da utilização de cada veículo ao longo do seu ciclo de vida. Para efeito da sua modelização, os custos de operação devem ser classificados tendo em conta os dois critérios seguintes (Sousa, Leite e Guimarães (1990)):

- (i) a sua natureza, que determina a forma como evoluem ao longo da vida de um veículo (das diferentes parcelas dos custos de operação, há algumas que só dependem dos quilómetros percorridos, outras que só dependem da idade e outras ainda que dependem dos quilómetros e da idade); e
- (ii) o tipo de análise que é imposto pelos dados disponíveis (de facto, enquanto nuns casos os custos são caracterizáveis pela via da análise estatística, noutros há que recorrer à experiência oficial ou às indicações do construtor).

Da decomposição efectuada com base nestes critérios devem resultar parcelas mutuamente exclusivas. Apresentam-se seguidamente as componentes consideradas neste estudo:

- consumo de carburante (CG);
- lubrificação do motor (CL);
- desgaste de pneus (CP);
- manutenção e reparação geral (CMR);
- conservação da carroçaria (CCC);
- seguro obrigatório (CSEG);
- imposto de selo (CIS).

Para além dos custos de operação, há que ter em conta os custos de substituição dos veículos. O custo associado a cada substituição, admitindo que um veículo é sempre substituído por um outro novo da mesma gama, não é mais do que a diferença entre o preço de aquisição do veículo novo e o preço de venda do veículo usado. Este é função

de diversos factores entre os quais se salientam o preço de um veículo novo equivalente, a sua idade, o seu estado geral de conservação e a relação procura/oferta no mercado de veículos usados.

### **3 - Modelização dos custos de manutenção e reparação geral.**

#### **Caracterização da amostra utilizada**

De todos os custos enunciados na secção anterior os associados à manutenção e reparação geral são, sem dúvida, aqueles que constituem a maior fonte de variabilidade no comportamento dos custos de veículo para veículo, dentro da mesma gama. Além disso, pesem embora as dificuldades associadas à sua recolha e tratamento, são estes os custos relativamente aos quais é ainda assim possível encontrar um número de intervenções suficiente para se proceder à sua modelização estatística.

O estudo descrito neste artigo foi efectuado com base numa amostra de cerca de 2350 intervenções que cobriram o período de 1980/1990 de 158 veículos ligeiros pertencentes a particulares. Apenas foram incluídos veículos em relação aos quais havia a confiança de que tinham sido assistidos praticamente sempre na mesma oficina, tendo sido consultados para este fim três concessionários, dois de Lisboa e um do Porto. Procurou-se que os veículos cobrissem uma vasta gama de regimes de utilização, tendo-se verificado taxas de utilização que variam aproximadamente entre 5000 e 45000 Km/ano.

Foram seleccionados em proporções idênticas veículos de três modelos, sendo dois deles da gama baixa (que cobre a faixa de penetração no mercado mais larga em Portugal) e um da gama média-alta. Os objectivos de tal escolha podem ser sintetizados do seguinte modo:

- (i) analisar, em modelos sucessivos da mesma gama, as alterações introduzidas do ponto de vista tecnológico;
- (ii) estudar a forma como as modificações introduzidas se refletem sobre os custos de manutenção e reparação; e
- (iii) analisar as diferenças entre as estruturas de evolução dos custos de manutenção e reparação de veículos de duas gamas distintas.

Os custos cobertos pela garantia foram excluídos da análise.

O efeito da inflação foi eliminado pela conversão dos custos a uma base de preços comum (Janeiro de 1990).

Recorreu-se, para efeitos de modelização, à técnica de regressão linear múltipla, ensaiando modelos em que se combinavam diversas variáveis explicativas, para diferentes durações (tempo) e extensões (quilómetros percorridos) dos períodos de agregação dos custos.

O modelo utilizado para efeitos deste estudo foi o que relaciona os custos incorridos em períodos sucessivos de 15000 km com a idade média dos veículos nos respectivos períodos. A título de exemplo, apresenta-se na Figura 1 (i) a relação ajustada, bem como

as observações dispersas em torno dela, para os veículos de um dos modelos de gama baixa. A heterogeneidade da variância dos desvios, que aumenta com o valor esperado dos custos, aconselhou a adoção de uma transformação logarítmica da variável dependente. O modelo finalmente obtido apresenta-se na Figura 1 (ii) (ainda para a mesma amostra de veículos), tendo-se verificado que, neste caso, os desvios tinham uma variância homogênea (teste de Bartlett (1937)) e uma distribuição normal (teste de Lilliefors (1967)). Conclusões idênticas foram obtidas para as duas restantes amostras de veículos estudados.

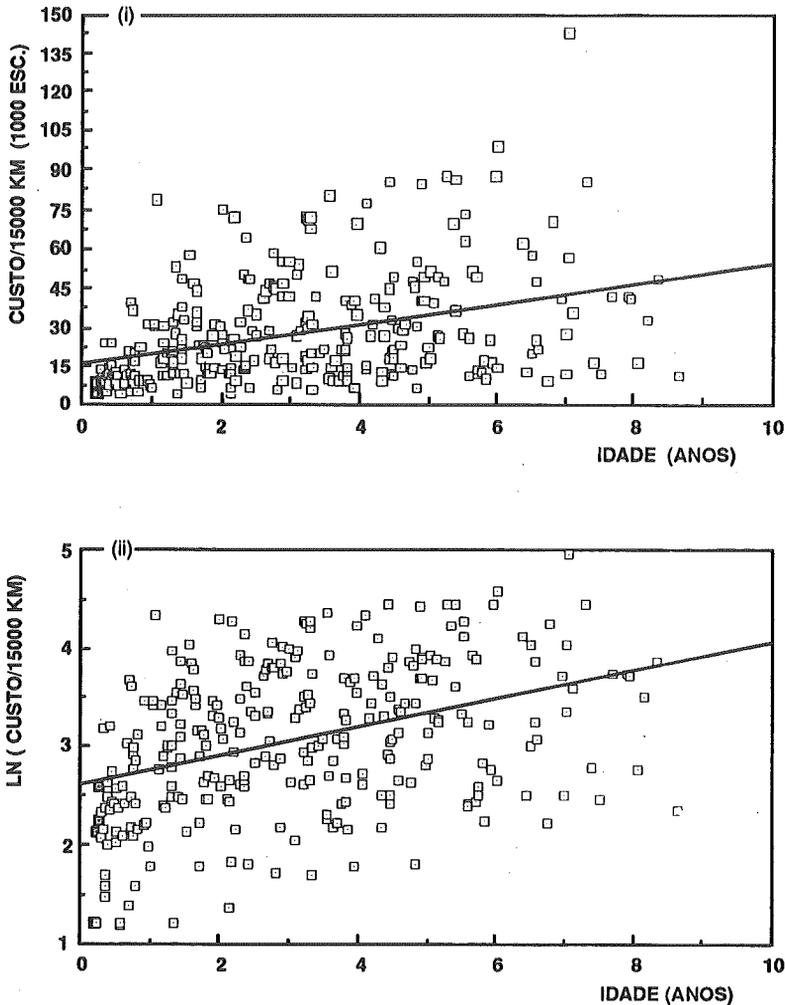


Figura 1: Custos de manutenção e reparação geral por 15000 Km e respectivo logaritmo em função da idade média dos veículos, para um dos modelos estudados

Estes resultados são consistentes com aqueles que foram anteriormente apresentados por Leite e Guimarães (1987) para um veículo da gama baixa de outra marca, bem como com os que se obtiveram neste estudo para veículos da mesma marca mas de outras gamas. Globalmente, estes resultados sugerem que o tipo de modelo aqui descrito é aplicável à generalidade dos veículos ligeiros, qualquer que seja a sua marca ou gama.

#### **4 - Modelização dos restantes custos**

As restantes parcelas dos custos de operação e os custos de substituição foram modelizados de forma determinística, de acordo com os procedimentos descritos em Leite e Guimarães (1987, 1989).

#### **5 - Modelos de substituição adoptados no estudo**

Apesar de se reconhecer que os factores de natureza económica não são os únicos que influenciam os processos de substituição, nos modelos considerados neste estudo apenas foram contemplados tais factores. A integração de factores de natureza não económica pode ser efectuada de acordo com o procedimento descrito em Leite e Guimarães (1989).

##### **5.1. Modelo determinístico**

O modelo que será apresentado nesta secção obedece, como aliás os restantes modelos determinísticos estudados, a alguns pressupostos básicos que se referem seguidamente:

- (i) os veículos usados são sempre substituídos por veículos novos da mesma gama;
- (ii) todos os veículos são usados à mesma taxa,  $\gamma$  (Km/ano); e
- (iii) os custos de operação e substituição são considerados determinísticos e contínuos.

Saliente-se que, de acordo com o pressuposto (ii), a quilometragem ( $x$ ) e a idade ( $t$ ) tornam-se convertíveis uma na outra ( $x = \gamma.t$ ) e que, deste modo, todos os custos de operação podem ser expressos como função de qualquer uma destas variáveis. Saliente-se ainda que, ao modelizar de forma determinística tais custos, a variabilidade que se verifica no seu comportamento não é tomada em conta, admitindo-se que evoluem sempre de acordo com um padrão "médio" (por exemplo, no caso da Figura 1 (ii), admite-se que os logaritmos dos custos de manutenção e reparação geral evoluem segundo a recta ajustada, para todos os veículos).

Os modelos determinísticos conduzem a políticas "de vida económica", segundo as quais todos os veículos do mesmo modelo submetidos a condições idênticas de utilização são substituídos com a mesma idade. A especificação da idade óptima de substituição depende do critério adoptado e, como refere Christer (1984), "cada critério tem os seus

méritos e os seus defeitos e constitui claramente um compromisso no qual está considerado o juízo do analista".

O critério adoptado neste estudo foi o da minimização do custo global actualizado incorrido ao longo de um número infinito de ciclos de substituição.

Se cada veículo for substituído sempre com a idade T, o custo global actualizado vem dado por:

$$\begin{aligned} CGA(T) &= \sum_{n=1}^{\infty} CGA_n(T) \cdot r^{nT} = \left[ \int_0^T c(t) \cdot r^t dt + CS(T) \cdot r^T \right] \cdot (1 + r + r^2 + \dots) \\ &= \frac{\int_0^T c(t) \cdot r^t dt + CS(T) \cdot r^T}{1 - r^T} \quad (r < 1) \end{aligned}$$

onde  $r = 1 / (1 + j / 100)$  representa o factor de actualização anual do capital e  $j$  é a taxa real de valorização anual do capital (em condições normais,  $j > 0$  e  $r < 1$ );  $c(t)$  é a função densidade do custo de operação e  $c(t) \cdot dt$  representa o custo incorrido entre  $t$  e  $t + dt$  e  $CS(T)$  é o custo de substituição do veículo quando a sua idade é  $T$ .

## 5.2. Modelo estocástico

A hipótese de que as variáveis incontrolláveis evoluem de forma determinística foi introduzida com o objectivo de simplificar os modelos descritos na secção anterior. Particularmente no que diz respeito aos custos de manutenção e de reparação, esta hipótese é, como se pode ver na Figura 1, manifestamente inadequada. O reconhecimento desta limitação conduziu ao desenvolvimento de modelos estocásticos.

Analisados os dados relativos aos custos de manutenção e reparação, agregados para cada veículo em períodos de 15000 Km, foram obtidas funções densidade de probabilidade de custo (distribuição lognormal) correspondentes às diferentes idades dos veículos.

O modelo de Jardine et al (1976), adoptado neste estudo, propõe que cada veículo seja inspeccionado periodicamente, normalmente no início de cada ano, com o objectivo de se estimar quais os custos de manutenção a incorrer nesse ano. Se tais custos forem inferiores a um determinado Custo Limite Anual de Manutenção (CLAM), o veículo é mantido em operação por mais um ano; no caso contrário, opta-se pela sua substituição. O valor do CLAM, que é função da idade do veículo, é calculado recorrendo a um modelo de programação dinâmica.

A política de substituição é estabelecida com base num horizonte temporal pré-fixado (de  $H$  anos), findo o qual o veículo então em uso é substituído.

O critério utilizado é o da minimização do valor esperado do custo global actualizado de operação e substituição ao longo do horizonte temporal  $H$ . O cálculo dos CLAM,  $L_i(k)$ , é efectuado recursivamente, considerando  $k$  sucessivamente igual a  $H-1, H-2, \dots, 1$ ,

fazendo  $i$  (idade do veículo) igual a  $1, 2, \dots, i_{\max}$  anos para cada  $k$ , e minimizando a expressão seguinte:

$$\begin{aligned} CGA_k(i) &= \left[ m_i(L_i) \cdot r^{1/2} + CGA_{k+1}^*(i+1) \cdot r \right] \cdot F_i(L_i) \\ &+ \left[ CS_i + \int_0^{+\infty} x \cdot f_0(x) dx r^{1/2} + CGA_{k+1}^*(1) \cdot r \right] \cdot [1 - F_i(L_i)] \\ &= A \cdot F_i(L_i) + B \cdot [1 - F_i(L_i)] \end{aligned}$$

onde  $x$  representa o custo de operação do veículo, incorrido no ano em causa, ou seja, entre as idades  $i$  e  $i+1$ ,  $f_i(x)$  é a função densidade de probabilidade do custo  $x$  e  $F_i(L_i)$  representa a probabilidade de o custo limite  $L_i$  não ser excedido pelo custo anual de

operação,  $x$ , vindo dada por  $F_i(L_i) = \int_0^{L_i} f_i(x) dx$ .

Na expressão de  $CGA_k(i)$ ,  $A$  é a soma de duas parcelas, das quais a primeira representa o valor esperado do custo actualizado de operação do veículo no ano  $k$ , quando não é atingido o limite  $L_i$  (admite-se que tais custos ocorrem a meio do ano), e a segunda representa o custo actualizado óptimo de operação do veículo, desde o ano  $k+1$  até ao fim do horizonte temporal.

Ainda na expressão de  $CGA_k(i)$ ,  $B$  é obtido somando o custo da substituição imediata do veículo, o valor esperado do custo actualizado de operação do veículo novo no ano  $k$  e o custo actualizado óptimo de operação do veículo novo desde o ano  $k+1$  até ao fim do horizonte temporal.

## 6 - Resultados obtidos pela aplicação do modelo determinístico

Os resultados que a seguir se apresentam e comentam foram obtidos tomando em consideração o efeito de diversos factores. Cada um destes factores merecerá uma análise particular em cada uma das secções seguintes.

### 6.1. Tipo de veículo, dentro da mesma gama

Foram estudados dois modelos de veículos da gama baixa, que diferem de 5 anos, relativamente ao seu período de lançamento. Designemos por modelo 1 o modelo mais antigo e por modelo 2 o modelo mais recente. Os resultados obtidos estão sintetizados no Quadro 1.

Taxa de utilização $\gamma$ (1000 km/ano)	Idade óptima de substituição		Quilometragem óptima de substituição		Custo global actualizado	
	$i_{op}$ (anos)		$x_{op}$ (1000 km)		CGA (contos)	
	MOD.1	MOD.2	MOD.1	MOD.2	MOD.1	MOD.2
10	10.3	8.0	102.5	80.0	5516	5660
15	8.3	6.7	125.0	100.0	7097	7157
20	6.9	6.0	137.5	120.0	8658	8630
30	6.0	5.3	180.0	157.5	11809	11645

Quadro 1: Resultados obtidos recorrendo ao modelo determinístico, para dois modelos de veículos da gama baixa

Verifica-se, pela análise do quadro, que o modelo mais recente apresenta valores mais baixos para a sua vida económica óptima.

Procurou-se, no sentido de permitir uma análise mais fundamentada dos resultados, estudar a forma como evolui o custo global actualizado em função da quilometragem efectuada pelo veículo até à data da sua substituição. Para a situação em que os veículos cumprem uma taxa anual de 15000 Km, apresentam-se na Figura 2 os resultados sob a forma gráfica.

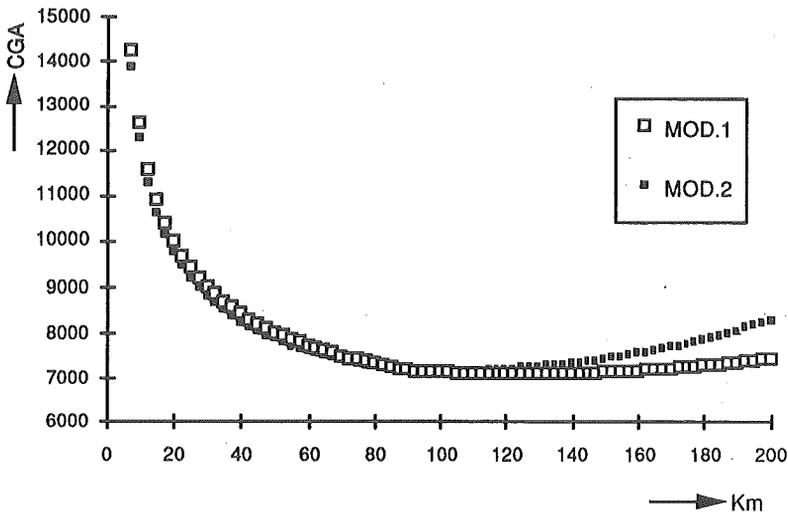


Figura 2: Curvas do custo global actualizado em função da quilometragem, para os modelos 1 e 2, quando os veículos cumprem uma taxa anual de 15000 Km

A primeira observação que o perfil das curvas sugere é a de que a grandes variações na quilometragem correspondem pequenas variações no valor do custo global actualizado respectivo. Este facto vem, num certo sentido, relativizar a importância da determinação precisa do valor óptimo, uma vez que este passa a assumir um carácter mais indicativo de ordem de grandeza.

A segunda observação é a de que o modelo mais recente apresenta valores mais baixos de custo para valores de quilometragem que se estendem sensivelmente até aos 100 000 Km, invertendo-se esta tendência, de uma forma significativa, após os 140 000 Km. Estes resultados sugerem que modelos equivalentes mais recentes são mais fiáveis durante o período de vida útil normal de um veículo da gama em causa; no entanto, após este período, os custos aumentam mais rapidamente nos modelos mais recentes, podendo associar-se este facto à progressiva redução que se tem vindo a verificar no ciclo de vida dos veículos ligeiros.

## 6.2. Tipo de gama dos veículos

O modelo 3 designa o veículo de gama média-alta e o modelo 2 continua a designar o modelo de gama baixa já referido na secção anterior. Os resultados relativos a estes dois modelos são apresentados no Quadro 2.

Taxa de utilização $\gamma$ (1000 km/ano)	Idade óptima de substituição $i_{op}$ (anos)		Quilometragem óptima de substituição $x_{op}$ (1000 km)		Custo global actualizado CGA (contos)	
	MOD.2	MOD.3	MOD.2	MOD.3	MOD.2	MOD.3
10	8.0	6.5	80.0	65.0	5660	9491
15	6.7	5.8	100.0	87.5	7157	11651
20	6.0	5.1	120.0	102.5	8630	13758
30	5.3	4.2	157.5	125.0	11645	17806

Quadro 2: Resultados obtidos recorrendo ao modelo determinístico, para um modelo de veículo da gama baixa e para outro da gama média-alta

No mercado português, os veículos apresentam uma vida económica óptima tanto menor quanto mais elevado for o seu nível de gama, pelo menos no que diz respeito às gamas de maior penetração. Este facto é atribuível ao mais rápido e mais acentuado crescimento dos custos gerais de reparação e manutenção nos veículos de gama alta.

Apresentam-se na Figura 3, para uma taxa de utilização anual de 15000 Km, as curvas do custo global actualizado em função da quilometragem, para os dois modelos estudados, confirmando-se, para quaisquer valores de quilometragem, as diferenças significativas de custo expressas nas duas últimas colunas do Quadro 2.

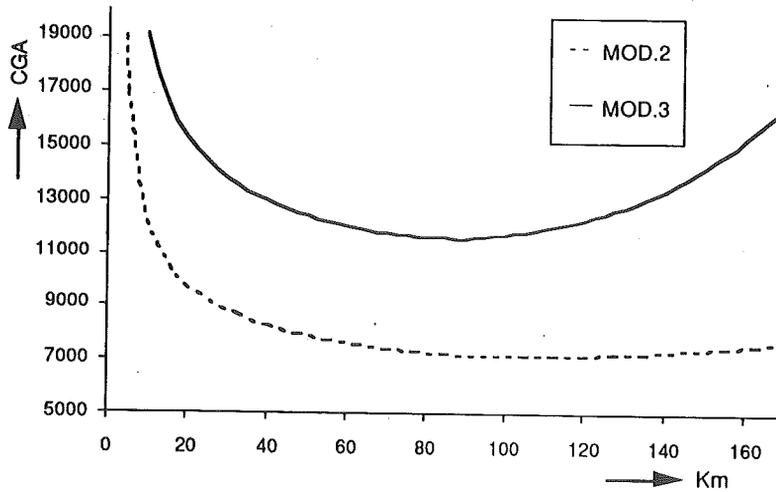


Figura 3: Curvas do custo global actualizado em função da quilometragem para os modelos 2 e 3 quando os veículos cumprem uma taxa anual de 15000 Km

### 6.3. Estrutura dos custos

Dependendo do grau de desenvolvimento do mercado automóvel, a estrutura dos custos que afectam as decisões de substituição pode apresentar configurações substancialmente diferentes.

Pretende-se nesta secção, utilizando os casos de Portugal e de França, analisar a influência que diferentes tipos de estrutura de custos têm sobre as políticas óptimas de substituição. A expectativa é a de que, através desta análise, se possam confirmar ou rejeitar algumas conjecturas que os resultados apresentados nas secções precedentes suscitam.

Refiram-se, em primeiro lugar, alguns factos importantes para a discussão dos resultados:

- (i) os preços de venda ao público dos carros são, em termos relativos (e, em alguns casos, mesmo em termos absolutos) mais elevados em Portugal do que em França;
- (ii) a desvalorização dos veículos acentua-se mais rapidamente em França do que em Portugal;
- (iii) os custos de manutenção são, relativamente aos custos de substituição, mais significativos em França do que em Portugal;
- (iv) o peso relativo dos custos de mão-de-obra nos custos globais de manutenção é maior em França.

Acrescente-se ainda que, considerando um cabaz de intervenções mecânicas correntes sobre 100 000 Km de utilização de um veículo (cabaz esse que serve de base à revista

francesa L'Auto Journal para o cálculo do custo por quilómetro), o custo da mão-de-obra representa em Portugal cerca de 21% do total dos custos, contra cerca de 40% em França (os pesos relativos dos custos de mão-de-obra e peças vão variando ao longo da vida de um automóvel). Refira-se por fim que, em relação às peças envolvidas nas intervenções de tal cabaz, o seu preço é em média em Portugal cerca de 77% do preço de França, enquanto que o custo/hora de mão-de-obra de manutenção automóvel representa em Portugal apenas 32% do mesmo custo em França.

A conjugação dos factores (i), (ii) e (iii) determina que, no cenário francês, os períodos óptimos de substituição sejam mais curtos do que em Portugal. Os resultados apresentados no Quadro 3 quantificam tal redução.

A primeira linha do Quadro corresponde aos resultados obtidos para o modelo mais recente da gama baixa, para uma taxa de utilização anual de 15000 Km. A segunda corresponde ao modelo da gama média-alta, para 20000 Km anuais. A selecção destas taxas anuais de quilometragem deve-se ao facto de elas corresponderem sensivelmente aos valores médios verificados nas gamas respectivas.

Taxa de utilização $\gamma$ (1000 km/ano)	Idade óptima de substituição $i_{op}$ (anos)		Quilometragem óptima de substituição $x_{op}$ (1000 km)		Custo global actualizado CGA (contos)	
	PORT.	FRA.	PORT.	FRA.	PORT.	FRA.
15 (BG)	6.7	5.3	100.0	79.5	7157	12589
20 (MA)	5.1	4.6	102.5	92.0	13758	20638

Quadro 3: Resultados obtidos recorrendo ao modelo determinístico, para um modelo de veículo da gama baixa e para outro da gama média-alta, em Portugal e em França

Analise-se agora a Figura 4, onde se representam, para os dois países em estudo, os perfis de evolução e a importância relativa das diferentes parcelas de custo, ao longo da vida de um veículo.

Há dois aspectos que é necessário esclarecer previamente, por forma a permitir uma correcta leitura dos diagramas. O primeiro refere-se aos custos de conservação da carroçaria (CCC), presentes no caso português mas ausentes no caso francês, por não haver a informação necessária à sua inclusão. O segundo refere-se aos seguros (CSEG): apesar do custo dos seguros ser mais elevado em França do que em Portugal, a diferença entre a importância relativa de tal custo nos dois países aparece agravada nas figuras; tal agravamento deve-se ao facto de se ter considerado em Portugal apenas o custo do seguro obrigatório, enquanto em França se considerou o do seguro contra todos os riscos.

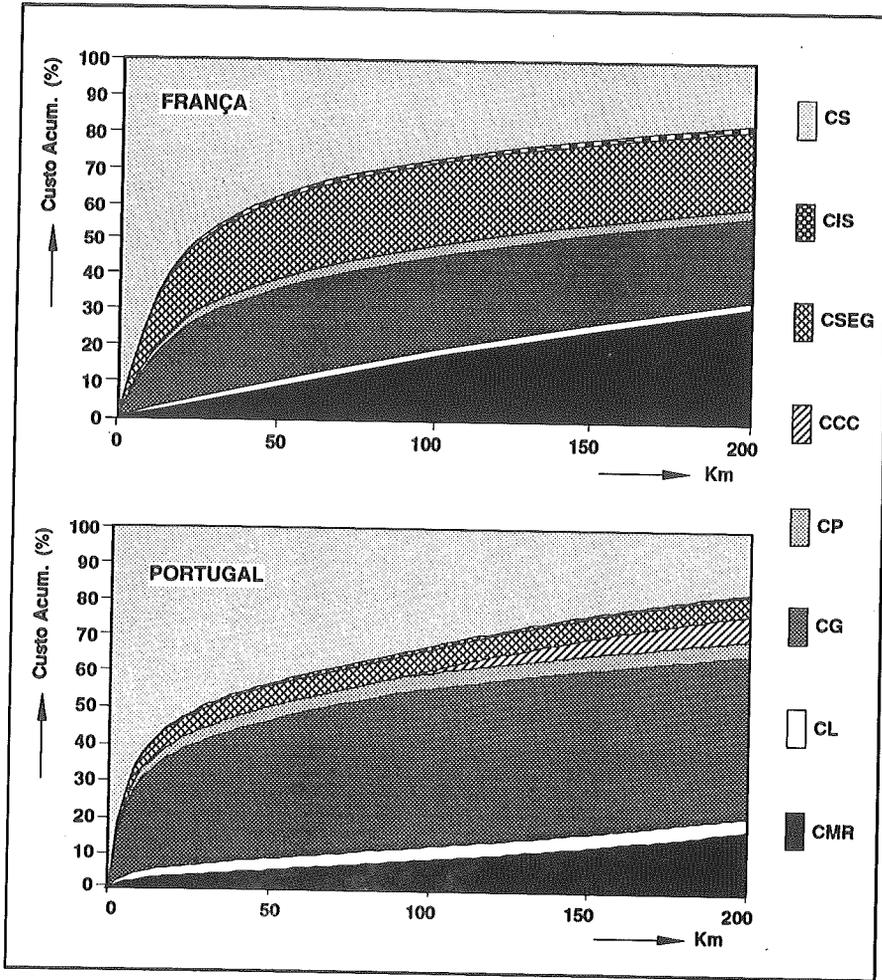


Figura 4: Perfis de evolução das diferentes parcelas de custo, ao longo da vida de um veículo, em Portugal e em França

Da observação das figuras, constata-se que os custos de manutenção e reparação geral (CMR) e os de substituição (CS) são aqueles que maior variação de peso relativo apresentam ao longo do tempo. As restantes parcelas de custo mantêm o seu peso relativo praticamente inalterável.

Por outro lado, para qualquer quilometragem que se situe para além do período inicial da vida dos veículos, é clara a menor importância dos custos de substituição em França relativamente a Portugal, verificando-se o inverso no que respeita aos custos de manutenção e reparação.

Deste modo, conclui-se que o estabelecimento da política óptima de substituição resulta fundamentalmente de um "trade-off" entre o ritmo de desvalorização do veículo e a forma como progredem os seus custos de manutenção e reparação.

**7 - Resultados obtidos pela aplicação do modelo estocástico**

**7.1. Selecção de um horizonte temporal adequado**

De acordo com o que previamente se referiu na secção 5.2., a política de substituição estabelecida pelo modelo de Jardine et al (1976) toma como base um horizonte temporal, findo o qual o veículo então em uso é substituído.

O primeiro passo do método será portanto a determinação de uma dimensão do horizonte temporal que não influencie artificialmente a sequência do processo de substituição (Sousa, Leite e Guimarães (1990)).

Para tal, optou-se por, obtidos os custos actualizados mínimos para diferentes horizontes temporais através do modelo estocástico, estender esses custos a um número infinito de horizontes. Os resultados, para o modelo da gama baixa e para  $\gamma = 15000$  Km/ano, podem ser observados na Figura 5.

O objectivo de tal procedimento é o de determinar a dimensão do horizonte temporal que conduz à minimização do valor esperado do custo total actualizado ao longo de um número infinito de horizontes.

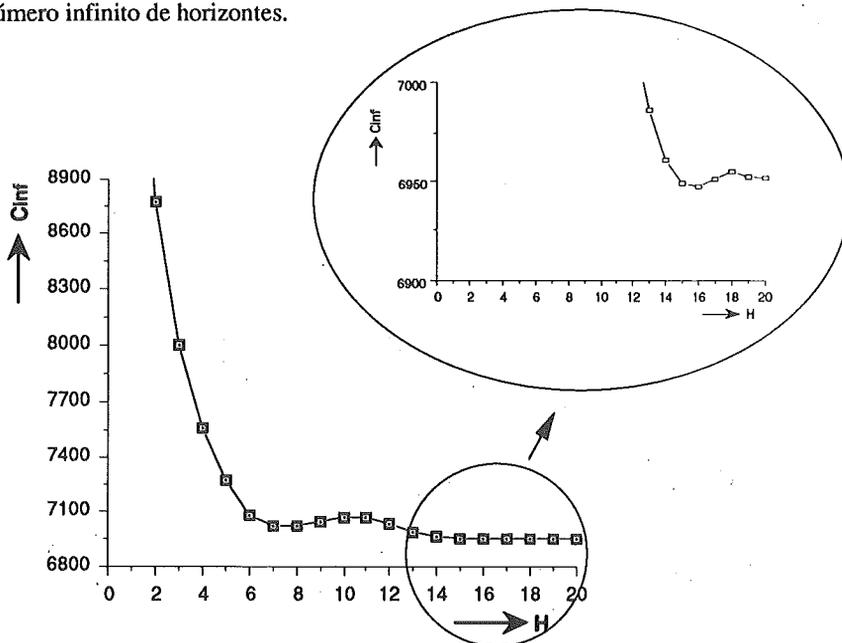


Figura 5: Curva dos custos globais actualizados mínimos (#H = ∞) para diferentes dimensões do horizonte temporal, para um veículo do modelo 2 e para  $\gamma = 15000$  Km/ano

O horizonte temporal a utilizar no processo de determinação da idade óptima de substituição, de acordo com o modelo estocástico, será assim o de  $H=16$  que representa um dos mínimos locais da função, como se pode observar na Figura 5.

Poder-se-ia usar o valor de  $H = 8$  ou, previsivelmente, o de  $H = 24$ , mas o valor escolhido representa um compromisso entre não considerar um horizonte demasiado pequeno, apontando para uma única substituição, nem um horizonte demasiado grande, que se tornaria muito pesado do ponto de vista computacional.

## 7.2. Interpretação e análise dos resultados obtidos

Para uma taxa anual de utilização de 15000 Km, comparam-se, no Quadro 4, os valores obtidos pelos modelos determinístico e estocástico, para as duas gamas de veículos em estudo.

Dado que a interpretação dos resultados é semelhante para os veículos das duas gamas, passar-se-á, nos parágrafos seguintes, a comentar apenas o caso do modelo da gama baixa.

Taxa de utilização $\gamma =$ 15000 km/ano	Idade óptima de substituição		Quilometragem óptima de substituição		Custo global actualizado	
	$i_{op}$	$E(i)_{op}$	$x_{op}$		CGA	
	(anos)		(1000 km)		(contos)	
	DET.	EST.	DET.	EST.	DET.	EST.
BG	6.7	7.9	100.0	118.5	7157	6947
MA	5.8	6.4	87.5	96.0	11651	11108

Quadro 4: Resultados obtidos pelos modelos determinístico e estocástico para um modelo de veículos da gama baixa e para outro da gama média-alta, em Portugal

Na Figura 6 pode observar-se o comportamento do custo limite anual de reparação e manutenção ao longo da vida do veículo (para o horizonte de 16 anos), bem como a evolução do valor médio das distribuições desses custos e respectiva banda, considerando um intervalo de  $\mu \pm 2\sigma$ . Para um veículo de 8 anos de idade representa-se ainda, a título de exemplo, a função densidade de probabilidade de custos/15000 Km (distribuição lognormal).

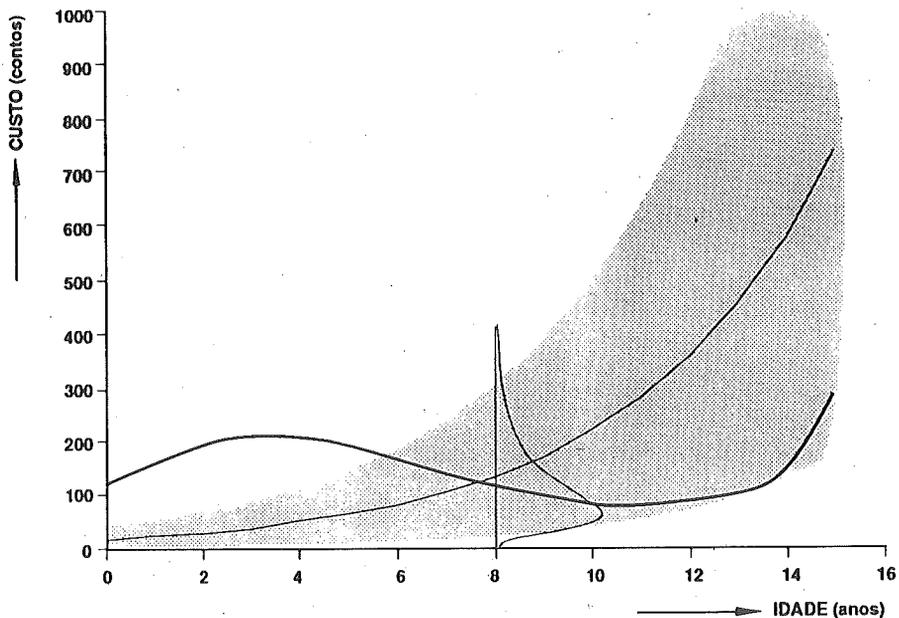


Figura 6: Evolução do custo limite anual de manutenção e do valor médio da distribuição dos custos de manutenção e reparação geral, ao longo da vida de um veículo

No início da vida do veículo, o CLAM situa-se nitidamente acima da banda do custo, o que significa que há uma probabilidade muito pequena do veículo ser substituído. A partir de  $I=6$ , o valor do custo limite entra progressivamente na banda e a probabilidade de substituição começa a aumentar significativamente. Considerando, por exemplo,  $I=13$ , a probabilidade de que se efectue a substituição é de cerca de 97.5 %, uma vez que o CLAM coincide aproximadamente com o limite inferior da banda a 95%. Note-se que esta probabilidade é condicionada ao facto de que o veículo não tenha sido substituído em anos anteriores. A probabilidade incondicional de substituição é, no entanto, muito mais baixa, uma vez que a probabilidade de o veículo ter sido substituído em anos anteriores é muito elevada.

É interessante notar ainda o reflexo no CLAM da proximidade do fim do horizonte temporal (e da obrigatoriedade de então substituir o veículo de que se dispuser). O custo limite começa a subir, na tentativa de evitar a penalização elevada correspondente à substituição de um veículo com uma idade muito baixa no fim do horizonte.

Como resultado da aplicação do modelo estocástico, obtemos assim não uma idade óptima de substituição mas antes uma distribuição da probabilidade de substituição ao longo do horizonte temporal considerado. Na Figura 7 apresentam-se as distribuições da probabilidade de substituição dos veículos.

No diagrama (i) representa-se a probabilidade condicional à não substituição em anos anteriores (interpretável directamente à luz da Figura 6) e, no diagrama (ii), a probabilidade incondicional (aquela que caracteriza melhor o processo de substituição).

Resta referir que o valor esperado da idade óptima de substituição é, no modelo estocástico, mais elevado do que a correspondente idade obtida pelo modelo determinístico. Este facto é atribuível à forma da curva dos custos globais actualizados, representada na Figura 2, que sugere que a penalização resultante de substituir o veículo antes do momento óptimo é maior do que a de adiar por algum tempo tal substituição.

Como se pode observar no Quadro 4, para  $\gamma = 15000$  Km/ano, as reduções no valor esperado do custo obtidas pela aplicação do modelo estocástico vão desde cerca de 3% para os veículos baixo de gama até cerca de 5% para os de gama média-alta.

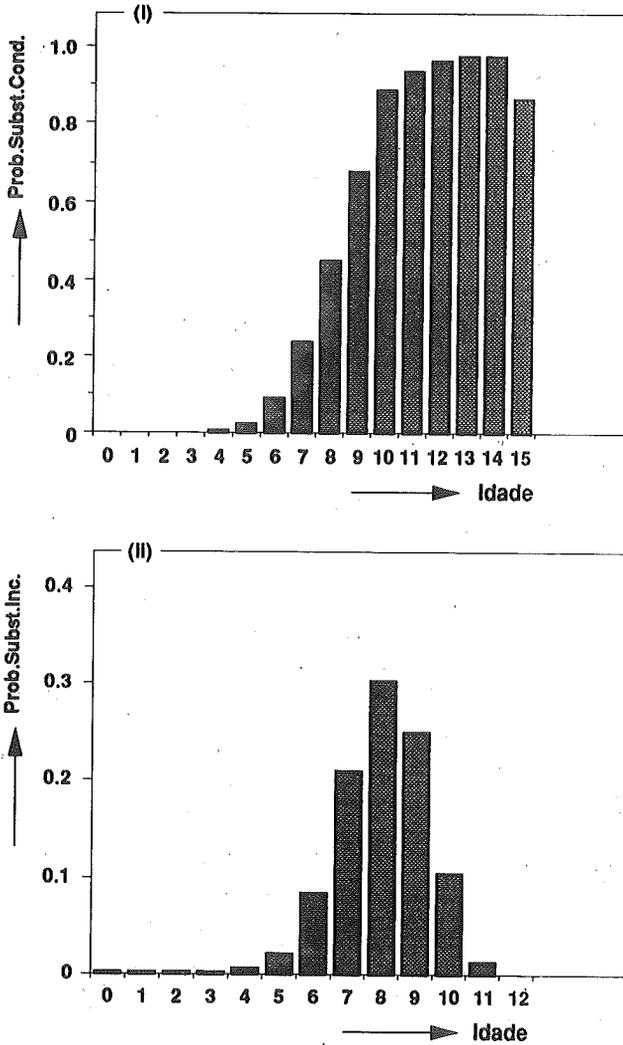


Figura 7: Distribuições de probabilidade (condicional e incondicional) de substituição de um veículo modelo 2, para um horizonte de 16 anos e  $\gamma = 15000$  Km/ano

## 8 - Conclusão

A modelização que foi objecto das secções anteriores deste artigo contribuiu para o esclarecimento do processo de substituição de veículos ligeiros e, em particular, do papel desempenhado por um conjunto de factores tidos como significativos.

Neste contexto, tomando o mercado francês como cenário indicativo da evolução do mercado nacional, justificam-se e quantificam-se as expectativas de evolução do processo de substituição em Portugal.

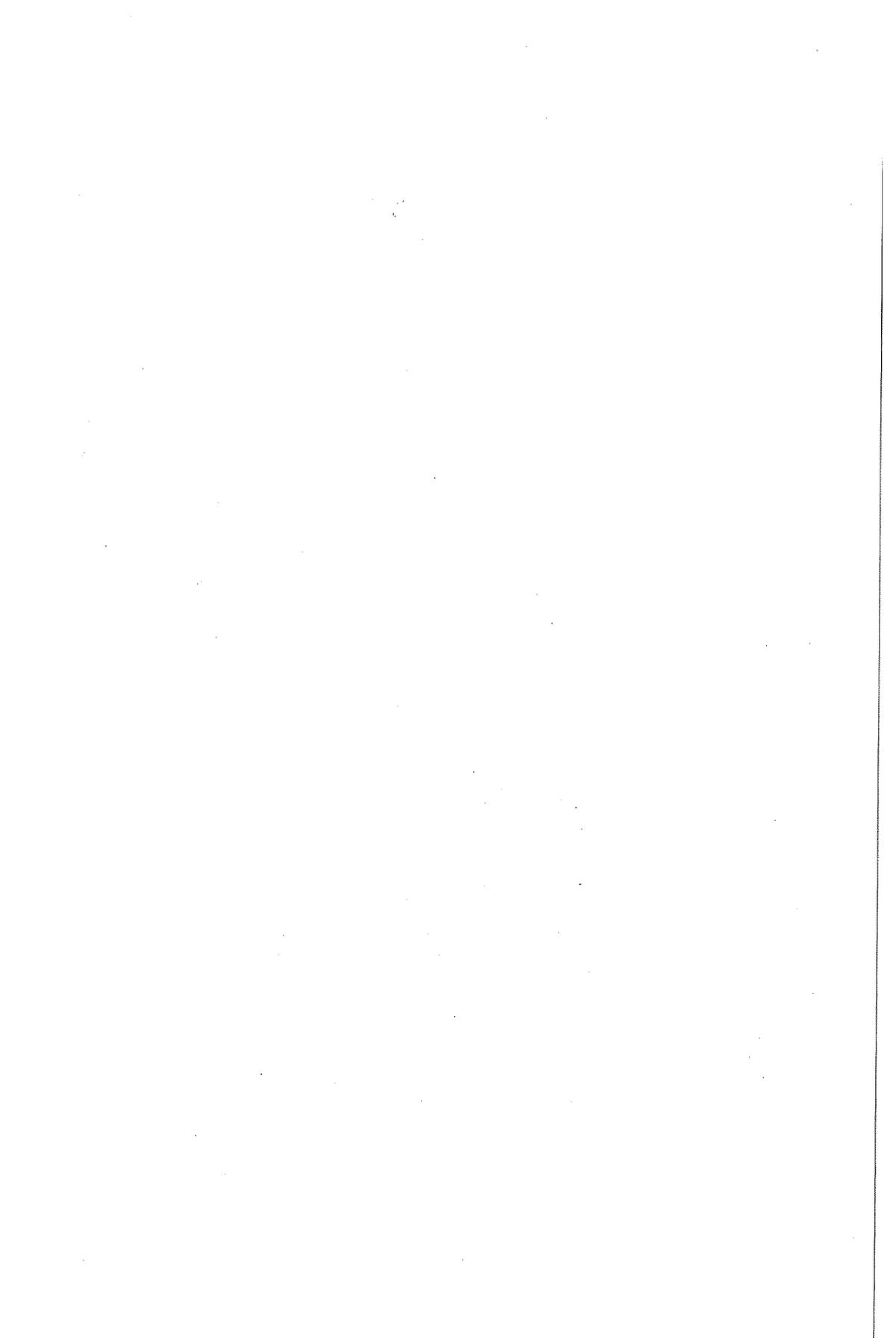
O recurso simultâneo a modelos determinísticos e probabilísticos permitiu evidenciar as vantagens da política de substituição que deriva da adopção do segundo tipo de modelos, que conduz a uma redução significativa do valor esperado do custo global de operação e substituição. Admite-se que as reduções de custo de 3 a 5%, obtidas para taxas médias de utilização dos veículos em Portugal, poderão tornar-se ainda mais significativas para utilizações mais intensivas dos veículos.

## Agradecimentos

Os autores agradecem à Régie Renault e à Renault Portuguesa o apoio concedido na realização deste projecto. Desejam, em particular, expressar os seus agradecimentos ao Senhor Engenheiro José Maria Coelho, da Renault Portuguesa, pelo interesse e disponibilidade sempre demonstrados.

## Referências

- [1] Bartlett, M. J. (1937), Properties of Sufficiency and Statistical Tests, Proc. Roy. Soc., Série A, Vol. 160, 268-282.
- [2] Christer, A. H. (1984), Operational Research Applied to Industrial Maintenance and Replacement, in Developments in Operation Research, Eglese, R. W. e G. K. Rand (editores), Pergamon Press.
- [3] Jardine, A. K. S., T. S. Goldrick e J. Stender (1976), The Use of Annual Maintenance Cost Limits for Vehicle Fleet Replacement, Proc. Inst. Mech. Eng., Vol. 190, 121-138.
- [4] Leite, J. D. e R. C. Guimarães (1987), Estabelecimento de Políticas Óptimas de Substituição de Veículos: Uma Aplicação ao Modelo Fiat 127, Relatório Interno, INEGI.
- [5] Leite, J. D. e R. C. Guimarães (1989), Análise Comparativa de Políticas Óptimas de Substituição de Veículos, Pesquisa Operacional, Vol. 9, No. 1, 79-109.
- [6] Lilliefors, H. W. (1967), On Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown, American Statistical Association Journal, Vol. 62, No. 318, 339-402.
- [7] Sousa, J. F., J. D. Leite e R. C. Guimarães (1990), A Abordagem Quantitativa na Substituição de Veículos, in Proceedings das Jornadas sobre Projecto, Manutenção e Gestão de Veículos de Transporte Colectivo de Passageiros, Porto.



## MÉTODO DE AMPLIFICAÇÕES SUCESSIVAS

**Manuel A. Matos**  
INESC Porto  
DEEC/FEUP  
Universidade do Porto

**L. Miguel Proença**  
INESC Porto

### Abstract

When dealing with multiple criteria decision problems, the Decision Maker may be confronted with a large number of nondominated alternatives, which is excessive for his limited processing capacity, although he has no difficulties in choosing among a reduced number of solutions. A possible strategy for non prescriptive decision-aid on this type of problems consists on gradually approaching the preferred alternative, without surpassing, at any stage, the information processing limits of the DM.

The Successive Amplification Method uses fuzzy clustering techniques to define a limited number of *macro-solutions*, which represent holistically the decision set. At each stage, the DM selects the most promising macro-solutions, and the actual decision set is reduced accordingly. This corresponds to a progressive amplification of the preferred area of the attribute space, until a final decision is reached. In the process, the DM has only to decide among alternatives (real or virtual).

The paper describes the method and presents an illustrative example.

### Resumo

Em problemas de decisão com critérios múltiplos, o Agente de Decisão pode ter que analisar um conjunto de soluções eficientes com um grande número de elementos, ultrapassando a sua capacidade de processamento limitada, mesmo que ele não tenha dificuldade de decidir entre um número reduzido de alternativas. A ajuda à decisão não prescritiva em relação a este aspecto corresponde a fornecer ao Agente de Decisão meios que lhe permitam aproximar-se progressivamente da solução preferida, à custa de decisões parciais que respeitem os referidos limites de capacidade de processamento.

O Método de Amplificações Sucessivas baseia-se numa representação do conjunto das alternativas através dos pontos característicos de macro-soluções, em número não superior ao limite aceitável para o Agente de Decisão, mas mantendo uma perspectiva global sobre o problema. O AD pode então seleccionar a(s) macro-soluções que lhe parece(m) mais promissor(as), o que permite a amplificação sucessiva da sua zona de preferência mais forte, à custa de uma redução progressiva do conjunto das alternativas, até à obtenção de um conjunto final, no qual é escolhida a solução preferida. Os julgamentos pedidos ao Agente de Decisão envolvem apenas selecção de alternativas (reais ou virtuais), não sendo necessários quaisquer outros parâmetros (pesos, metas, etc), que afectem a decisão final.

Na comunicação descrevem-se os fundamentos e o funcionamento do método, e apresenta-se uma aplicação de demonstração, desenvolvida em ambiente MS-DOS, incluindo um exemplo ilustrativo.

### Keywords

Decision Aid, Multiple Criteria, Multiattribute Decision Making.

## 1 - Introdução

Uma das possíveis estratégias de abordagem de problemas de decisão com critérios múltiplos envolve uma primeira fase, de geração ou identificação das soluções eficientes (ou uma sua aproximação, no caso multiobjectivo), seguida de uma fase de decisão, na qual o Agente de Decisão escolhe a solução preferida de entre as que lhe são apresentadas.

A primeira fase baseia-se no conceito de *dominação*: uma alternativa domina outra se é melhor do que ela em pelo menos um critério, não sendo pior em nenhum outro. Em consequência, são rejeitadas todas as alternativas dominadas, uma vez que não há qualquer racionalidade em preferi-las às soluções que as dominam (a não ser que o problema esteja mal formulado e haja critérios escondidos). As soluções não-dominadas (designadas também por eficientes ou óptimas de Pareto) mantêm-se em consideração. Devido à natureza conflituosa dos critérios, não é possível, em geral, obter uma solução final única como resultado deste processo, tornando-se necessária a segunda fase, na qual se faz intervir, directa ou indirectamente, a estrutura de preferências do Agente de Decisão. Essa intervenção pode assumir diversas formas, desde a construção de uma função de valor (ou função utilidade) que transforma o problema num problema operacional de maximização, até à suposição de que o Agente de Decisão pode decidir directamente sobre o conjunto de alternativas resultante da primeira fase, passando pela aplicação de regras auxiliares "racionais" (mas não incontestáveis, como é o caso da dominação).

Quando o Agente de Decisão se considera capaz de escolher directamente entre alternativas, surgem muitas vezes dificuldades associadas ao grande número de elementos do conjunto de soluções candidatas, por ser ultrapassada a capacidade de processamento limitada do Agente de Decisão [6]. A ajuda à decisão não prescritiva em relação a este aspecto corresponde a fornecer ao Agente de Decisão meios que lhe permitam aproximar-se progressivamente da solução preferida, à custa de decisões parciais que respeitem os referidos limites de capacidade de processamento.

Nas secções seguintes da comunicação descrevem-se os fundamentos e o procedimento base do Método das Amplificações Sucessivas, desenvolvido inicialmente por Matos [4], e apresenta-se uma aplicação para MS-DOS que utiliza o método num ambiente amigável baseado em janelas, menus, botões e rato. A comunicação é completada por um exemplo baseado num problema multiatributo com 33 alternativas eficientes. Para além do que se incluiu nesta introdução, não se fará qualquer exposição sobre os fundamentos e terminologia dos problemas de decisão multicritério, que podem ser vistos, por exemplo, em Matos [4].

## 2 - Método de Amplificações Sucessivas

A metodologia que se descreve nesta secção pretende ultrapassar os problemas, referidos na introdução, de tratamento de conjuntos de soluções eficientes com elevado número de elementos. Respeitando as limitações na capacidade de processamento do Agente de Decisão, o método permite, interactivamente, uma aproximação progressiva à solução preferida, mantendo uma perspectiva global sobre o problema.

### 2.1. Conceptualização

O Método de Amplificações Sucessivas baseia-se no conceito de macro-solução (introduzido por Matos [4]), definida como um conjunto impreciso (*fuzzy set*) de alternativas, e representada, para efeitos operacionais, por um ponto característico que funciona como uma alternativa virtual, "típica" de uma certa tendência de relações entre critérios. A globalidade do conjunto pode, assim, ser representada por um número limitado de macro-soluções, que são apresentadas ao AD. Este é, em consequência, sempre confrontado com escolhas entre alternativas (reais ou virtuais) no espaço dos atributos, em número não superior ao que desejar, podendo seleccionar a(s) macro-soluções que lhe pareça(m) mais promissor(as). Por sua vez, todas as alternativas reais têm um determinado grau de pertença a cada uma das macro-soluções, o que permite manter em consideração apenas as soluções cujo grau de pertença total às macro-soluções seleccionadas é superior a um certo valor (limiar de inclusão). Isto inclui alternativas "intermédias", ou seja, não claramente associadas a nenhuma macro-solução individual, mas relacionadas com a união de duas ou mais macro-soluções seleccionadas.

A repetição do processo, com determinação de novas macro-soluções que representem o conjunto reduzido e novas decisões parciais, permite a redução progressiva até um conjunto final com poucos elementos, no qual é escolhida a solução preferida. A possibilidade de escolha múltipla em cada decisão parcial exclui a incómoda obrigação, para o AD, de tomar decisões "forçadas", para poder prosseguir. A velocidade de amplificação da zona preferida pode ser regulada pelo limiar de inclusão.

Operacionalmente, o Método de Amplificações Sucessivas baseia-se num processo interno de *fuzzy clustering* (agrupamento impreciso), onde as macro-soluções correspondem aos agrupamentos imprecisos, e os seus pontos característicos são os protótipos (centróides) dos agrupamentos.

### 2.2. Macro-soluções

A modelização revela-se aplicável tanto ao caso multiatributo, em que o conjunto das alternativas seja de grande dimensão, como a problemas multiobjectivo, em princípio considerando o conjunto dos pontos extremos eficientes (é possível, também, "enriquecer" este conjunto com alguns pontos eficientes interiores). Em qualquer dos casos, falar-se-á, formalmente, de um conjunto

$$Z = \{z_1, z_2, \dots, z_n\}$$

de  $n$  alternativas  $z_i = (z_{i1} z_{i2} \dots z_{im})$  definidas no espaço de atributos de dimensão  $m$ . Devido às características do método, é irrelevante que o problema seja de maximização ou de minimização.

O fundamento da modelização desenvolvida é, como se disse, a consideração de macro-soluções que representem, numa forma agregada e estruturada, o conjunto  $Z$ . O número  $s$  de macro-soluções não deve exceder o "limite de discriminação"  $n_{AD}$  do Agente de Decisão, ou seja, o número máximo de alternativas que o AD aceita considerar simultaneamente.

Uma macro-solução  $M_i$  é definida como um conjunto impreciso de soluções:

$$M_i = \{ (z_k, u(z_k, M_i)) \mid z_k \in Z \} \quad i = 1 \dots s \leq n_{AD}$$

onde  $u: Z \rightarrow [0, 1]$  designa o grau de pertença da alternativa  $z_k$  em relação à macro-solução  $M_i$ , representado usualmente por:

$$u_{ik} = u(z_k, M_i)$$

Definem-se também pontos característicos das macro-soluções,  $s_i$  ( $i = 1 \dots s$ ), no espaço dos atributos. Estes pontos não terão necessariamente que pertencer a  $Z$ .

Saliente-se que o conceito de macro-solução tem um carácter geral para este tipo de problemas, não se esgotando no Método de Amplificações Sucessivas. Supondo definidos os graus de pertença e os pontos característicos, é possível, por exemplo, representar compactamente o conjunto  $Z$  de várias formas:

- (a) O conjunto  $S = \{s_1 s_2 \dots s_s\}$  dos pontos característicos das macro-soluções.
- (b) Núcleos de nível  $\alpha$  para macro-soluções  $M_i$  seleccionadas, dados por:

$$M_i^\alpha = \{z_k \mid z_k \in Z \text{ e } u_{ik} \geq \alpha\}$$

- (c) Uniões de dois ou mais núcleos de nível  $\alpha$ .

No Método de Amplificações Sucessivas é usada a primeira hipótese indicada, enquanto que as hipóteses (b) e (c) são utilizadas em outros modelos (ver [4] e [5]).

### 2.3. Descrição do método

Como já se referiu, o método baseia-se na identificação das macro-soluções (e dos seus pontos característicos) directamente a partir do conjunto  $Z$ . Essa identificação permite apresentar ao Agente de Decisão uma representação reduzida do conjunto  $Z$ , constituída pelos pontos característicos. Sucessivas aplicações do procedimento sobre subconjuntos de  $Z$ , derivados a partir de decisões parciais, permitem focar as zonas preferidas de forma progressivamente mais nítida, até à definição de um conjunto final de dimensão reduzida. A identificação das macro-soluções como conjuntos imprecisos, supõe ainda, como condições naturais:

- (a) A totalidade de pertença de qualquer solução  $z_k$  em  $Z$  é igual à unidade:

$$\sum_i u_{ik} = 1 \quad k = 1 \dots n$$

- (b) Não existem macro-soluções vazias (ou absorventes):

$$0 < \sum_k u_{ik} < 1 \quad i = 1 \dots s$$

As condições anteriores, conjugadas com a condição de normalização dos graus de pertença ( $u_{ik} \in [0,1]$ ,  $i = 1 \dots s$ ,  $k = 1 \dots n$ ), correspondem à definição de uma partição imprecisa do conjunto  $Z$ . Em consequência, os protótipos dos agrupamentos serão os pontos característicos das macro-soluções, podendo utilizar-se, por exemplo, o algoritmo de agrupamento impreciso conhecido por fuzzy-ISODATA (ou fuzzy c-means), desenvolvido por Bezdek [3] para obter os valores pretendidos de  $u_{ik}$ ,  $i = 1 \dots s$ ,  $k = 1 \dots n$  e de  $s_i$ ,  $i = 1 \dots s$ . Os valores de  $u_{ik}$  são normalmente agrupados numa matriz  $U$  ( $s \times n$ ), designada por matriz de partição imprecisa.

Dada a especificidade desta utilização das técnicas de análise de agrupamentos imprecisos, a escolha do número de agrupamentos tem menor importância do que no caso geral; como se compreenderá no desenvolvimento do método, o impacto de qualquer possível "má escolha" (nesse sentido) é completamente atenuada pelo carácter progressivo das decisões tomadas ao longo do processo, e pela possibilidade de regulação da velocidade de progressão através do parâmetro  $u_0$ . Poder-se-á, portanto, utilizar um valor de  $s$  igual ao limite de discriminação  $n_{AD}$  do Agente de Decisão, ou um valor definido por este em cada estágio do processo.

Apresenta-se em seguida o procedimento base:

### Método de Ampliações Sucessivas

- (i) Obter uma partição imprecisa  $U$  do conjunto  $Z$ , por meio de um algoritmo de agrupamento impreciso. Apresentar os protótipos  $s_i$  ( $i=1..s$ ) ao AD, como representação das macro-soluções;

- (ii) **Decisão:** O AD selecciona as macro-soluções mais prometedoras. Seja  $P$  o conjunto dos índices dos agrupamentos seleccionados;

- (iii) Construir o conjunto reduzido:

$$Z^R = \{z_k \mid z_k \in Z \text{ e } \sum_{i \in P} u_{ik} \geq u_0\}$$

onde  $u_0$  é um limiar de aceitação pré-especificado;

- (iv) Se o número de elementos de  $Z^R$  é maior do que o pretendido, fazer  $Z=Z^R$  e regressar a (i);

No caso contrário,  $Z^R$  é o conjunto final a apresentar ao Agente de Decisão;

- (v) **Decisão:** O AD selecciona a sua decisão preferida.

São de realçar algumas características deste método, no que respeita ao Agente de Decisão:

- (a) É proporcionada uma perspectiva global dos conjuntos, ao longo do processo;
- (b) O Agente de Decisão nunca tem de considerar, em nenhuma fase do procedimento, mais alternativas do que pretende;
- (c) Com excepção da decisão final, é sempre possível tomar decisões múltiplas;
- (d) Todas as decisões são escolhas entre alternativas: O AD nunca tem de fornecer, ou tomar decisões, sobre pesos, valores de compensação, ou níveis de aspiração;
- (e) O método não é prescritivo, mas impede a deambulação, devido à progressiva redução do conjunto de alternativas activas.

A apresentação dos protótipos dos agrupamentos deve ser feita da maneira mais adequada ao AD. Entre as várias possibilidades deve salientar-se a apresentação dos protótipos sob a forma de histograma, com referência aos valores máximos e mínimos dos atributos, de forma a manter uma perspectiva global, mesmo nas decisões sobre conjuntos reduzidos (suconjuntos do conjunto inicial). Também é favorável a possibilidade de reconsiderar (*backtraking*), voltando alguns passos atrás e recomeçando. Qualquer destes aspectos está contemplado na aplicação que se descreve na secção seguinte.

### 3 - Uma aplicação do MAS

Uma versão fundamental do MAS foi codificada em MicroSoft C, utilizando o *software* gráfico ATEGA desenvolvido no INESC [1], para utilização em ambiente MS-DOS. O programa dispõe de uma interface gráfica que permite, interactivamente e de um modo amigável, a escolha de uma solução, entre um vasto conjunto de alternativas, de acordo com o procedimento descrito na secção anterior.

O programa tem capacidade para problemas com 200 alternativas, 10 atributos e 8 agrupamentos, não tendo havido preocupação em alargar estes limites, nem de minimizar tempos de execução, devido à natureza demonstrativa da aplicação. Referem-se em seguida os comandos existentes, disponíveis sob a forma de botões no menu principal, cujo aspecto pode ser visto na secção 4.

#### 3.1. Criação e gestão de ficheiros de dados

Directório: Permite obter uma listagem dos casos disponíveis em disco.

Criar: Este comando permite criar, de forma interactiva, um ficheiro com os dados relativos a um determinado problema. O utilizador pode, depois de criar um ficheiro, alterar os dados, visualizá-los e gravar o problema em disco.

Gravar: Este comando permite gravar em disco os dados do problema corrente.

Carregar: Permite transferir do disco para memória os dados relativos a um dado problema, previamente gravado.

Ver: Esta opção permite listar todas as alternativas, com identificação (por cor) das alternativas rejeitadas e activas num dado passo do processo de decisão.

Alterar: Este comando permite alterar os dados relativos ao caso corrente, incluindo a adição de atributos ou alternativas.

### 3.2. Operação do método

MAS: Ao ser accionado este comando, serão pedidos ao utilizador o número máximo de agrupamentos que está disposto a considerar e o limiar de aceitação  $u_0$ . O programa accionará, então, o seu algoritmo interno de agrupamento com vista a calcular os protótipos, considerando as alternativas activas nesse momento do processo. No final, serão apresentados ao utilizador os agrupamentos calculados (indicando-se os centros dos *clusters*).

Histograma: Para ajuda à selecção das macro-soluções preferidas, são apresentados ao utilizador histogramas correspondentes aos protótipos calculados pelo processo de agrupamento. Os histogramas estão sempre referidos ao máximo e mínimo disponíveis em cada atributo, de forma a manter permanentemente o utilizador situado no conjunto global das alternativas. Em face da informação fornecida, o utilizador deve marcar a(s) macro-soluções(s) que lhe pareça(m) mais promissor(as). Por razões de espaço no monitor, apenas são apresentadas, de cada vez, um máximo de quatro soluções, mas o utilizador pode fazer com facilidade as combinações que entender.

Decisão: Este comando permite ao Agente de Decisão tomar a decisão final, entre as alternativas seleccionadas nesse momento.

Backtracking: Esta opção permite ao utilizador reconsiderar, anulando algumas decisões parciais e regressando ao ponto do processo que entender, para continuar a partir daí.

Alternativas: Lista as alternativas activas, permitindo a visualização do histograma correspondente a qualquer alternativa seleccionada (máximo: 20).

### 3.3. Outros comandos

Informação: Este comando permitirá, futuramente, apresentar informações úteis ao utilizador.

Sair: Termina a execução do programa.

#### 4 - Exemplo ilustrativo

Nesta secção ilustra-se a utilização do programa, em relação a um problema de escolha de uma máquina de lavar, adaptado de Zeleny [8], simulando as decisões parciais de um hipotético Agente de Decisão, de forma tanto quanto possível coerente, mas sem recorrer a qualquer tipo de função de valor implícita.

O Agente de Decisão deve escolher entre 33 alternativas, caracterizadas pelos seguintes atributos: (1) Preço da máquina, (2) Tempo total de lavagem, (3) Consumo de energia e (4) Consumo de água. Com excepção do preço, todos os atributos estão referidos ao programa mais frequentemente usado. Em relação à minimização destes atributos, nenhuma alternativa é dominada.

Máquina	Preço (US\$)	Tempo de Lavagem (min)	Consumo Energia (kWh)	Consumo água (litros)
1	509	74	1.5	114
2	425	80	1.5	110
3	446	72	1.6	135
4	564	65	1.6	118
5	547	53	1.8	140
6	450	68	1.6	135
7	473	65	1.6	130
8	484	56	1.7	115
9	456	68	1.6	130
10	488	72	1.6	114
11	530	55	1.7	135
12	477	76	1.5	110
13	589	53	1.6	130
14	534	61	1.4	122
15	536	57	1.7	110
16	494	71	1.5	135
17	425	65	1.8	120
18	555	53	1.7	125
19	543	57	1.6	120
20	515	68	1.5	130
21	452	76	1.5	112
22	547	68	1.5	120
23	421	76	1.4	130
24	498	68	1.6	120
25	467	65	1.7	130
26	595	50	1.8	135
27	414	68	1.7	125
28	431	66	1.7	110
29	452	72	1.5	115
30	408	77	1.6	119
31	478	59	1.8	110
32	395	76	1.5	120
33	543	57	1.5	135

Os dados referentes a este problema encontram-se num ficheiro de dados (LAVAR) criado anteriormente. O utilizador deverá utilizar o comando CARREGAR para transferir os dados para memória (Fig 1).

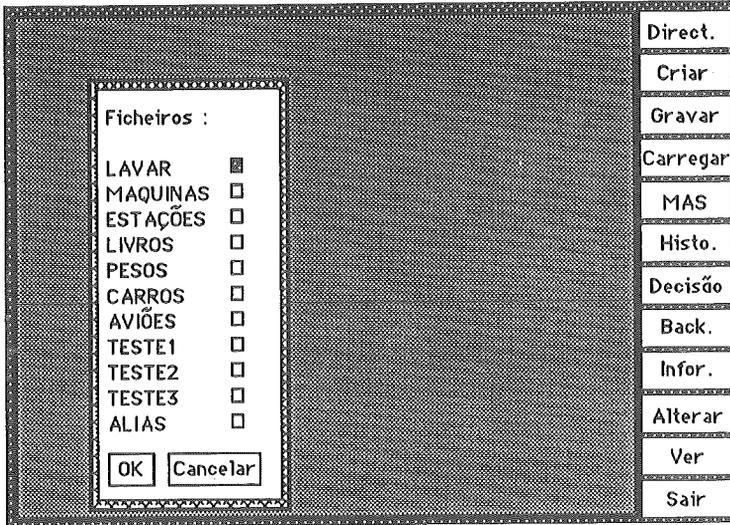


Fig 1. Comando CARREGAR.

Ficará então activo o ficheiro LAVAR. O utilizador poderá listar as opções (comando VER) ou, se o desejar, acrescentar alternativas ou atributos (comando ALTERAR). O utilizador dará então início ao processo, premindo a opção MAS.

O Agente de Decisão indicará ao programa o número máximo de alternativas que está disposto a considerar (4, neste exemplo) e o limiar de aceitação  $u_0$ , que parametriza as sucessivas reduções do conjunto.

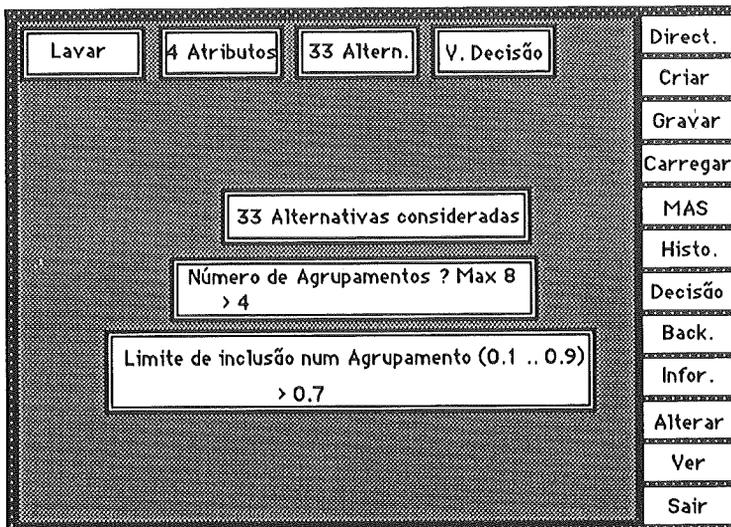


Fig 2. Comando MAS

O programa realiza uma partição imprecisa do conjunto inicial (33 alternativas) apresentando os protótipos ao AD (ver tabela).

Atributo	macro-soluções			
	1	2	3	4
Preço	492.9	553	417	456.4
Tempo lavagem	67.91	57.83	71.73	70.68
Consumo energia	1.588	1.627	1.6	1.585
Consumo água	118.7	126.4	119.2	126.5

O AD deverá prosseguir, utilizando o comando HISTOGRAMA que lhe permite visualizar os histogramas dos protótipos calculados pelo processo de agrupamento. O máximo e o mínimo de cada atributo são utilizados como limites (esta característica mantém-se durante todo o processo, mesmo se as soluções correspondentes ao máximo ou ao mínimo forem eliminadas). Neste momento, o utilizador deverá marcar o(s) agrupamento(s) que lhe pareça(m) mais promissor(es). No nosso exemplo o Agente de Decisão marcou os agrupamentos 1 e 4.

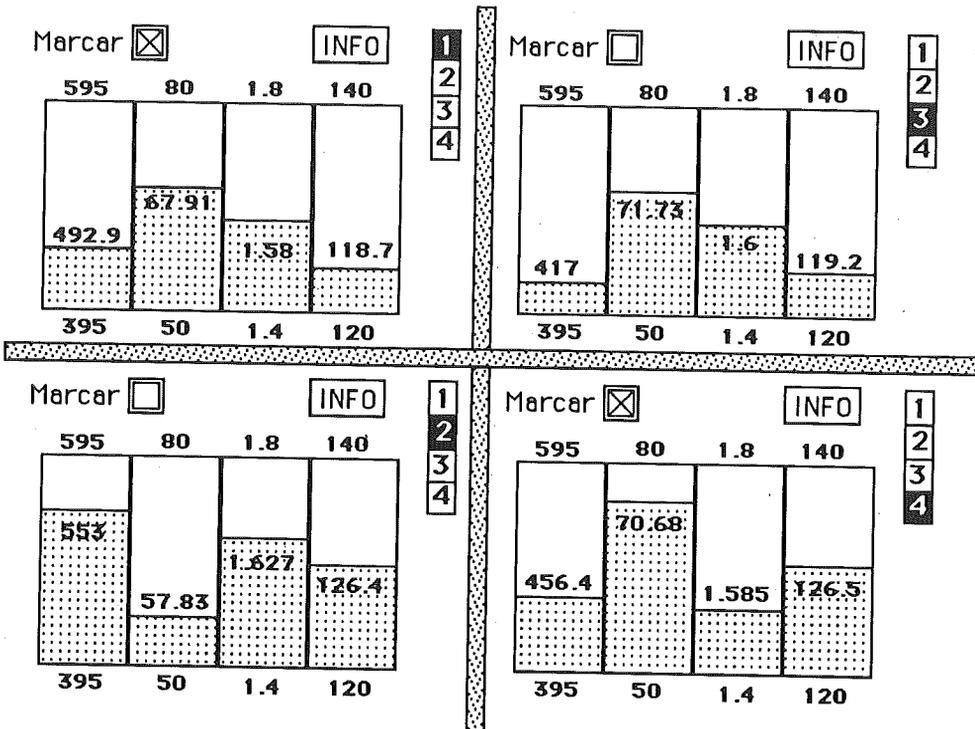


Fig 3. Comando Histograma

Neste momento, o Agente de Decisão pretendeu continuar o processo de escolha. Para isso, prime o comando MAS. O programa indica-lhe então que estão activas 15 alternativas que o

AD decide agrupar em 5 agrupamentos, indicando um limiar de aceitação de 0.8. A tabela seguinte apresenta o resultado:

Atributo	macro-soluções				
	1	2	3	4	5
Preço	481.7	496	451.2	469.9	512
Tempo lavagem	65.73	69.91	72.81	65.02	71
Consumo energia	1.65	1.55	1.56	1.65	1.5
Consumo água	112.2	127.5	125.4	130.	122

Neste ponto o utilizador pode regressar a um ponto anterior da execução (*backtracking*), tomar a decisão final ou continuar o processo, utilizando o comando HISTOGRAMA. No nosso exemplo, o AD optou pela terceira hipótese, marcando os agrupamentos 3 e 4, do que resultou a redução do conjunto para 7 alternativas.

O AD decide então prosseguir com o método (comando MAS) fornecendo ao programa indicações para agrupar as alternativas activas em 3 agrupamentos, com limiar de aceitação 0.8.

Resultados obtidos :

Atributo	macro-soluções		
	1	2	3
Preço	450.7	452	470
Tempo lavagem	72.01	74	65
Consumo energia	1.6	1.5	1.65
Consumo água	133.3	113.5	130

Utilizando o comando HISTOGRAMA, o AD marcou o agrupamento 3, ao qual correspondem as alternativas 7 e 25. Neste momento, o AD, sentindo-se capacitado para tomar uma decisão entre estas duas alternativas, premiu o comando DECISÃO e seleccionou a alternativa 25.

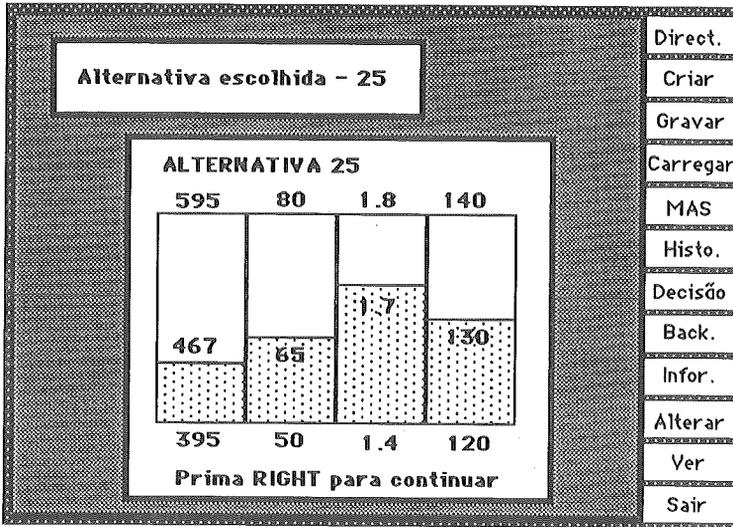


Fig 4. Comando DECISÃO.

Neste momento, o AD pode não estar completamente satisfeito com a sua opção. Eventualmente, poderá ter perdido alguma alternativa melhor durante o processo de escolha. Decide então fazer BACKTRACKING, voltando ao ponto da execução onde estavam activas 15 alternativas. A análise poderá prosseguir normalmente a partir daqui.

5 - Conclusões

O método descrito na comunicação permite ao Agente de Decisão progredir num conjunto com um grande número de alternativas através de reduções do conjunto inicial, amplificando sucessivamente a zona correspondente às suas preferências. Todas as decisões são tomadas sobre alternativas (reais ou virtuais), com a ajuda de uma interface amigável baseada em janelas, menus, botões e rato, e que inclui uma permanente referência global às hipóteses extremas disponíveis em cada atributo. O método não é prescritivo, mas impede a deambulação, embora permita ao AD reconsiderar, retomando o processo num ponto anterior.

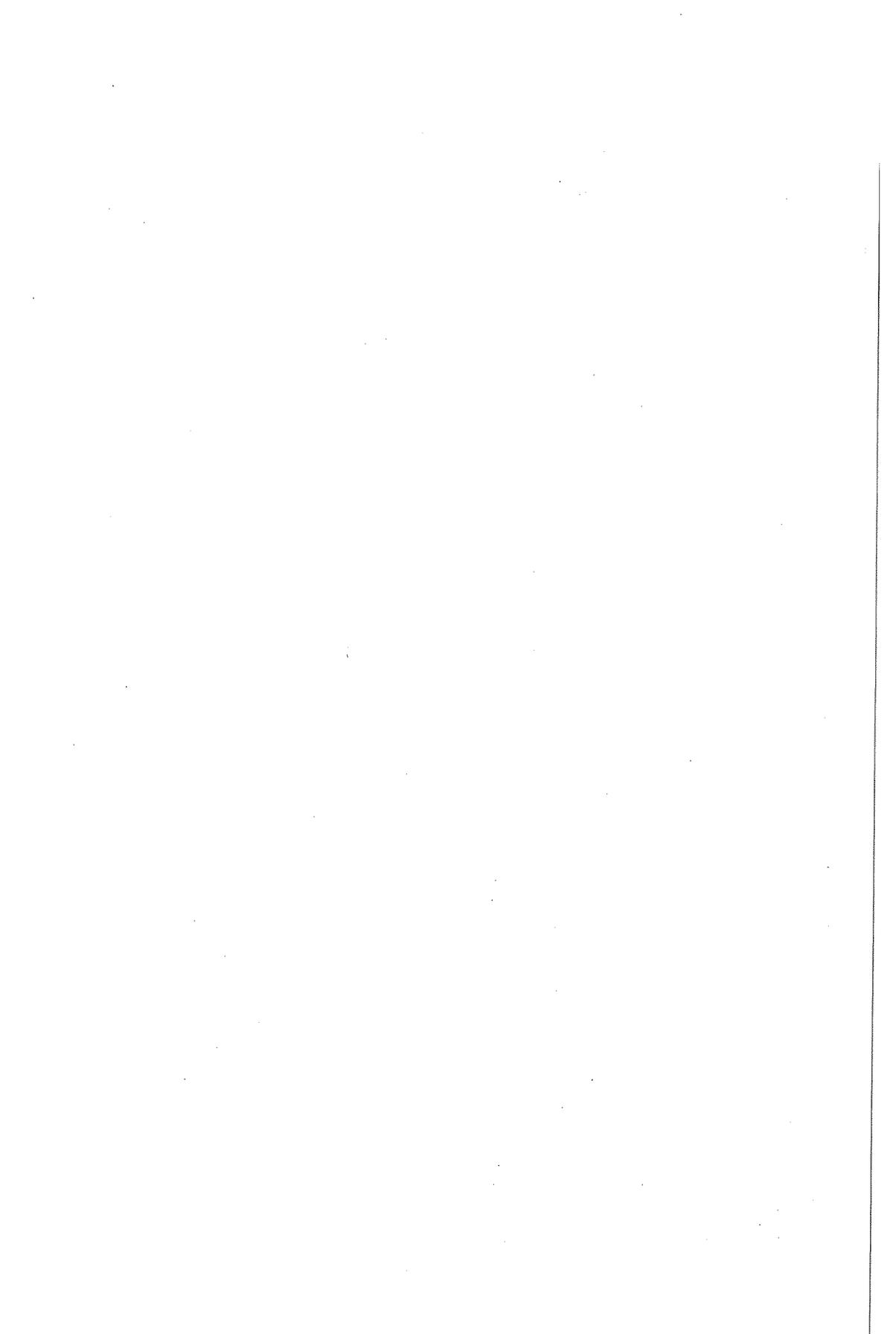
O Agente de Decisão não é nunca confrontado com mais alternativas do que deseja, nem tem de fornecer ou tomar decisões sobre pesos, valores de compensação ou níveis de aspiração, sendo-lhe ainda possível tomar decisões parciais múltiplas. A metodologia pretende, em geral, apoiar o AD na selecção de um conjunto final de dimensão aceitável por este, a partir de um conjunto inicial de maior dimensão. A fase final da decisão pode ser realizada por apreciação directa sobre o conjunto final, ou recorrendo a métodos prescritivos.

## 6 - Agradecimentos

Os autores estão gratos ao Eng<sup>o</sup> João Paulo Germano, bolseiro BIC no INESC, que programou a rotina de agrupamento impreciso (no âmbito do projecto PMCT/C/TIT/413/90, parcialmente subsidiado pela JNICT), e ao aluno da FEUP, Paulo Marques, pela elaboração de algumas rotinas gráficas.

## 7 - Bibliografia

- [1] Alves, J.C., Manual de utilização do Atega, INESC, Porto, 1988.
- [2] Benbasat, I., Dexter, A.S., "An Experimental Evaluation of Graphical and Color-Enhanced Information Presentation", Management Science 31 nº 11, 1985.
- [3] Bezdek, J. C., Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
- [4] Matos, M.A., Ajuda à Decisão Multicritério - novas contribuições, dissertação de doutoramento, FEUP, Porto, 1988.
- [5] Matos, M.A., "Uma Metodologia para Melhorar a Comunicação na Ajuda à Decisão Multicritério", 4º Congresso da APDIO - IO89, Lisboa, 1989.
- [6] Miller, G.A., "The magical number seven, plus or minus two: some limits on our capacity for processing information", Psychological Rev. 63 nº 2, 1956.
- [7] Proença, L.M., Sistema Gráfico para o Método das Ampliações Sucessivas, relatório do trabalho realizado no âmbito da disciplina de Metodologias de Ajuda à Decisão, do Mestrado em Engenharia Electrotécnica e de Computadores da FEUP, 1992.
- [8] Zeleny, M., Multiple Criteria Decision Making, McGraw-Hill, New York, 1982.



# LOCATION OF EXTENSIVE FACILITIES: POTENTIAL APPLICATIONS

**Juan A. Mesa**

Depto. de Matemática Aplicada  
Universidad de Sevilla,  
Sevilla, Espanha

## **Abstract**

While problems involving location of facilities, which can be represented as points, have been sufficiently studied in many cases, most of the corresponding ones for the location of facilities such as highways or communication subnetworks have only been considered during last ten years and open problems often appear.

This paper is devoted to the second type of location problem. After the introduction of the already known theoretical and computational results, their possible applications are reviewed, and future directions of research are suggested.

## **Keywords**

Location, extensive facilities, survey.

## **1 - Introduction**

Most of the papers in network location theory are concerned with the selection of places for locating facilities, which can be represented as isolated points in a network, but during the last decade some papers about location of dimensional structures as paths or trees, have been published. Thus location theory has been extended in a way that is close to some classical problems of the operations research: network design, routing, shortest path.

In order to try to establish differences between these problems and the ones of interest in this paper, by location of structure we understand the selection of such structure in the network according to one or several criteria that take into account the interaction between the object to locate and some points ("demand/antidemand points") of the network. Therefore, functions either in the objective or in the restrictions, are dependent on these interactions.

However, the criteria usually applied for determining a path or a tree for those classical problems, only take into account inherent magnitudes of the structure to being considered, as for instance costs or travelling times.

Often practical problems assume the construction of the selected facility, but in other cases problems only involve the selection of an already existent subnetwork, as for instance for the problem of routing vehicles containing hazardous waste from source to disposal place.

Some of the criteria used in point location have been applied to the location of paths and sometimes of trees in a network, and different results and algorithms have been obtained, but the literature about these problems, which is still incomplete, is quite scarce; partially because of the different and possible heading used. In the present paper we are trying to bring together this

research in order to classify potential applications. For this reason paragraph 2 is devoted to a summary description of the different models already studied.

Thus the point of view that follows is from the theory of the models already known, though applications are sometimes sources of mathematical problems, and future directions of research from the applications are also suggested. The final purpose of this paper is to provide in a structured way, the potential applications of this group of location models and problems.

## 2 – Models and criteria

Several models have already been researched and they can be classified according to the criterion/criteria used in the objective function and the restrictions imposed.

Each location criterion can be applied to the location of central or antientral facilities. For a central facility we understand that which is desired to be "close to" demand Points, whereas an antientral facility is one which is desired to be far from the demand (or more properly antidemand) point. For instance a route for transporting hazardous materials must be far from the populated areas. Depending on the central or antientral character the objective function must be minimized or maximized.

Problems of location of structures can be restricted or not restricted, but location of antientral structures gives rise to restricted problems. Restrictions often used are the two extreme points for a path, the total length, the cardinality of the vertex set of the structure or some maximum distance or travel time between demand points and structure.

If  $G$  is a finite non directed graph, for which  $V(G)$  is the vertex set and  $A(G)$  is the arc set, we suppose that for each  $a = (v_i, v_j) \in A(G)$ , the length  $l(a) = d(v_i, v_j)$  is known. This allows us to define the total length of a structure with tips at vertices as the sum of the lengths of all edges of the structure. Should a structure have tips or end points at points different than vertices, an embedding of the graph in an Euclidean space it is assumed to be possible.

Let  $d(u, v)$  be the distance (the minimum distance through all the paths connecting  $u$  and  $v$ ) between two vertices  $u$  and  $v$  of  $G$ , and for structure  $S$  and a vertex  $v \in V(G)$  let  $d(v; S) = \min_{u \in V(S)} d(v, u)$  be the distance between  $v$  and  $S$  ( $V(S)$  is the set of vertices of  $S$ ).

One of the most used decision criterion is the eccentricity which consists of minimizing (maximizing):

$$e(S) = \max_{v \in V(G)} d(v, S)$$

between all the structures  $S$  included in some class of structures  $S$  (paths, trees, forest, etc.). In particular if  $S$  is the class  $P$  of all the paths of  $G$  the structures of  $P$  which minimizes cardinality of the vertex set of minimum eccentricity is called *path center* (Hedetniemi et al. [14], Slater [24]), and linear time algorithms for finding it in tree graphs, have been provided.

While the above criterion takes into account the worst situation of the demand points, the optimization of the total sum of the distances from the vertices to the structure is a measure of efficiency, and those paths that minimize:

$$S(P) = \sum_{v \in V(G)} d(v, P)$$

among all the paths  $P$  of the graph are called core or path median (Morgan and Slater [22]), in which a linear time algorithm when the underlying structure is a tree graph, is devised. This definition can also be applied for types of structures other than paths.

A related criterion is the branch weight, which for a tree  $T$  consists of the selection of the paths  $P$  which minimize the maximum of the total weights of the components of  $T - P$  (Slater, [24]). The path which minimizes cardinality of the vertex set of minimum branch weight is called spine or path centroid.

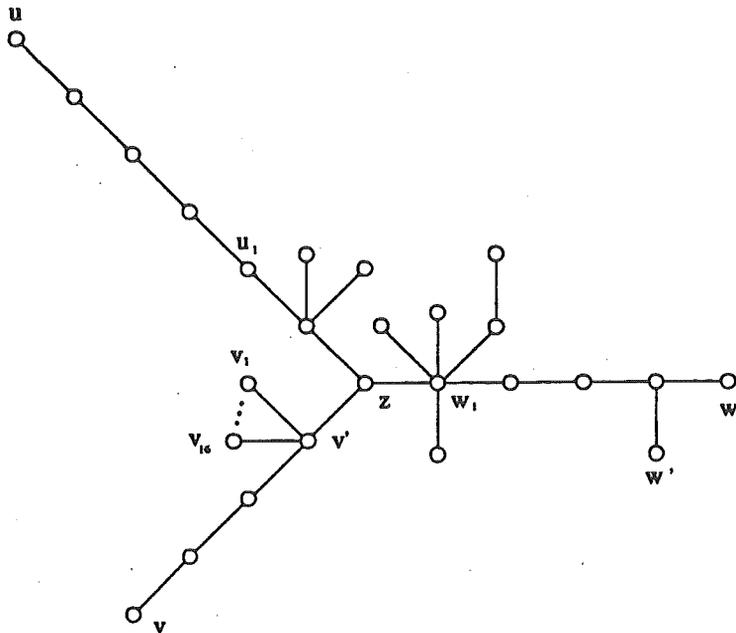
When a trade-off between efficiency and effectiveness is desired, the  $\alpha$ -cent-dian criterion should be used [19]. This consists in optimizing the convex combination:

$$F(\alpha) = \alpha e(P) + (1 - \alpha) S(P)$$

for each  $\alpha \in [0, 1]$ .

For the tree graph of the figure in which the length of each arc is the unit, the absolute center  $c$  is the midpoint of the arc  $(u_1, z)$ , the median and the centroid set are  $\{z, v'\}$ , while the path center  $P_c = P(u_1, w_1)$  connect the vertices  $u_1, w_1$ , there are two path medians  $P_m = P(v, w)$ ,  $P'_m = P(v, w')$ , the path centroid is  $P_{cd} = P(v', w_1)$  and the path  $\alpha$ -cent-dians are

$$P_\alpha = \begin{cases} P_m \text{ and } P'_m & \text{if } \alpha \in [0, \frac{1}{2}], \\ P(u, v) & \text{if } \alpha \in [\frac{1}{2}, \frac{2}{3}], \\ P(u, w) & \text{if } \alpha \in [\frac{2}{3}, 1]. \end{cases}$$



However, in practice problems do not often consist only of the optimization of an objective function, they are in fact restricted. Restrictions on the total length of the structure have been considered in (Minieka [21]) and on the cardinality of the vertex set in (Kincaid et al. [18]). In this paper covering type problems are considered, that is to say to minimize either

$$|V(S)| \text{ or } l(S)$$

such that  $f_j(d(v_j, S)) \leq r$ , in which  $r$  represents a given threshold and  $f_j$  are continuous increasing functions. A good survey of the computational complexity issues for the different length restricted problems is given in (Halimi et al. [13]).

The restriction to the origin and the destination points rise to a group of path location problems, that have important applications, for instance, to vehicle routing (Berman and Handler [2]) or transportation of hazardous materials (Batta and Chiu [1], Boffey and Karkazis [4]).

The combination of the requirement of the shortest path with some locational problems, such as, for instance, the median path (Current et al. [8]) or the covering one (Current et al. [5], [7]), leads to multiobjective problems which seem realistic models of many practical problems.

A different way of combining several criteria is to include one term for each criterion in the objective function, as for instance in the paper by Kim et al. [17], in which the objective function is to minimize the sum of the setup cost of establishing the facility and the cost of providing service which is given by the median function.

### **3 - Possible applications of the models**

In this paragraph, and in order to assist practitioners in decision-making, some potential applications are grouped in different subjects, though boundaries are sometimes hazy. However, the following points do not to be a complete list of applications but only some of the most important ones.

#### **3.1. Design of transportation networks**

The introduction of space into the economic theory was made in the last century, in which several papers were devoted to the study of the best tracing of railroad lines. However, these were forgotten until approximately the middle of the present century. In fact, it is since the early 70's, that problems of optimally locating transportation network have been systematically researched. According to their nature, facilities in location theory can be classified into two groups: the existent ones, which represent the demand points, and the new facilities which will provide a service. The former can be concentrated in a finite number of points or can be distributed continuously in some subset of a Euclidean space. At this point, it is convenient to mention the distinction between continuous location models and the discrete ones. If the new facilities may be located anywhere in a homogeneous transportation area, the problem is referred to as a continuous one, and a good reference for these type models is the Mosler's

book [23]. However, if they must be in a basis graph, the problem is called discrete, with which this paper is concerned. Advantages of the continuous are that they provide more general results, informations on the geometry of the structure for locating, on the shape of lines (arcs) and on the junctions (nodes) of the optimal structure. However, some discrete models are good synthesis of the real world and there are often efficient algorithms for them. A survey of network design and transportation planning is provided in the paper of Magnanti and Wong [20].

The location criteria which can be used for locating a segment (path) or a basic system (tree or subgraph) of highways, pipelines or railroad lines between two fixed points or connecting some of them, are the minimization of the distance sum or the total travelling time from the structure to the demand points (towns or cities), when targets of efficiency are prevalent, and the minimization of the eccentricity when it is convenient to take into account the worst situation of the demand points.

However, sometimes problems in practice have a multiobjective nature, and costs, travelling times or total length of the structure are primary goals. Thus the combination of the criteria usually applied for the generically called shortest path problem, as for instance the total cost of constructing the structure, with the minimization of the eccentricity or the distance sum or with covering objective, is a better way to modelize these problems.

Thus the median "shortest" path problem consists of minimizing the functions:

$$\sum_{a \in P} c(a)$$

$$\sum_{v_i \in V(G)} f(v_i) d(v_i, P)$$

in which  $c(a)$  is the cost of constructing the arc  $a$  of  $P$  and  $f(v_i)$  is the demand of the vertex  $v_i$ , among all the paths  $P \in P(O,D)$  connecting the origin vertex  $O$  with the destination  $D$ .

This problem can be formulated as a two objective integer program and an algorithm to generate noninferior solutions is provided in the paper of Current et al. [8].

The combination of the shortest path objective, with restrictions of covering, leads to the shortest covering path problems [5]:

$$\min_P \sum_{a \in P} c(a)$$

s.t.  $d(v, P) \leq r$  for each  $v \in V(G)$

Assuming that a vertex is covered when it is within a given distance  $r$  from the path to locate, the restriction of covering could be substituted for the maximization of the demand covered [6]. For urban mass transit systems, more realistic models are obtained, when the

minimization of the total travelling time that costumers spend from their origins to their destinations, is included as an objective.

A related problem is the shortest center path, which consists of:

$$\min_P \sum_{a \in P} c(a)$$

$$\min_P \max_{v \in V(G)} \sum_{a \in P} d(v, P)$$

and combination of the shortest path problem with the  $\alpha$ -cent-dian path,  $\alpha \in [0, 1]$ , can be interesting because of its parametric nature, as a way to weight efficiency and effectiveness.

When demand points define the nodes of the basis graph, the upgrading of road systems, the central pipelines network for irrigation or for gas, or a subway system are further potential applications, for which the corresponding models for subtrees can be used.

### 3.2. Routing in transportation networks

Routing problems have been researched from different points of view, but the ones of interest here are those in which some locational criterion is taken into account.

Some of the problems described in the relevant literature are: routing for newspaper distribution, in which nodes on the path to be covered by trucks are road intersections from which newspapers are picked up to be transported by carriers or vans. This problem was cited as an instance of the shortest covering path problem by Current et al. [5]; determination of routes for express lanes (path medians), mass transit systems, airline and shipping routes, optimal paths for mobile service to a "nonemergency" destination (Berman and Rahnama [3]) and to a "nonservice" destination (Berman and Handler [2]) are further potential applications.

Other potential application such as routing of health care teams in developing countries or some instances of the travelling salesman problem, can be modelized as median or covering shortest tour problem [9], [10].

### 3.3. Routing of undesirable facilities

Optimal transportation of hazardous materials is a problem with increasing interest in developed countries, so that distinguished attention is paid in present research. An example of this, is the special issue on transportation of hazardous materials of *Transportation Science*: vol.25, n°2, 1991. Analogously to the point-location of undesirable facilities, the problem in mind has an "antical" character, that is to say to locate a facility to be furthest away from the demand points (in this case antidemand points or populated areas). In order to get this antical character, maximization of the eccentricity or the distance sum function can be considered, but surely the use of the covering models are more realistic approaches (Current et al. [7]). The impact on the populated areas may be aggregated as weights; in the closest part of the path to be located using a minisum criterion (Batta and Chiu [1]), the minimization of the

expected damage effects (Boffey and Karkazis [4]), the finding of a shortest path subject to risk equity restrictions (Gopalan et al. [11]) and the selection of an equitable set of routes (Gopalan et al. [12]), have been also considered.

Because of the multiobjective nature of routing of hazardous materials, there are combinations of criteria that lead to real problems but they have not yet been researched. Also, the p-dispersion path problem remains open which is to be set of p paths furthest away each other.

### 3.4. Other problems

Applications of the tree covering problems:

$$\begin{aligned} \min_T \quad & l(T), \\ \text{s.t.} \quad & d(v_i, T) \leq r_i \quad \text{for all } i \in M \end{aligned}$$

that is to say to find the tree T of minimum length such that the point  $v_i$  must be at most  $r_i$  units of distance from T, to the determination of a snow emergency zone in a city (Kim et al. [15], [17]), as well as to the positioning of troops along a front, when the troops must protect demand points to be covered (Kim et al [16]).

Location of a main material handling system for a flexible manufacturing system, can be modeled by the problem of minimizing the eccentricity among all the paths, which cardinality of the vertex set or the total length would be less than or equal to a fixed L (Kincaid et al. [18]), and repavement of a contiguous segment of highway can be regarded as the location of a tree-shaped facility.

### References

- [1] Batta, R. and Chiu, S.S., Optimal Obnoxious Paths on a Network, *Oper.Res.* 36 (1988) 84-92.
- [2] Berman, O. and Handler, G.Y., Optimal Minimax Path of a Single Service Unit on a Network to Nonservice Destination, *Trans.Sci.* 21 (1987) 115-122.
- [3] Berman, O. and Rahnama, M.R., Optimal Path of a Single Service Unit on a Network to an "Nonemergency" Destination, *Trans.Sci.* 17 (1983) 218-232.
- [4] Boffey, T.B. and Karkazis, J., Routing of Hazardous Materials, Proceedings Meeting V of the Euro Working Group on Locational Analysis, Namur (1990) 1-7.
- [5] Current, J., ReVelle, C. and Cohon, J., The Shortest Covering Path Problem: an Application of Locational Constraints to Network Design, *Journal of Regional Science* 24 (1984) 161-183.
- [6] Current, J., ReVelle, C. and Cohon, J., The Maximum Covering Shortest Path Problem: A Multiobjective Network Design and Routing Formulation, *EJOR* 21 (1985) 189-197.
- [7] Current, J., ReVelle, C. and Cohon, J., The Minimum Covering/Shortest Path Problem, *Decision Theory* 19 (1988) 490-503.
- [8] Current, J., ReVelle, C. and Cohon, J., The Median Shortest Path Problem: a Multiobjective Approach in the Design of Transportation Networks, *Trans.Res.*21 (1987) 188-197.
- [9] Current, J. and Schilling, D., The Covering Salesman Problem, *Trans.Sci.* 23 (1989) 208-213 .
- [10] Current, J. and Schilling, D., The Median Tour and Maximal Covering Tour Problems: Formulations and Heuristics, to appear in *EJOR* (1992).
- [11] Gopalan, R., Batta, R. and Karwan, M.H., The Equity Constrained Shortest Path Problem, *Computers Opns.Res.* 17 (1990 a) 297-307.
- [12] Gopalan, R., Kolluri, K.S., Batta, R. and Karwan, M.H., Modelling Equity of Risk in the Transportation of Hazardous Materials, *Opns.Res.* 38 (1990 b) 961-973.

- [13] Hakimi, S.L., Schmeichel, E.F. and Labbé, M., On Locating Path-or Tree-shaped Facilities on Networks, Report 9055/A Econometric Institute, Erasmus University (1990).
- [14] Hedetniemi, S.M., Cockaine, E.J. and Hedetniemi, S.T., Linear Algorithms for Finding the Jordan-Centre and Path Centre of a Tree, *Trans.Sci.* 15 (1981) 98-114.
- [15] Kim, T.U., Lowe, T.J., Ward, J.E. and Francis, R.L., A Minimum Length Covering subgraph of a Network, *Annals of Operations Research* 18 (1989) 245-260.
- [16] Kim, T.U., Lowe, T.J., Ward, J.E. and Francis, R.L., A Minimum Length Covering Subtree of a Tree, *Naval Research Logistics* 37 (1990) 309-326.
- [17] Kim, T.U., Lowe, T.J. and Ward, J.E., Locating a Median Subtree on a Network, *INFOR* 29 (1991) 153-166.
- [18] Kincaid, R.K., Lowe, T.J. and Morin, T.L., The Location of Central Structures in Trees, *Comput.Opns.Res.* 15 (1988) 103-113.
- [19] López de los Mozos, M.C. and Mesa, J.A., Location of Cent-dians Paths in Tree Graphs, *Proceedings of the VI Meeting of the Euro Working Group on Location Analysis (Tenerife, Spain), 1992.*
- [20] Magnanti, T.L. and Wong, R.T., *Network Design and Transportation Planning: Models and Algorithms*, *Trans.Sci.* 18 (1984) 1-55.
- [21] Minieka, E., The Optimal Location of a Path or Tree in a Tree Network, *Networks* 15 (1985) 309-321.
- [22] Morgan, C.A. and Slater, P.J., A Linear Algorithm for a Core of a Tree, *Journal of Algorithms* 1 (1980) 247-258.
- [23] Mosler, K.C., *Continuous Location of Transportation Networks*, Springer-Verlag, 1987.
- [24] Slater, P.J., Locating Central Paths in a Graph, *Trans.Sci.* 16 (1982) 1-18.

# LINEAR AND LAGRANGEAN PENALTIES FOR VARIABLE BOUNDING IN ILP: AN APPLICATION TO A COVERING PROBLEM

**Margarida Vaz Pato**  
Departamento de Matemática  
Instituto Superior de Economia e Gestão  
Rua Miguel Lupi, 20  
1200 Lisboa - Portugal

## Abstract

Lagrangian and linear penalties can be used for variable bounding in ILP. Such penalties, embedded in a branch-and-bound algorithm, yield remarkable reductions in the search procedure effort for large scale problems.

In this paper, four different ways of exploring this idea for a covering problem with integer variables are presented. Computing results taken from test problems have revealed the efficiency of the technique in reducing the amplitude of variable intervals, and even in fixing them at feasible values.

## Resumo

Num problema de PLI com variáveis limitadas os intervalos de variação das variáveis podem ser "apertados" tomando em consideração penalidades lagrangeanas ou lineares, as quais vêm assim facilitar a enumeração implícita em PLI, especialmente se os problemas forem de grande dimensão.

Para um problema de cobertura com variáveis inteiras limitadas apresentam-se quatro alternativas de aplicação desta ideia, bem como os resultados computacionais obtidos num conjunto de problemas-teste. A experiência evidenciou a eficácia da técnica na fixação de variáveis e na redução, percentualmente elevada, da amplitude dos intervalos.

## Keywords

integer linear programming, covering problems, lagrangian relaxation, penalties.

## 1 - Introduction

The purpose of this paper is to show how lagrangian and linear relaxations can be used for variable bounding in integer linear problems with bounded variables (ILPs). Two general results, stated in Section 2, validate the updated variable bounds for all the feasible solutions better than a stipulated one. Such features have been called lagrangian or linear penalties though they do not lead to penalties in the strict sense of the word. This nomenclature arises from the penalization incurred when the solution goes outside the updated variable interval.

In Section 3, we shall point to different ways of defining lagrangian penalties for a covering problem with integer variables, referred to as the generalized set covering problem (Paixão and Pato (1988)). Finally, some computational results will be given to illustrate that

penalties introduce remarkable reductions in the solution space, at least in the covering problems tested so far.

Therefore, as this technique is very inexpensive computationally, it is most suitable for accelerating the bounding procedure in large scale problems.

Mention should be made of the following works, where lagrangean or linear penalties were applied to several problems: Sweeney and Murphy (1981) – the multi-item scheduling problem; Balas and Ho (1980), Hey (1980), Paixão (1983), Beasley (1987) – the set covering problem; Chan, Bean and Yano (1987) – the set partitioning problem; Hall and Hochbaum (1985) – the multi-covering problem. It may be worth noting that, in these studies, the updating of the variable bounds does no more than fix the variables at 0 or 1, because the problems are binary. In the case of the ILP, Nemhauser and Wolsey (1988) present a linear penalty result which is weaker than the one stated here. These authors apply the penalties to a binary problem – the travelling salesman problem. In fact, the possibilities for bounding are much more relevant in binary problems.

As is known similar penalties have already been derived from the linear relaxation by Beale and Small (1965), Driebeek (1966) and Tomlin (1971). However such post-linear penalties are extremely time-consuming as they require the optimal linear tableau.

## 2 – Linear and Lagrangean Penalties for ILP

Let us take the following integer linear problem with bounded variables:

$$\begin{array}{ll}
 \text{(ILP)} & \text{minimize} & \mathbf{c} \mathbf{x} \\
 & \text{subject to} & \mathbf{A} \mathbf{x} \geq \mathbf{b} \\
 & & \mathbf{0} \leq \mathbf{x} \leq \mathbf{h} \\
 & & \mathbf{x} \text{ integer}
 \end{array}$$

where  $\mathbf{c}$  is a real vector,  $\mathbf{b}$ ,  $\mathbf{h}$  are integer vectors,  $\mathbf{A}$  is an integer matrix, and all have appropriate dimensions.

As we know, the linear relaxation of the integer linear problem is the ILP without the integrality constraints, and is denoted by LinILP.

Now let  $\bar{z}$  be an upper bound on the optimal value of ILP,  $z_{\text{ILP}}$ , and  $[\mathbf{u} \ \mathbf{v}]$  a feasible solution for the dual linear problem. As usual,  $\mathbf{r}$  represents the reduced costs associated with  $[\mathbf{u} \ \mathbf{v}]$  and  $\underline{z}$  the dual objective value corresponding to  $[\mathbf{u} \ \mathbf{v}]$ .

Following these definitions, consider the general result for linear penalization which can be directly taken from duality:

### Proposition 1

For a particular  $k$ ,

- if  $r_k > 0$  then  $x_k \leq \lfloor (\bar{z} - \underline{z})/r_k \rfloor$   
in any feasible solution better than  $\bar{z}$ .  
( $\lfloor a \rfloor$  – the largest integer strictly less than  $a$ )

Now, rather than rejecting the integrality constraints, some of the main constraints are rejected by embedding them in the objective function associated with non-negative multipliers,  $\lambda \geq 0$ . Thus, this lagrangean relaxation of ILP may be presented as follows:

$$\begin{array}{ll}
 \text{(Lag}_\lambda\text{ILP)} & \text{minimize} & \mathbf{c} \mathbf{x} - \lambda (\mathbf{A}^2 \mathbf{x} - \mathbf{b}^2) \\
 & \text{subject to} & \mathbf{A}^1 \mathbf{x} \geq \mathbf{b}^1 \\
 & & \mathbf{0} \leq \mathbf{x} \leq \mathbf{h} \\
 & & \mathbf{x} \text{ integer}
 \end{array}$$

where the vectors and matrices have appropriate dimensions.

The decomposition of the constraint set into two blocks,  $\mathbf{A}^1 \mathbf{x} \geq \mathbf{b}^1$  and  $\mathbf{A}^2 \mathbf{x} \geq \mathbf{b}^2$ , is performed in such a way that Lag $_\lambda$ ILP possesses the integrality property (Geoffrion (1974)). In an extreme case, constraints  $\mathbf{A}^1 \mathbf{x} \geq \mathbf{b}^1$  may not exist. Thus, all constraints, with the exception of the bounding ones, are relaxed.

Once more, let  $\bar{z}$  stand for an upper bound on  $z_{\text{ILP}}$ . Let Lag $_\lambda$ ILP represent the optimal value for Lag $_\lambda$ ILP. Here vector  $\hat{\mathbf{a}}$  stands for the reduced lagrangean costs defined through the expression

$$\hat{a}_j = c_j - \lambda \mathbf{A}^{2j} - \mathbf{u}^* \mathbf{A}^{1j}$$

where  $\lambda$  is the vector of lagrangean multipliers and  $\mathbf{u}^*$  is the vector of optimal dual linear variables corresponding to the constraints  $\mathbf{A}^1 \mathbf{x} \geq \mathbf{b}^1$  in Lag $_\lambda$ ILP.

One may, at this point, derive the second result for lagrangean penalization:

**Proposition 2**

For a particular  $k$ ,

• if  $\hat{a}_k > 0$  then  $x_k \leq \lfloor (\bar{z} - z_{\text{Lag}_\lambda\text{ILP}}) / \hat{a}_k \rfloor$  (case 1)

• if  $\hat{a}_k < 0$  then  $x_k \geq \lceil h_k + (\bar{z} - z_{\text{Lag}_\lambda\text{ILP}}) / \hat{a}_k \rceil$  (case 2)

in any feasible solution better than  $\bar{z}$ .

( $\lceil a \rceil$  - the smallest integer strictly greater than  $a$ )

**Proof**

The main ideas involved are as follows.

In the first case, where  $\hat{a}_k > 0$ , we include the following additional constraint in ILP

$$x_k \geq (\bar{z} - z_{\text{Lag}_\lambda\text{ILP}}) / \hat{a}_k$$

which produces the enlarged problem, ILP<sup>E</sup>.

Firstly, the constraints  $\mathbf{A}^2 \mathbf{x} \geq \mathbf{b}^2$  are relaxed and associated with any vector of non-negative multipliers  $\lambda \geq 0$ . Then we find problem Lag $_\lambda$ ILP<sup>E</sup>.

The other constraints,  $\mathbf{A}^1 \mathbf{x} \geq \mathbf{b}^1$ , are now relaxed by taking  $\mathbf{u}^*$  as multipliers, which leads to:

$$\begin{array}{ll}
 \text{(Lag}_{\mathbf{u}^*}\text{Lag}_\lambda\text{ILP}^E) & \text{minimize} & \mathbf{c} \mathbf{x} - \lambda (\mathbf{A}^2 \mathbf{x} - \mathbf{b}^2) - \mathbf{u}^* (\mathbf{A}^1 \mathbf{x} - \mathbf{b}^1) \\
 & \text{subject to} & 0 \leq x_j \leq h_j \text{ and integer} \quad (j \neq k) \\
 & & (\bar{z} - z_{\text{Lag}_\lambda\text{ILP}}) / \hat{a}_k \leq x_k \leq h_k.
 \end{array}$$

The optimal value for the enlarged doubly-relaxed problem verifies:

$$z_{\text{Lag}_u^* \text{Lag}_\lambda \text{ILPE}} = z_{\text{Lag}_u^* \text{Lag}_\lambda \text{ILP}} + \hat{u}_k (\bar{z} - z_{\text{Lag}_\lambda \text{ILP}}) / \hat{u}_k =$$

and thus, from the integrality of the involved problems,  $\text{Lag}_u^* \text{Lag}_\lambda \text{ILP}$  and  $\text{Lag}_\lambda \text{ILP}$ , and the specific selection of multipliers for the last relaxation,  $u^*$ , the following equality arises:

$$= z_{\text{Lag}_\lambda \text{ILP}} - \bar{z} - z_{\text{Lag}_\lambda \text{ILP}} = \bar{z}.$$

Therefore, by applying the definition of lagrangean relaxation we have

$$z_{\text{ILPE}} \geq \bar{z}$$

that is, the additional constraint must be violated in a solution (to the ILP) whose value is greater than  $\bar{z}$ . And from the integrality of this variable,  $x_k$ , one may immediately conclude the validity of the updated variable bound presented in this proposition (case 1).

For (case 2) the proof is similar. ◆

Returning to the two propositions stated above, we can see that penalties provide some kind of valid inequalities for the feasible solutions better than  $\bar{z}$ .

As has already been seen, these results simply require an upper bound on the optimal value of the ILP, together with the by-products of a lagrangean relaxation (Proposition 2), or a feasible dual linear solution (Proposition 1). They may therefore be easily computed.

For the above reasons such penalties may be used for variable bounding purposes in the case of large scale ILPs.

### 3 – Linear and Lagrangean Penalties for the Generalized Set Covering Problem

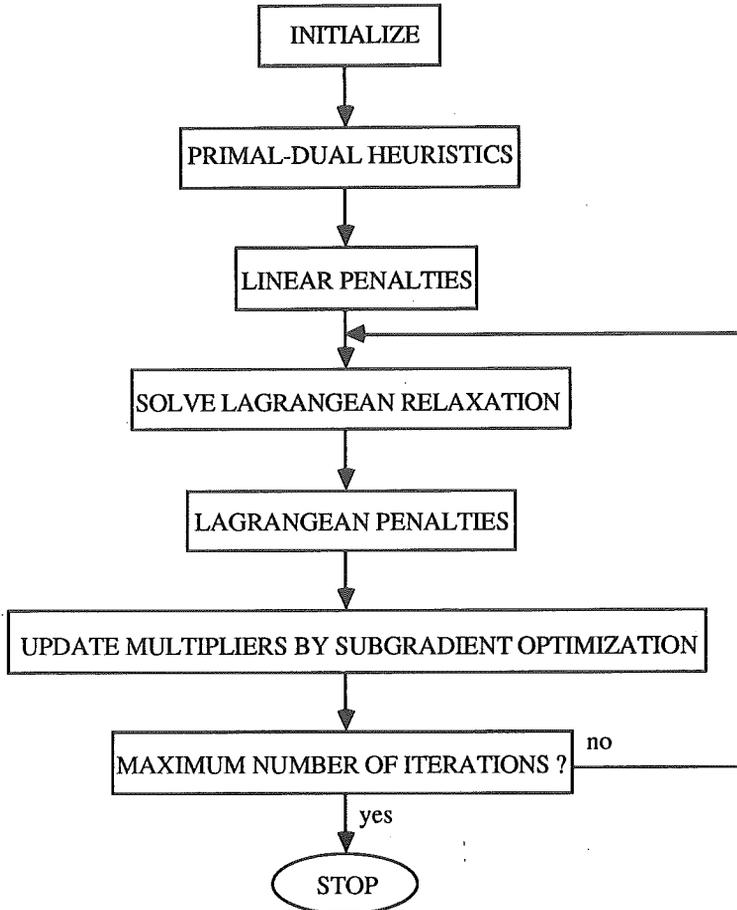
Let us now take a particular ILP which is a covering problem, the so-called generalized set covering problem, in short GSCP. It is expressed in the same way as the ILP, where matrix  $A$  is a binary one.

Several relaxations were developed and tested on the GSCP (Paixão and Pato (1988)).

One of them is the linear relaxation. Another corresponds to the relaxation of all covering constraints in a lagrangean fashion. The corresponding relaxed problem is trivially solved by inspection.

Two different lagrangean relaxations were also specially designed for the scheduling applications of the GSCP, taking into account the structure of such instances. One of them, the structural relaxation, includes a reformulation of the GSCP with redundant constraints. It then relaxes some of the original constraints. The other one, the decomposition relaxation, comes from a standard use of variable splitting. Both relaxed problems can be efficiently solved by a minimum cost network flow code.

Appropriate penalties were deduced from each of the four relaxations mentioned. These penalties can be easily calculated through a heuristic-lagrangean procedure such as the following:



**Figure 1** – Heuristic-Lagrangean Procedure

This heuristic-lagrangean algorithm with penalties (Figure 1) was devised to bound the optimal value of the GSCP. In our case, it was used not only in the root node of a branch-and-bound, but also – again for bounding purposes – in every node of the branching procedure. It should be noted that in this algorithm the calculation was truncated when a maximum number of generated nodes was reached. Such a combined procedure was tested in a set of medium-sized GSCP instances. Some of them were taken from literature whereas others were not. They were either derived through random generation, according to different patterns, or from scheduling real applications.

Structural relaxation was the most favourable of the three lagrangean relaxations in this experiment.

Computing results for the above truncated branch-and-bound using structural relaxation are summarized in the following table.

Table 1 – Summary of Typical Cases (Combined Procedure)

probl. (1)	dimensions density (2)	type references (3)	effect a penalties relative behaviour		
			node zero fixed (4)	reduction (5)	(6)
RN18	36 × 865 56%	scheduling applications Pato (1989)	587	86%	BETTER UPPER BETTER CPU
RN21	36 × 865 56%	idem	731	51%	BETTER UPPER BETTER CPU OPTIMAL SOLUTION
RN7	46 × 1500 45%	idem	103	80%	BETTER UPPER BETTER LOWER BETTER CPU
RNPA	41 × 378 52%	idem	157	69%	BETTER LOWER BETTER CPU OPTIMAL SOLUTION
CRF1	47 × 333 39%	idem	18	83%	BETTER LOWER BETTER CPU
RN12	36 × 865 56%	idem	580	94%	BETTER CPU
HAH2	50 × 200 5%	random generation Hall and Hochbaum (1985)	144	10%	WORSE UPPER BETTER CPU
MOR1	168 × 168 24%	scheduling applications Morris and Showalter (1983)	0	0%	—

This table sets out to demonstrate the effects of linear plus structural lagrangean penalties by showing only one of each typical case found throughout the experiment. This small set of problems (see column (1)) thus illustrates all the relevant situations found so far.

Column (2) indicates the dimensions and density of the problems, whereas column (3) indicates the type and source of each.

Column (4) represents the number of variables for which the upper and lower bounds were fixed at equal values during the process. And column (5) gives the percentual reduction on

interval width over non-fixed variables. All information given in columns (4) and (5) was obtained at the end of node zero.

As may be seen, the penalties were efficient in reducing the size of the solution space. This resulted from the fixing of variables or the updating of their bounds. In fact, in some problems the number of fixed variables was high. Moreover, in all cases with the exception of the last two, the percentual reduction was significant. These exceptions were due to the poor quality of the upper bound solutions produced by the heuristics.

The last column (6) shows the results at the end of the procedure. Not all the computational results obtained through the tests are to be found in the table. We preferred to emphasize the relative behaviour of this algorithm with penalties as compared to a version without penalties.

Let us now stress the results for the algorithm (with penalties):

- the final gap for the optimum was slightly better in most cases;
- the CPU time was lower in almost all problems and, in some cases, very much lower;
- on a number of occasions the algorithm optimally solved the problem before reaching the stipulated maximum number of nodes – the version without penalties did not;
- in rare situations one found that the upper bound was worse.

One should add that, in many other problems, quality and time proved to be the same for both versions.

In short, due to the high number of fixed variables and the significant reduction in variable intervals, the performance of the combined procedure was improved by penalties – not to the extent that one would have expected – but at no computer expense. One may add that its behaviour proved to be better in the practical cases (the algorithm was, in fact, developed with them in mind).

#### **4 – Conclusions**

Penalties from linear and lagrangean relaxations of ILPs are easily calculated and computationally cheap. In addition, they can be effective in reducing the size of the problem by fixing variables at particular values and shortening the width of variable intervals.

Literature presents the favourable application of these penalties for binary linear problems and, as has been seen, the above experiment applies such penalties to medium-scale covering problems with integer variables. Therefore, it is felt that similar results would be obtained from other IPLs with bounded variables, whenever the resolution option includes relaxation procedures and the problem dimensions are considerable.

## 5 - Referencies

- [1] Balas, E. and Ho, A. (1980), "Set Covering Algorithms Using Cutting Planes, Heuristics and Subgradient Optimization: A Computational Study", *Mathematical Programming* 12, 37-60.
- [2] Beale, E.M.L. and Small, R.E. (1965), "Mixed Integer Programming by a Branch and Bound Technique", *Proceedings of the Third IFIP Congress 1965*, vol.2, 450-451.
- [3] Beasley, J.E (1987), "An Algorithm for Set Covering Problem", *European Journal of Operational Research* 31, 85-93.
- [4] Chan, T.J, Bean, J.C. and Yano, C.A. (1987), "A Multiplier-Adjustment-Based Branch-and-Bound Algorithm for the Set Partitioning Problem", *Technical Report 87-OR-O8*, Department of Operational Research and Engineering, Management School of Engineering and Applied Science, Dallas.
- [5] Driebeek, N.J. (1966), "An Algorithm for the Solution of Mixed Integer Programming Problems", *Management Science* 12, 576-587.
- [6] Geoffrion, A.M. (1974), "Lagrangean Relaxation for Integer Programming", *Mathematical Programming Study* 2, 82-114.
- [7] Hall, N.G. and Hochbaum, D.S. (1985), "The Multicovering Problem: The Use of Heuristics, Cutting Planes and Subgradient Optimization for a Class of Integer Programs", *Working Paper WPS 85-73*, College of Administrative Science, The Ohio State University, Columbus.
- [8] Hey, A. (1980), "Algorithms for the Set Covering Problem", Ph.D. Thesis, Imperial College, London.
- [9] Morris, J.G. and Showalter, M.J. (1983), "Simple Approaches to Shift Days-Off and Tour Scheduling Problems", *Management Science* 29, 924-950.
- [10] Nemhauser, G.L. and Wolsey, L.A. (1988), "Integer and Combinatorial Optimization", John Wiley & Sons, New York.
- [11] Paixão, J.P. (1983), "Algorithms for Large Scale Set Covering Problems", Ph.D.Thesis, Imperial College, London.
- [12] Paixão, J.P. and Pato, M.V. (1988), "A Structural Lagrangean Relaxation for the Two-Duty Period Bus Driver Scheduling Problem", *European Journal of Operational Research* 39, 213-222.
- [13] Pato, M.V. (1989), "Algoritmos para o Problema de Cobertura Generalizado", Ph.D.Thesis, Faculdade de Ciências, Lisboa.
- [14] Sweeney, J.D. and Murphy, R.A.(1981), "Branch and Bound Methods for Multi-Item Scheduling", *Operations Research* 29, 853-864.
- [15] Tomlin, J.A. (1971), "An Improved Branch-and-Bound Method for Integer Programming", *Operations Research* 19, 1070-1075.

## PRÉMIOS APDIO-SCHNEIDER 92

Os prémios APDIO-SCHNEIDER 92 foram instituídos com o objectivo de galardoar trabalhos de Investigação Operacional (contribuições teóricas, aplicações ou software) desenvolvidos recentemente por alunos de licenciaturas e de cursos de mestrado de universidades portuguesas.

Incluem-se neste número da Revista os trabalhos seleccionados para publicação, no âmbito dos referidos prémios:

- PRÉMIO PARA OS TRABALHOS DE LICENCIATURA

**"Um Sistema Informático para o Apoio ao Sequenciamento de Tarefas", por Fernando Antunes Vieira (Licenciatura em Engenharia Electrotécnica e Computadores, Faculdade de Engenharia da Universidade do Porto)**

- PRÉMIO PARA OS TRABALHOS DE MESTRADO

**"Escalonamento de Pessoal Tripulante: Utilização de uma Técnica de Relaxação de Espaço de Estados para Problemas de Cobertura", por Ana da Silva Paias (Mestrado em Estatística e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa)**

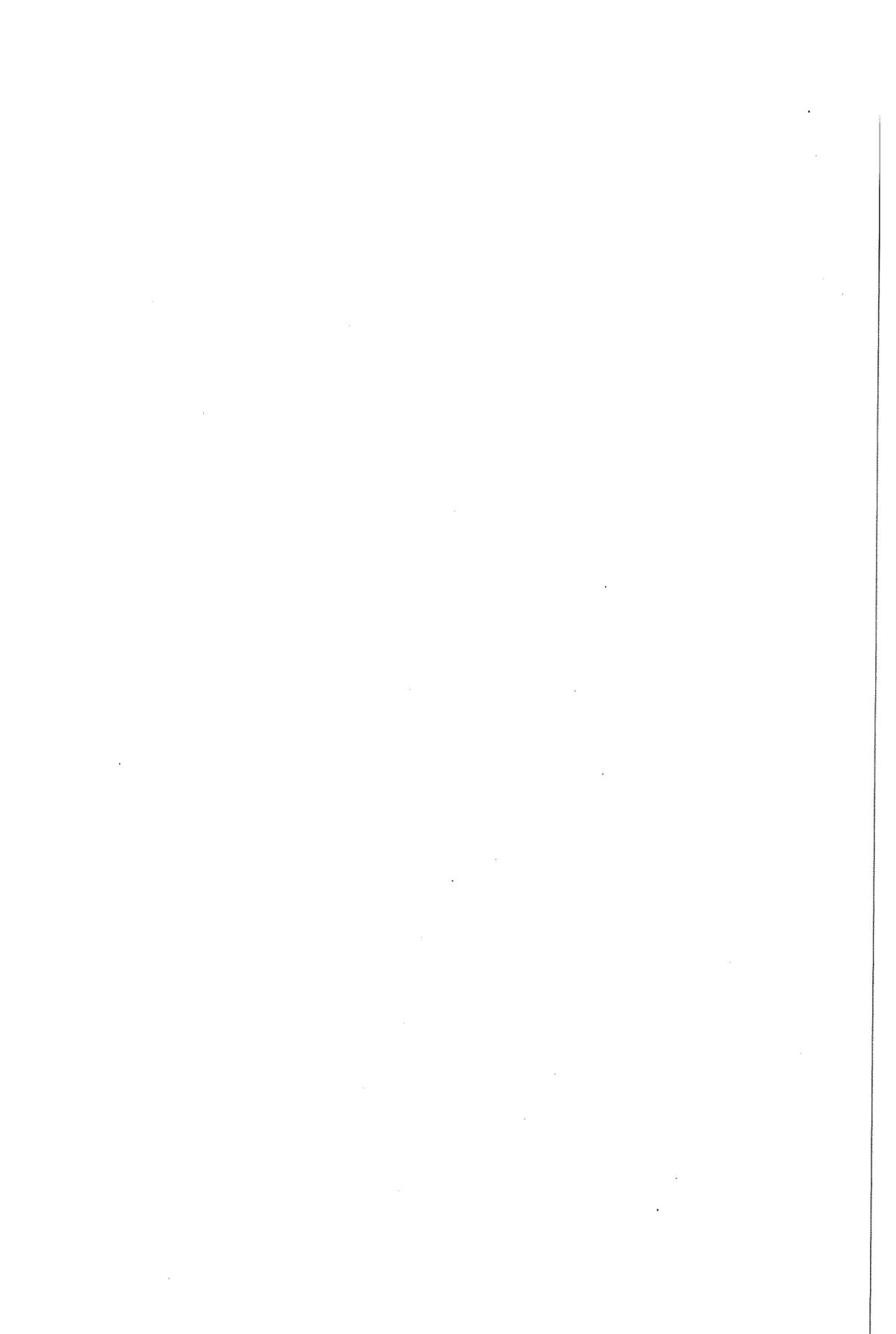
- MENÇÃO HONROSA PARA OS TRABALHOS DE MESTRADO

**"Diferentes Abordagens do Problema de Localização", por Maria da Conceição Fonseca (Mestrado em Estatística e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa).**

Perante a elevada qualidade dos trabalhos que agora se publicam, é devida aos seus autores uma manifestação de grande apreço e uma palavra de encorajamento para que prossigam na via que tão bem souberam iniciar.

O significativo número de candidaturas aos prémios APDIO-SCHNEIDER, oriundas de diferentes cursos das mais prestigiadas universidades portuguesas, é sintoma da ampla difusão da Investigação Operacional no ensino superior português, que nos apraz registar. Adicionalmente, é revelador do interesse que os Prémios suscitaram, que não pode ser dissociado do patrocínio concedido pela IFS-Informática, Formação e Serviços, SA. A esta empresa apresentamos os nossos agradecimentos.

Rui Campos Guimarães  
Presidente da APDIO



# UM SISTEMA INFORMÁTICO PARA APOIO AO SEQUENCIAMENTO DE TAREFAS

**Fernando Vieira**

INESC Porto - Instituto de Engenharia de Sistemas e Computadores  
Largo Mompilher, 22 - 4000 Porto

## **Abstract**

Production planning often implies scheduling a set of tasks, involving very complex product routings. Nevertheless, it is often possible to identify critical points for which a careful planning leads to clear gains in productivity. In this paper we describe a real situation of this type, and identify a particular scheduling problem where there are "time windows" for the tasks, and "setup times" exist. This problem was modelled as a Travelling Salesman Problem with additional constraints.

Given the complexity and the size of the practical situation under study, a local search algorithm was developed. After constructing an initial feasible schedule, this algorithm iteratively improves that solution, by an exchange procedure specially designed for this case.

Finally, illustrative computational results are presented, and a practically useful Decision Support System, which integrates the algorithm, is described.

## **Resumo**

O planeamento da produção passa frequentemente pelo escalonamento de um conjunto de tarefas, envolvendo seqüências complexas de operação. Apesar disso, é frequentemente possível identificar pontos críticos para os quais um planeamento cuidadoso pode resultar em ganhos significativos de produtividade. Nesta comunicação, descreve-se sucintamente um caso real em que tal acontece, e através de uma análise hierárquica, caracteriza-se um problema particular de planeamento que consiste em sequenciar, numa máquina, um conjunto de tarefas com janelas temporais e tempos de "setup" ou de preparação. Este problema foi modelizado como um problema de caixeiro viajante com restrições adicionais.

Reconhecida a complexidade e dimensão da situação prática em consideração, desenvolveu-se um algoritmo de pesquisa local que, após a construção de um escalonamento inicial admissível, procede a um melhoramento da solução inicial, através de uma permutação criteriosa das tarefas.

São apresentados resultados computacionais que ilustram o desempenho da abordagem adoptada, descrevendo-se sucintamente um Sistema de Apoio à Decisão em que foi integrado o algoritmo desenvolvido, e cuja utilização parece, na prática, bastante promissora.

## **Keywords**

Scheduling, Traveling Salesman Problem

## 1 - Introdução

Na generalidade das situações industriais, o planeamento da produção passa pelo escalonamento de um conjunto de tarefas, envolvendo o recurso a vários postos de trabalho e seqüências complexas de operação. Apesar disso, é frequentemente possível identificar pontos críticos para os quais um planeamento cuidadoso pode resultar em ganhos significativos de produtividade. Neste artigo, descreve-se sucintamente um caso real em que tal acontece, e através de um processo de decomposição hierárquico, caracteriza-se um problema particular de planeamento.

A análise destes pontos críticos é geralmente entregue a pessoas que, através de uma longa experiência, foram desenvolvendo procedimentos considerados de utilidade prática. No entanto, por vezes, a complexidade dos problemas práticos é tal que o responsável pela sua resolução tem apenas como preocupação respeitar as restrições impostas, não podendo garantir a qualidade das soluções obtidas. Por outro lado, a empresa corre o risco de que essas pessoas se tornem indispensáveis ao seu funcionamento, com todas as implicações daí resultantes.

Actualmente, grande parte dos problemas com que os responsáveis pela produção se deparam são resolvidos de uma forma pouco satisfatória, manualmente ou utilizando métodos automáticos simples (por exemplo, uma folha de cálculo). Em particular, o problema real no qual este trabalho foi inspirado, e que se passa a descrever, é dificilmente resolvido, pelo menos de uma maneira satisfatória, recorrendo a métodos tradicionais.

Trata-se de uma empresa de mobiliário metálico em que a quase totalidade dos lotes de fabrico tem de passar por um último centro de trabalho, onde é realizado um conjunto de operações de pintura. As diversas operações (tratamento, aplicação de primário, pintura final e secagem automática) são efectuadas numa única máquina, de grandes dimensões, e cujos custos de funcionamento são bastante elevados.

As encomendas ao sector de produção podem ser externas (feitas por clientes exteriores), logo sujeitas a prazos de entrega mais ou menos apertados, ou internas (produção para stock) pelo que têm um maior grau de liberdade no que respeita ao cumprimento de prazos.

Existe um programa de produção diário que permite prever o momento de chegada à pintura, dos lotes provenientes dos centros anteriores (soldadura, por exemplo).

A mudança do tipo de produto pode exigir um trabalho de preparação prévia na máquina de pintura ("setup"). Por exemplo, a mudança na cor de pintura exige que a máquina seja limpa antes de se iniciar a pintura na nova cor. O tempo dedicado ao "setup" pode ser bastante significativo.

Numa primeira fase, os produtos a pintar foram agrupados em famílias. A escolha da família depende, por exemplo, da cor final do produto, das suas dimensões, do tipo de pistolas

de pintura necessárias, etc. Tornou-se, assim, possível, criar uma matriz de tempos de "setup" entre as várias famílias de produto.

O sequenciamento dos lotes na máquina de pintura é actualmente feito de uma forma manual, por alguém com bastante experiência, e que procura exclusivamente minimizar o número de "setups" efectuados.

Verifica-se na prática que os lotes, ao chegar à secção de pintura, se vão acumulando junto à máquina, à espera de vez para serem pintados. Por esta razão, os atrasos na pintura originam, por vezes, falta de espaço, a qual se pode tornar extremamente crítica.

O desconhecimento dos prazos de entrega, por parte do responsável pela pintura, leva a que o departamento de vendas exija, frequentemente, a entrada em pintura dos lotes em atraso. Estas exigências fazem com que saia frustrada qualquer tentativa de minimizar o número de "setups".

Por outro lado, as constantes alterações no programa (por exemplo a chegada de uma encomenda considerada mais urgente) levam à necessidade de se refazer várias vezes a sequência de entrada dos lotes na máquina. Dado que este trabalho tem de ser feito rapidamente, e na falta de um apoio informático adequado, as soluções que se conseguem obter são em geral bastante insatisfatórias.

Conclui-se, finalmente, que a máquina de pintura representa um verdadeiro "gargalo" de todo o sistema. Pelo que os atrasos aqui originados resultam em atrasos em todo o fluxo produtivo. A principal consequência deste facto é, obviamente, uma diminuição significativa na produtividade global da fábrica.

## 2 - Descrição do Problema

A situação descrita levou a caracterizar um problema genérico em que o objectivo é sequenciar um conjunto de tarefas numa única máquina, de modo a minimizar o tempo de finalização da última tarefa e cumprir rigorosamente as restrições impostas.

Para as diferentes tarefas, considera-se disponível a seguinte informação:

- (i) à tarefa  $i$  é associado um tempo de processamento  $t_i$  e cada tarefa está sujeita a uma janela temporal, ou seja, a tarefa  $i$  só está disponível para ser processada na data  $e_i$  e tem uma data limite para estar concluída  $l_i$ ;
- (ii) existe um tempo de "setup" (preparação) entre cada duas tarefas que depende, simultaneamente, da tarefa anterior e da actual (fazer um "setup" significa levar a máquina para um certo estado e, em geral, o custo associado depende do estado de partida).

São ainda consideradas as seguintes restrições gerais:

- (i) a máquina só pode executar uma tarefa de cada vez;
- (ii) cada tarefa, depois de iniciada, é processada até ao fim (não podendo ser interrompida ou cancelada);
- (iii) as tarefas são agrupadas por famílias (grupos de tarefas com as mesmas exigências de preparação), originando assim uma matriz de tempos de "setup" entre famílias que não é necessariamente simétrica;
- (iv) não se considera o tempo de "setup" necessário ao início da primeira tarefa.

### 3 - Modelo

Para o problema descrito, foi considerado como modelo de base o Problema do Caixeiro Viajante ("Traveling Salesman Problem"), com restrições adicionais respeitantes às janelas temporais ("Traveling Salesman Problem with Time Windows" ou TSPTW).

No TSP "puro", pretende-se encontrar o percurso (rota) de menor distância que visita cada uma de  $N$  cidades, começando e terminando na cidade 1, e passando uma só vez em cada cidade.

Uma janela temporal é o intervalo de tempo dentro do qual a cidade tem de ser visitada. Designar-se-á por " $e_i$ " (de "earliest") a data de serviço mais cedo à cidade  $i$ , ou seja, a cidade não pode ser servida antes desse instante. Por isso, caso se chegue à cidade  $i$  antes de  $e_i$  dever-se-á esperar. Chamar-se-á " $l_i$ " (de "latest") ao tempo de serviço mais tarde da cidade, pelo que é obrigatório visitar a cidade  $i$  antes de  $l_i$ .

A janela temporal associada à cidade  $i$  fica, desta forma, definida pelo intervalo  $[e_i, l_i]$ .

Fica assim estabelecido o paralelismo entre este modelo e o problema do Sequenciamento de Tarefas com Janelas Temporais e Tempos de Setup (que se designará por WINSET de "Time Windows and Setups"). Esse paralelismo torna-se claro através das correspondências apresentadas na tabela seguinte:

<u>TSPTW</u>	<u>WINSET</u>
Cidade	- Tarefa
Tempo de serviço	- Tempo de processamento
Dist. entre cidades	- Tempo de "setup" entre tarefas
$[e_i, l_i]$ (janela da cidade)	- $[e_i, l_i]$ (janela da tarefa)
min.(tempo de percurso)	- min.(fim da última tarefa)
min. $(\sum d_{ij})$	- min. $(\sum \text{tempos de "setup"})$
Todas as cidades visitadas	- Todas as tarefas cumpridas
Chegada a uma cidade	- Máquina disponível para próxima tarefa
Início de serviço de um cliente	- Início de processamento de uma tarefa

Partida de uma cidade	- Fim de process. de uma tarefa
No percurso entre duas cidades	- Máquina em "setup"
Espera por cliente	- Espera por tarefa (máquina vazia)

#### 4 - Abordagem utilizada

Para além da reconhecida complexidade do problema, verifica-se que os dados iniciais do problema, em casos reais, podem ser muito pouco fiáveis. Por exemplo, se a data prevista para uma tarefa estar disponível não se verificar, o escalonamento que possivelmente foi estabelecido poderá não ter sentido, pelo que os cálculos terão de ser totalmente refeitos.

Torna-se, assim, evidente, que a principal característica de um sistema automático para a resolução deste problema terá de ser a sua eficiência (rapidez na obtenção de soluções), em detrimento da sua eficácia (qualidade das soluções obtidas), que permita uma interacção fácil com o utilizador.

Posta de lado a hipótese de se tentar obter soluções óptimas, optou-se pelo recurso a um algoritmo de pesquisa local, que determina uma solução inicial e, calculando o valor do objectivo para um conjunto de soluções vizinhas, tenta melhorar a solução anteriormente obtida. Corre-se, assim, o risco de se atingir um mínimo local (para o qual nenhum vizinho é "melhor"), embora se observe na prática que as soluções obtidas são, em geral, bastante satisfatórias.

#### 5 - Algoritmo

##### 5.1. Introdução

A abordagem adoptada para a resolução do problema é constituída, essencialmente, por duas etapas:

- 1 - Construir um escalonamento inicial admissível. Usam-se, para o efeito, heurísticas para a inserção das tarefas na lista que define o seu sequenciamento;
- 2 - O escalonamento obtido no ponto anterior, apesar de admissível, pode ser pouco satisfatório no que respeita ao objectivo pretendido. Neste caso, procurar-se-á melhorar esta solução inicial, através da permutação de tarefas. A melhor solução obtida, dentro do conjunto de permutações que resultarem em soluções admissíveis, será escolhida como solução final (a estratégia de permutações será discutida adiante).

Relativamente ao modelo adoptado (TSP com janelas temporais), a primeira etapa do algoritmo consiste na construção de uma rota que seja admissível (isto é, respeite todas as janelas temporais) através de heurísticas de inserção. O uso destas heurísticas justifica-se, pois prova-se que encontrar uma rota admissível é, só por si, um problema complexo (NP-difícil).

Na busca da solução inicial, as heurísticas consideradas incluem, simultaneamente, os aspectos temporais e espaciais do problema. Numa primeira etapa, procede-se da seguinte forma:

- (i) Para cada cidade actualmente não incluída, calcular o melhor lugar de inserção na rota em construção;
- (ii) escolher a cidade "mais apropriada" e inseri-la na posição calculada anteriormente (os critérios de escolha aqui usados serão discutidos mais adiante);
- (iii) se todas as cidades foram já incluídas na rota, terminar; senão, retornar a (i).

Quando se insere a última cidade na rota, obtém-se um percurso inicial admissível, mas que é, em geral, pouco satisfatório.

Uma forma "tradicional" de melhorar a solução inicial é substituir  $K$  arcos da rota inicial por outros  $K$  arcos diferentes ( $K$ -permutação), dizendo-se que uma rota é  $K$ -ótima se não for possível obter outra rota que aumente a satisfação do nosso objectivo, aplicando apenas  $K$ -permutações.

Como é evidente, o número de permutações aumenta com  $K$ . Com efeito, para  $K=2$  existem  $\binom{N}{3}$  permutações possíveis e para  $K=3$  existem  $\binom{N}{2}$ , o que corresponde a uma complexidade global de  $O(N^2)$  e  $O(N^3)$  respectivamente. Pelo que, neste trabalho, se consideraram apenas as 2-permutação (ver figura 1).

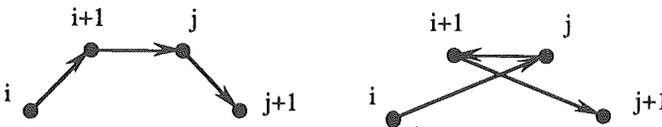


Fig. 1 – Uma 2-permutação

A principal dificuldade do método das permutações, quando aplicado ao TSPTW, reside no tempo necessário para verificar a validade de uma permutação, quando se consideram janelas temporais. Este tempo é de ordem  $O(N)$ , o que implica uma complexidade de  $O(N^3)$  para um ciclo completo de permutações.

No presente trabalho, apresentam-se algoritmos que conseguem reduzir este tempo de verificação para  $O(1)$ , resultando num tempo total de  $O(N^2)$ . O método apresentado baseia-se no trabalho realizado por M. Savelsbergh (descrito no artigo "Local search in routing problems with Time Windows", *Annals of Operations Research* 4 (1985/86) 285-305).

A estratégia usada para executar as permutações é a seguinte:

- (i) Fixar cada arco  $(i, i+1)$  na ordem em que ele aparece na rota actual, começando no arco  $(1, 2)$ , e até ao arco  $(N-1, N)$ , nessa ordem;

- (ii) Escolher os arcos  $(j, j+1)$ , sendo estes iguais a  $(i+2, i+3), (i+3, i+4), \dots, (N-1, N)$  nesta ordem.

Considerem-se todas as permutações possíveis do arco fixado  $(i, i+1)$ . Usando permutações de 2 arcos, verifica-se que ao troço da permutação anterior  $(i+1, \dots, j-1)$  é agora acrescentado o novo arco  $(j-1, j)$ . Pode-se, assim, aproveitar a informação referente à permutação anterior para calcular o comprimento do novo troço  $(i+1, j)$ , e verificar a sua admissibilidade, em tempo constante.

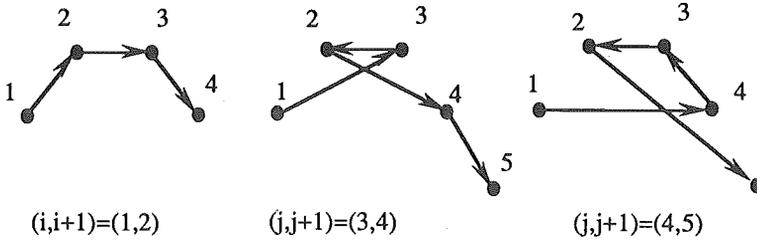


Fig. 2 – Fixando  $(1, 2)$ , as 2 primeiras trocas

De modo a facilmente se poder distinguir a permutação que originou um determinado arco, usar-se-á como índice:

- $(j, j+1)$  - nos arcos que se referem à permutação actual;
- $(j-1, j)$  - nos arcos que se referem à permutação anterior.

### 5.2. Notação adicional

Antes de se apresentar uma descrição sucinta dos algoritmos, introduz-se alguma notação adicional, que diz respeito às quantidades a considerar:

- $T_{ij}$  – distância a percorrer, para ir directamente da cidade  $i$  à cidade  $j$ .
- $A_i$  – data de chegada à cidade  $i$ .
- $D_i$  – data de partida da cidade  $i, D_i = \text{Max}(A_i, e_i)$ .
- $W_i$  – tempo de espera em  $i, W_i = D_i - A_i$ .
- PFS – "Possible Forward Shift" - o possível avanço no tempo de partida de  $j$ , não violando as janelas ao longo do troço  $(j, \dots, i+1)$ .

$$\text{PFS}^{(j,j+1)} = \min \left\{ l_k - D_j^{(j,j+1)} + \sum_{p=k}^{j-1} (T_{p,p+1}) \right\}$$

TWT – tempo de espera total, no troço  $(j, \dots, i+1)$  excluindo o tempo de espera em  $j$  e incluindo o tempo de espera em  $i+1$ .

$$\text{TWT}^{(j,j+1)} = \sum_{k=i+1}^{j-1} W_k$$

TTT – tempo de viagem total (excluindo os períodos de espera) no troço  $(j, \dots, i+1)$ .

$$\text{TTT}^{(j,j+1)} = \sum_{k=i+1}^{j-1} (T_{k,k+1})$$

$$\text{SHIFT}^{(j,j+1)} = D_j^{(j,j+1)} + T_{j,j-1} - D_{j-1}^{(j-1,j)}$$

### 5.3. Construção de uma rota inicial

#### 5.3.1. Estratégia geral

O procedimento de inserção de uma cidade  $u$ , entre as cidades  $i$  e  $j$ , na rota em construção (que será repetido até todas as cidades fazerem parte da rota), tem por base os dois parâmetros seguintes: C1 para determinar a melhor posição de inserção, e C2 para escolher qual das cidades deverá ser inserida.

Sendo  $(i_1, i_2, i_3, \dots, i_m)$  a rota parcial já construída, definem-se C1 e C2 da forma seguinte:

$$C1(i_u, u, i_{u+1}) = \min \{ C1(i_p, u, i_{p+1}) \}$$

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

$$C2(i_u^*, u, i_{u+1}^*) = \min \{ C2(i_u, u, i_{u+1}) \}$$

$$u \text{ fora da rota}$$

#### 5.3.2. Janelas largas e apertadas

Na escolha da cidade a inserir, o critério a utilizar depende fortemente da largura da janela temporal considerada. Se uma janela é relativamente larga, o aspecto espacial revela-se mais importante, pelo que se deverá minimizar o aumento no comprimento da rota. No entanto, se uma janela é apertada, o aspecto temporal torna-se predominante.

Consideram-se duas etapas no algoritmo de construção de uma rota inicial: na primeira, são inseridas as cidades com janelas apertadas, e na segunda, as cidades com janelas largas. Note-se que a distinção entre janelas largas e apertadas é deixada ao critério do utilizador, por forma a aumentar a flexibilidade do procedimento.

#### 5.3.3. Critérios de inserção

Defina-se a quantidade  $em$  (extra mileage) como o custo de inserção da cidade  $u$  no arco  $(i,j)$ :

$$em(i, u, j) = \max(D_i + T_{i,u}; e_u) + T_{u,j} - A_j$$

ou seja,  $em(i, u, j) = A_j' - A_j$ , sendo  $A_j'$  o tempo de chegada a  $j$ , anteriormente calculado.

Na primeira fase da inserção, em que o aspecto temporal é o mais importante, são considerados os critérios referidos, pela ordem seguinte:

$$C1(i_u, u, i_{u+1}) = \max \{ \min(l_u - \max(D_{ip} + T_{ip,u}; e_u); PFS_{ip+1} - em(i_p, u, i_{p+1})) \}$$

$p = 1, \dots, m$  admissível

$$C2(i_u^*, u, i_{u+1}^*) = \min \{ em(i_u, u, i_{u+1}) \}$$

$u$  fora da rota

Com o primeiro critério procura-se inserir a cidade na posição em que se maximiza a flexibilidade do resto da rota em construção, no que respeita às janelas temporais. Com o segundo tenta-se minimizar o aumento no comprimento da rota.

Na segunda fase, em que são inseridas as cidades com janelas largas, é valorizado o aspecto espacial, invertendo-se, por isso, a ordem dos critérios.

Por último, note-se que a construção de uma rota inicial envolve uma elevada quantidade de cálculos, devido à necessidade de verificar se cada nova inserção é, ou não, admissível. Este aspecto foi, de certa forma, menosprezado no artigo referido. Foram, por isso, feitas adaptações por forma a tornar esta etapa do algoritmo bastante mais eficiente. Para o efeito, sempre que uma inserção é efectuada, são guardados todos os parâmetros a ela referentes, de modo a facilitar a verificação da validade da próxima inserção. Esta técnica teve como resultado uma redução de tempos computacionais na ordem das 20 vezes.

## 5.4. Estratégia de permutações

### 5.4.1. Obtenção da solução 2-óptima

Depois de obtida uma rota inicial admissível, pretende-se melhorar o valor da função objectivo, recorrendo à estratégia de permutações, atrás descrita. Para obter a solução 2-óptima, adoptou-se o procedimento seguinte:

- (i) Executar um ciclo completo de permutações, escolhendo, naquelas que são admissíveis, a que resultou numa melhoria mais significativa da função objectivo;
- (ii) Se não houver nenhuma melhoria, terminar; se não, executar a permutação escolhida em (i), passando a rota resultante a constituir a nova rota inicial;
- (iii) Retornar a (i).

### 5.4.2. Admissibilidade de uma permutação

Uma nova rota é admissível se se verificar:

$$\text{SHIFT}^{(j,j+1)} \leq \text{PFS}^{(j-1,j)}$$

Como resulta da figura 3, ao executar a permutação actual, não é necessário analisar a admissibilidade em  $j$  pois verifica-se que demora sempre "menos tempo" ir directamente de  $i$  para  $j$ , do que através do troço  $i+1, i+2, \dots, j-1$ .

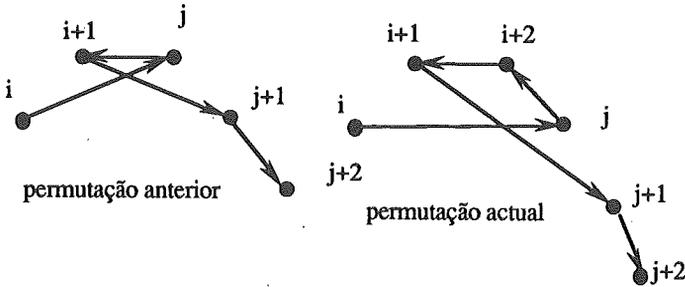


Fig. 3 - A sequência de duas permutações

É fácil comprovar a admissibilidade em  $j+1$  e verificar se houve melhoramento até àquele ponto, bastando para isso saber o tempo de partida de  $i+1$  mais a distância a  $j+1$ . Sendo assim, a condição de admissibilidade em  $j+1$  vem:

$$D_{i+1}^{(j,j+1)} + T_{i+1,j+1} \leq l_{j+1}$$

sendo  $D_{i+1}$  calculado por:

$$D_{i+1}^{(j,j+1)} = D_j^{(j,j+1)} + TTT^{(j,j+1)} + TWT^{(j,j+1)}$$

As quantidades usadas podem ser actualizadas, em tempo constante, da seguinte forma:

$$TTT^{(j,j+1)} = TTT^{(j-1,j)} + T_{j,j-1}$$

$$TWT^{(j,j+1)} = \max(TWT^{(j-1,j)} - \text{SHIFT}^{(j,j+1)}, 0)$$

$$PFS^{(j,j+1)} = \min(PFS^{(j-1,j)} - \text{SHIFT}^{(j,j+1)}, l_j - D_j^{(j,j+1)})$$

Facilmente se comprova que a verificação da admissibilidade, o teste de melhoria local, e a actualização das quantidades envolvidas requerem tempo  $O(1)$ , o que resulta numa complexidade global de  $O(N^2)$ .

As quantidades referidas permitem saber se houve uma melhoria local em  $j+1$ . Para verificar se essa melhoria se propaga até à cidade onde a rota termina, é necessário calcular o tempo de chegada a essa cidade. Tal cálculo torna-se trivial se se souber o valor de  $D_{i+1}$  anterior. Assim, se for  $P$  a cidade de partida e chegada, e se se verificar:

$$A_P^{(j,j+1)} < A_P^{\text{melhor}}$$

(onde  $A_P$  é o melhor valor obtido até ao momento), a nova solução é, então, melhor que a anterior. São guardados agora os valores de  $i$  e  $j$  que identificam a melhor permutação até ao momento. No fim de um ciclo completo de permutações, a melhor permutação (das admissíveis) levará à construção da nova rota.

### 5.5. Critérios de optimalidade

Podem-se, para o problema do TSPTW, considerar, em alternativa, diferentes funções objectivo:

- 1) Minimizar o tempo total de viagem

$$\min \left\{ \sum_{i=1}^{N-1} (T_{i,i+1}) + T_{(N,1)} \right\};$$

- 2) Minimizar a soma dos tempos de espera

$$\min \left\{ \sum_{i=1}^{N-1} W_i \right\};$$

- 3) Minimizar o tempo total demorado a percorrer a rota.

Por corresponder mais correctamente à situação prática em estudo, considera-se, no presente trabalho, o terceiro destes objectivos.

## 6 - Implementação Computacional

### 6.1. Sistema de Apoio à Decisão

Para o problema em estudo foi desenvolvido em C um primeiro protótipo, prevendo-se, no futuro, proceder à sua evolução. O produto final poderá vir a integrar-se em sistemas informáticos de apoio ao planeamento e gestão da produção em pequenas e médias empresas industriais, especialmente no caso das metalomecânicas.

Trata-se de um Sistema de Apoio à Decisão, cuja funcionalidade aliada a uma grande rapidez na obtenção de soluções, fazem com que constitua uma preciosa ferramenta no apoio ao planeamento da produção, ao nível oficial.

É possível ao utilizador deste sistema, realizar diversas simulações de um modo simples e rápido. Foi, nesse sentido, dada uma grande importância ao interface, sendo possível a análise dos resultados nas formas numéricas e gráficas. Deste modo, o utilizador pode facilmente comparar várias soluções e, analisando os valores fornecidos, escolher a que melhor o satisfaz.

Note-se que, grande parte das pequenas e médias empresas industriais ainda não faz um agrupamento dos produtos por famílias, no sentido aqui considerado. O uso de algoritmos deste tipo leva à necessidade de introdução deste conceito, o qual pode conduzir a alterações adicionais ao nível da organização do sector produtivo, com significativas melhorias na gestão da produção. Como é evidente, tal exigirá uma informatização da empresa, dada a grande quantidade de informação envolvida.

## 6.2. Dados de entrada

Para além dos dados directamente associados às tarefas (tempo de processamento; família; "earliest time"; "latest time") e da matriz de "setup" entre famílias, existem dois parâmetros de grande importância no cálculo do escalonamento.

O primeiro desses parâmetros é a "largura de distinção" que permite distinguir as janelas apertadas das largas. Diferentes valores, para este parâmetro, originarão diferentes soluções, competindo ao utilizador escolher o valor que melhor se adapte ao problema em análise.

O segundo desses parâmetros é o "máximo atraso permitido". Se se obrigar as tarefas a terminarem todas dentro das janelas atribuídas, verifica-se, muito frequentemente, a impossibilidade de construir uma solução inicial admissível. Por isso, é permitido ao utilizador atribuir um atraso máximo às tarefas. É, assim, possível fazer simulações com vista a determinar o mínimo atraso permitido que gera um escalonamento admissível.

## 6.3. Dados de saída

O programa produz uma listagem do escalonamento calculado, onde se podem consultar diversos dados com interesse para a análise e implementação da solução produzida.

Nesta listagem (ver figura 4), são apresentadas as tarefas escalonadas por ordem cronológica, indicando-se os tempos de início e fim de cada uma. Para uma mais fácil avaliação das soluções, são ainda apresentados os dados iniciais referentes à janela temporal das tarefas, o tempo que a tarefa é obrigada a esperar (desde o momento em que se encontra disponível para ser processada) e, o atraso da tarefa em relação ao "latest time". Esta listagem inclui ainda as operações de "setup" e de paragem, apresentando os respectivos tempos de duração (entre parênteses). Finalmente, apresentam-se diversas estatísticas como os tempos totais gastos em "setup" e em períodos de vazio, e ainda a soma dos tempos de espera das diversas tarefas.

Tempo	Operação	Fim	Earliest Time	Latest Time	Tempo de espera	Tempo de atraso
0	Espera	(85)	—	—	—	—
85	Tarefa 2	141	85	2205	0	0
141	Espera	(34)	—	—	—	—
175	Tarefa 14	370	175	2292	0	0
370	Setup	(141)	—	—	—	—
...	...	...	...	...	...	...

Fig. 4 - Excerto de uma listagem referente ao escalonamento calculado

O escalonamento calculado pode ainda ser representado graficamente sob a forma de um histograma de tarefas (conforme indica a figura 5), onde o eixo horizontal indica o tempo

decorrido desde o início da primeira tarefa, correspondendo cada padrão a um tipo de operação, conforme indica a legenda da figura.

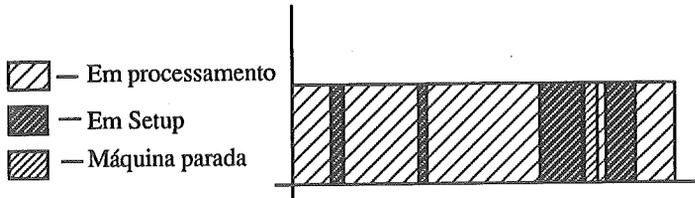


Fig. 5 - Histograma de Operações

**7. Resultados Computacionais Preliminares**

Apresentam-se de seguida, alguns problemas exemplificativos do funcionamento do programa e da forma como os resultados são apresentados.

**7.1. Problema A**

Dados referentes às tarefas:

Tarefa	Tempo Process.	Família	Earliest time	Latest time
1	100	A	0	500
2	100	A	0	700
3	100	C	100	1000
4	100	B	0	1000
5	100	B	0	1000
6	100	B	200	600
7	100	C	100	900
8	100	A	150	500
9	100	E	100	1200

Matriz de tempos de "setup" (por família):

	A	B	C	D	E
A	0	60	60	60	60
B	50	0	50	50	50
C	20	20	0	20	20
D	30	30	30	0	30
E	40	40	40	40	0

## ESCALONAMENTO CALCULADO

Tempo	Operação	Fim	Earliest Time	Latest Time	Tempo de Espera	Tempo de Atraso
0	Tarefa 1	100	0	500	0	0
100	Tarefa 2	200	0	700	100	0
200	Tarefa 8	300	150	500	50	0
300	Setup	(60)	—	—	—	—
360	Tarefa 6	460	200	600	160	0
460	Setup	(50)	—	—	—	—
510	Tarefa 3	610	100	1000	410	0
610	Tarefa 7	710	100	900	510	0
710	Setup	(20)	—	—	—	—
730	Tarefa 4	830	0	1000	730	0
830	Tarefa 5	930	0	1000	830	0
930	Setup	(50)	—	—	—	—
980	Tarefa 9	1080	100	1200	880	0

Somatório dos tempos de espera - 3670

	Nº	Tempo
"Setups"	4	180
Paragens	0	0
Total	4	180

## 7.2. Problema B

Ao problema anterior, foram feitas alterações no plano de produção, de modo a não existirem tempos de espera na secção de pintura (lógica "Just In Time").

O novo quadro de dados referentes às tarefas é o seguinte:

Tarefa	Tempo Process.	Família	Earliest time	Latest time
1	100	A	0	500
2	100	A	100	700
3	100	C	810	1000
4	100	B	560	1000
5	100	B	460	1000
6	100	B	360	600
7	100	C	710	900
8	100	A	200	500
9	100	E	930	1200

#### ESCALONAMENTO CALCULADO

Tempo	Operação	Fim	Earliest Time	Latest Time	Tempo de Espera	Tempo de Atraso
0	Tarefa 1	100	0	500	0	0
100	Tarefa 2	200	100	700	0	0
200	Tarefa 8	300	200	500	0	0
300	Setup	(60)	—	—	—	—
360	Tarefa 6	460	360	600	0	0
460	Tarefa 5	560	460	1000	0	0
560	Tarefa 4	660	560	1000	0	0
660	Setup	(50)	—	—	—	—
710	Tarefa 7	810	710	900	0	0
810	Tarefa 3	910	810	1000	0	0
910	Setup	(20)	—	—	—	—
930	Tarefa 9	1030	930	1200	0	0

Somatório dos tempos de espera - 0

	Nº	Tempo
"Setups"	3	130
Paragens	0	0
-----	-----	-----
Total	3	130

NOTA: Como se observa, a solução obtida foi melhor que a do problema anterior

## 8. Conclusões

O sistema informático descrito neste trabalho poderá constituir um auxiliar de grande utilidade no planeamento de sistemas produtivos onde seja possível identificar "máquinas críticas", isto é máquinas que constituam "estrangulamentos" da sequência operatória.

Como é sabido, se, ao longo de um fluxo produtivo, existir uma "máquina crítica", toda a cadência produtiva será marcada pela velocidade de processamento dessa máquina. Pelo que, neste caso, um planeamento da produção que tenha como objectivo, por exemplo, a maximização da utilização das máquinas a montante do "bottleneck", será provavelmente destituído de sentido, pois os lotes vão ter de esperar junto da "máquina crítica", provocando problemas de falta de espaço e a ocorrência de stocks excessivos.

A solução para o problema será provavelmente o escalonamento dos lotes a montante da "máquina crítica", a partir do escalonamento gerado para a referida máquina. Deste modo, cada lote deverá chegar à "máquina crítica" o mais perto possível do tempo de início de processamento, previsto no escalonamento inicial. Evitam-se, assim, os tempos de espera, recorrendo-se a uma estratégia de produção que tenderá para o "Just In Time".

Os escalonamentos gerados, principalmente se o número de tarefas for elevado, aparentam ser piores quanto mais próximo se estiver do fim do horizonte temporal escolhido. Isto acontece porque a quantidade de informação disponível referente à procura decresce com o tempo e, perto do fim do horizonte, já pouco se sabe quanto aos lotes esperados.

Este facto não constitui problema dado que, na prática, apenas a parte inicial do escalonamento será cumprida, já que a constante alteração do ambiente (como a chegada de novos lotes) obrigará a refazer frequentemente o planeamento. Deste modo, apenas a parte inicial dos escalonamentos deverá ser tida em conta.

Far-se-á finalmente alguns comentários ao valor escolhido para a "largura de distinção" que permite separar as tarefas em dois grupos: um com janelas apertadas e outro com janelas largas. Caso existam diferenças significativas entre os valores da largura das janelas, por exemplo uma janela bastante apertada em relação às outras, deve-se escolher um valor tal que distinga as janelas apertadas, de modo a assegurar que estas são inseridas na rota, logo no início do algoritmo. Se não se puder detectar diferenças significativas, concluiu-se, empiricamente, que um valor adequado é o da máxima largura de janela existente, ou seja considera-se que todas as janelas são apertadas.

A selecção deste valor é determinante no sucesso da geração de uma rota inicial admissível, pelo que se recomenda o teste de valores alternados até se permitir a introdução de atrasos. Por outro lado, valores diferentes para este parâmetro podem dar origem a escalonamentos admissíveis associados a sequências diferentes de tarefas, com diferentes valores da função

objectivo, pelo que o teste de vários valores, para a "largura de distinção", poderá melhorar as soluções obtidas.

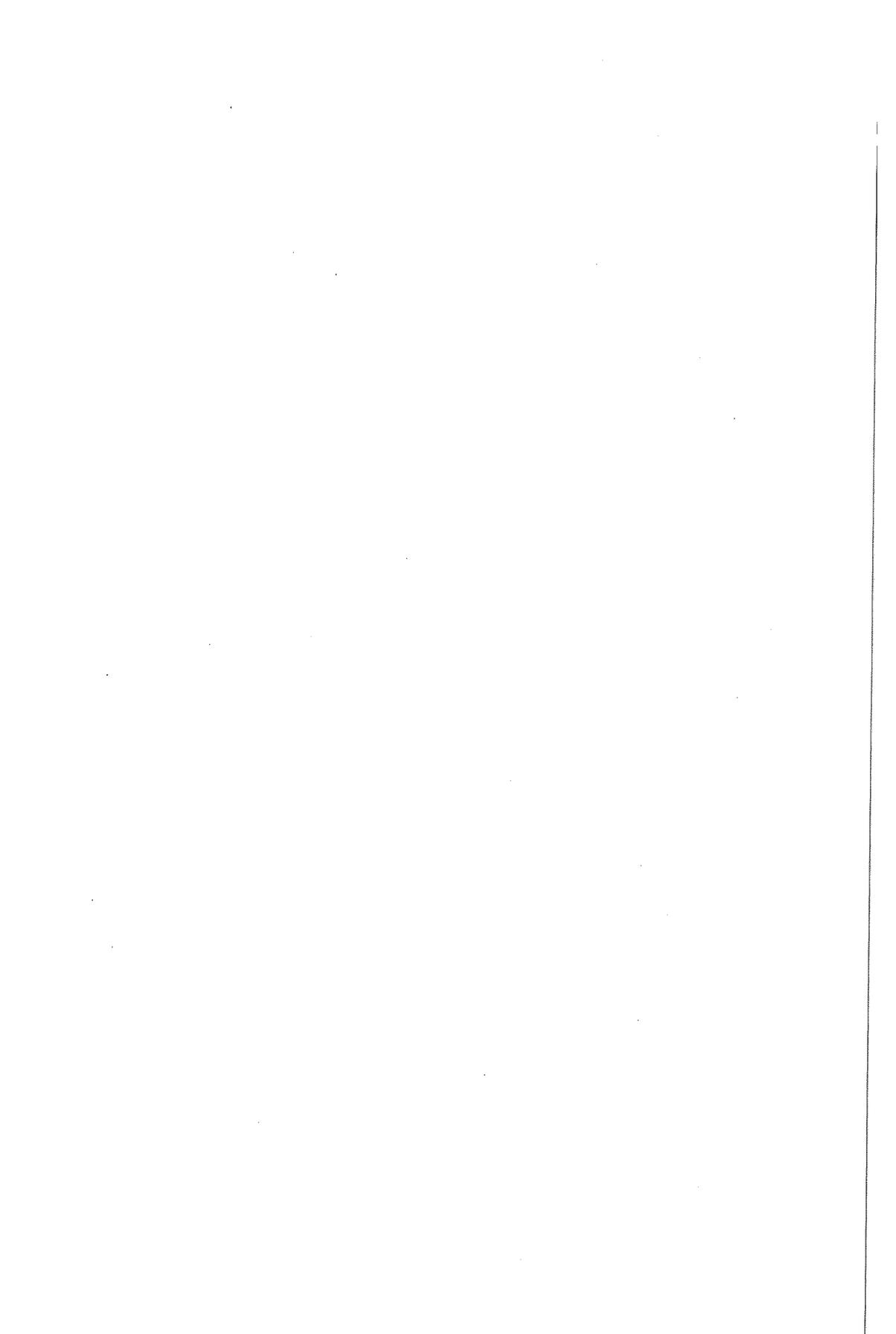
Note-se finalmente que, tratando-se de um protótipo, o sistema apresentado não incorpora ainda alguns aspectos que podem, em situações práticas, ser relevantes.

Conhecem-se, por exemplo, situações em que existe um problema adicional que se prende com o horário laboral, o qual inclui um tempo de paragem total da fábrica durante o período nocturno. Esta possibilidade não está prevista na versão actual do sistema, pois parte-se do princípio que a laboração é contínua.

Para resolver este problema, será criado um tipo de tarefa fictícia com uma janela temporal fixa, correspondente ao período da noite, e um tempo de processamento igual à duração deste período. Os tempos de "setup" de qualquer das outras tarefas para uma tarefa deste tipo, serão nulos, o que corresponde à realidade dado que os "setups" realizados no início de um novo dia não afectam o plano, por serem executados antes do início da laboração.

#### **Agradecimentos:**

Agradeço ao Prof. Jorge Pinho de Sousa todo o seu apoio moral e científico. Só a sua inextinguível dedicação tornou possível a realização deste trabalho.



# ESCALONAMENTO DE PESSOAL TRIPULANTE: UTILIZAÇÃO DE UMA TÉCNICA DE RELAXAÇÃO DE ESPAÇO DE ESTADOS PARA PROBLEMAS DE COBERTURA

Ana Paias  
D.E.I.O.  
Faculdade de Ciências  
Av. 24 de Julho, 134-5º  
1300 LISBOA

## Abstract

This paper reports on a lower bound technique based on state space relaxation for a dynamic program associated with a particular class of covering problems, namely covering problems related with crew scheduling. Both, the dynamic program and the state space relaxation are general but the latter revealed a very interesting behavior for this current application. Some reduction and dominance tests were derived from the relaxation. Two additional tests were developed in order to speed up the process. These tests are based on conditions that state whether some stages or some states in a particular stage can be ignored. The state space relaxation was combined with lagrangean relaxation and both were imbeded in a subgradient optimization scheme. Some computational results from real life applications are presented together with possible improvements on the technique. The inclusion of the tests described in this paper produces significant reduction on the dimension of the original problems and also on the computing times.

## Resumo

Na presente comunicação aborda-se a particularização de uma técnica de relaxação de espaço de estados a problemas de cobertura relacionados com a determinação de horários de tripulações num sistema de transporte colectivo de passageiros. Esta técnica permite a obtenção de limites inferiores para o valor óptimo do problema de cobertura associado. A partir da relaxação deduzem-se penalidades e testes de dominância que permitem reduzir significativamente as dimensões das instancias tratadas. São ainda desenvolvidos dois testes adicionais de forma a acelerar o processo de obtenção dos limites. Estes testes baseiam-se em condições que estabelecem quando é possível ignorar alguns estados ou mesmo uma fase do programa dinâmico. A relaxação de espaço de estados é combinada com relaxação lagrangeana e ambas são incluídas num esquema de optimização pelo método do subgradiente. Discutem-se, ainda, algumas extensões possíveis e são apresentados alguns resultados computacionais referentes a diversos casos de aplicação real.

## Keywords

Crew Scheduling, Set Covering, Dynamic Programming

## 1. Introdução

O objectivo deste trabalho consiste no estudo, desenvolvimento e implementação de um método para a obtenção de limites inferiores para o valor óptimo de problemas de cobertura associados a escalonamento de pessoal. Este método combina técnicas de relaxação lagrangeana com técnicas de relaxação de espaço de estados.

O problema da determinação dos horários de pessoal tripulante consiste em determinar o conjunto de serviços que assegure com custo mínimo o serviço rodoviário diário. Esse conjunto pode ser determinado de vários modos, existindo na literatura (ver por exemplo [Wren, 1981], [Rousseau, 1985] e [Daduna & Wren, 1988]) algoritmos baseados em métodos de programação matemática, métodos heurísticos e métodos interactivos.

No presente trabalho, o problema em estudo é formulado como um modelo de cobertura de custo mínimo de um conjunto (ver por exemplo [Paixão et al, 1985], [Paias, 1988], [Paixão, 1990]). Assume-se, para o efeito, que o serviço a efectuar por cada viatura foi previamente determinado. Os serviços das viaturas podem ter uma duração superior à duração máxima permitida para um horário de uma tripulação, pelo que são divididos nos pontos de transferência formando-se assim *peças de trabalho*. Os pontos de transferência correspondem às paragens das linhas, em que é permitida a rendição da tripulação. Assim, após a definição das peças de trabalho, geram-se todos os serviços admissíveis, ou seja, geram-se todas as combinações de peças de trabalho que verificam as normas contratuais (por exemplo, período mínimo de serviço diário, máximo período em serviço extraordinário, etc). O melhor conjunto de serviços que assegure o serviço rodoviário diário, pode ser obtido resolvendo um problema da determinação da cobertura de custo mínimo de um conjunto, em que se garante que cada peça de trabalho é realizada (coberta) por pelo menos um serviço.

Considerem-se, os conjuntos  $N = \{1, 2, \dots, n\}$  e  $M = \{1, 2, \dots, m\}$  correspondentes a uma indexação dos conjuntos, respectivamente, dos serviços admissíveis e das peças de trabalho a realizar. Segundo a terminologia vulgarmente utilizada em problemas de cobertura, os serviços designam-se por colunas e as peças de trabalho por linhas. Para cada coluna  $j$  ( $j = 1, \dots, n$ ), designa-se por  $M_j$ , o conjunto dos índices das linhas por ela cobertas.

São ainda conhecidos custos  $c_j$  ( $j \in N$ ) associados aos serviços e definem-se os elementos  $a_{ij}$  ( $i \in M, j \in N$ ) tais que:

$$a_{ij} = \begin{cases} 1 & \text{se o serviço } j \text{ realiza a peça } i \\ 0 & \text{caso contrário} \end{cases}$$

Considerem-se, ainda, as variáveis do problema, definidas da seguinte forma:

$$x_j = \begin{cases} 1 & \text{se o serviço } j \text{ está na solução óptima} \\ 0 & \text{caso contrário} \end{cases}$$

O problema da determinação dos serviços do pessoal tripulante pode ser formulado matematicamente do modo seguinte:

$$(PCC) \quad \min \quad \sum_{j=1}^n c_j x_j \quad (1.1)$$

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad i \in M \quad (1.2)$$

$$x_j \in \{0, 1\} \quad j \in N \quad (1.3)$$

O programa matemático (1.1) - (1.3) é conhecido na literatura da especialidade por problema da cobertura de custo mínimo de um conjunto, o qual tem sido profundamente estudado ao longo das últimas décadas. De facto, existem diversas abordagens baseadas nomeadamente na utilização de heurísticas, relaxação linear, relaxação lagrangeana e pesquisa em árvore.

Neste trabalho, considera-se uma abordagem em termos de programação dinâmica inicialmente apresentada em [Paixão e Christofides, 1986]. Em particular, desenvolve-se uma

aplicação da técnica da relaxação de espaço de estados ao programa dinâmico apresentado nesse trabalho.

O problema da cobertura de custo mínimo de um conjunto goza da propriedade de a sua resolução poder ser encarada como uma sequência de  $n$  decisões, em que em cada decisão  $j$  se pretende saber se a coluna  $j$ , vai ou não, pertencer à cobertura óptima.

Seja  $S$  um subconjunto de linhas e  $F_k(S)$  o custo mínimo para cobrir  $S$  usando as  $k$  primeiras colunas. A equação dinâmica pode ser escrita do seguinte modo [Paixão e Christofides, 1986] :

$$F_k(S) = F_{k-1}(S) \quad e$$

$$F_k(S \cup M_k) = \min [F_{k-1}(S \cup M_k), F_{k-1}(S) + c_k] \quad (1.4)$$

para todo o  $S \in \mathcal{T}_{k-1}$ ;  $k = 1, \dots, n$  e em que  $\mathcal{T}_k = \mathcal{T}_{k-1} \cup \{S : S = S' \cup M_k \text{ e } S' \in \mathcal{T}_{k-1}\}$  e  $\mathcal{T}_0 = \{\emptyset\}$ . O valor  $F_{k-1}(S \cup M_k)$  só é calculado se  $(S \cup M_k) \in \mathcal{T}_{k-1}$ . A equação dinâmica é inicializada fazendo  $F_0(\emptyset) = 0$ .

O programa dinâmico (1.4) tem associado um espaço de estados cuja dimensão é  $O(2^m)$ , i.e. cresce de forma explosiva com o número de linhas do problema de cobertura, o que implica que a memória requerida bem como o tempo de CPU, necessários para a obtenção de uma solução óptima são extremamente elevados, mesmo para problemas de pequena dimensão. Este facto torna impraticável a utilização daquela abordagem para resolução de ocorrências associadas a situações da vida real.

Uma via para ultrapassar esta dificuldade consiste em procurar reduzir a dimensão do espaço de estados através da aplicação de uma técnica de relaxação. É o que estudaremos a partir da próxima secção e em que exploraremos as ideias inicialmente propostas em [Christofides & Paixão, 1984] e [Paixão & Christofides, 1986]. Registe-se, desde já, que quando tal redução é operada se perde a garantia de optimalidade relativamente ao problema de cobertura original. De facto, como veremos adiante, passa-se a dispôr de um limite inferior para o custo da solução óptima.

## 2. Relaxação de Espaço de Estados

Como já foi mencionado, o objectivo de uma relaxação deste tipo é reduzir a dimensão do espaço de estados associado à equação original. Esta redução é feita considerando um novo espaço de estados  $Q$  (espaço de estados relaxado), que é a imagem do espaço de estados anterior por aplicação de uma função  $g$  (função "mapping") que verifica determinadas condições [Paixão & Christofides, 1986].

A redução da dimensão do espaço de estados deve-se ao facto de  $g$  ser uma função não injectiva, podendo existir estados que são distintos em  $\mathcal{T}$  mas com imagem idêntica em  $Q$ . Em consequência desta redução no espaço de estados, perde-se a garantia de optimalidade.

Na escolha da função  $g$  deve-se ter em conta que quanto mais estados tiverem a mesma imagem por aplicação da função  $g$ , menor será a dimensão do novo espaço de estados e

portanto maior será a rapidez de cálculo. Por outro lado, quanto menor for a dimensão do novo espaço de estados pior será a qualidade do limite inferior obtido.

Em [Paixão & Christofides, 1986] prova-se que para qualquer subconjunto de linhas S e para qualquer estágio k, é válida a seguinte desigualdade:

$$f_k(g(S)) \leq F_k(S) \quad k = 0, 1, \dots, n \text{ e } S \subseteq M \quad (2.1)$$

em que  $f_k(\cdot)$  é a função recursiva no espaço de estados relaxado e g a função "mapping".

Em particular tem-se que:

$$f_n(g(M)) \leq F_n(M) = v(\text{PCC}) \quad (2.2)$$

pois trata-se de um problema de minimização, onde v(PCC) representa o valor óptimo do problema (PCC).

Neste trabalho estuda-se uma relaxação de espaço de estados baseada numa das relaxações apresentadas em [Paixão e Christofides, 1986], particularmente interessante para instâncias de problemas de escalonamento e que se apresenta de seguida:

$$g : S \rightarrow g(S) = (|S|, \alpha_s, \beta_s)$$

em que

$$|S| = \text{cardinal de } S$$

$$\alpha_s = \text{índice da 1ª linha em } S$$

$$\beta_s = \text{índice da última linha em } S$$

Considerando a função g atrás referida obtém-se a seguinte função recursiva definida no espaço de estados relaxado Q:

$$\begin{aligned} f_k(\mathcal{A}, \alpha, \beta) &= f_{k-1}(\mathcal{A}, \alpha, \beta) \\ f_k\{(\mathcal{A}, \alpha, \beta) \oplus (\mathcal{A}_k, \alpha_k, \beta_k)\} &= \\ &= \min [f_{k-1}\{(\mathcal{A}, \alpha, \beta) \oplus (\mathcal{A}_k, \alpha_k, \beta_k)\}, f_{k-1}(\mathcal{A}, \alpha, \beta) + c_k] \end{aligned} \quad (2.3)$$

A equação acima apresentada define-se para  $k = 1, \dots, n$  e  $(\mathcal{A}, \alpha, \beta) \in Q_{k-1}$ , correspondendo  $(\mathcal{A}_k, \alpha_k, \beta_k)$  respectivamente ao cardinal de  $M_k$ , índice da primeira e última linhas em  $M_k$ , e  $\oplus$  representa a seguinte operação:

$$(\mathcal{A}, \alpha, \beta) \oplus (\mathcal{A}_k, \alpha_k, \beta_k) = (\min(\mathcal{A} + \mathcal{A}_k - \Delta_{\alpha, \beta, \alpha_k, \beta_k}, \beta' - \alpha' + 1), \alpha', \beta')$$

em que  $\alpha' = \min(\alpha, \alpha_k)$ ,  $\beta' = \max(\beta, \beta_k)$  e  $\Delta_{\alpha, \beta, \alpha_k, \beta_k} = \delta_{\alpha\alpha_k} + (1 - \delta_{\alpha\alpha_k}) \delta_{\alpha\beta_k} + (1 - \delta_{\alpha\beta}) \delta_{\alpha\beta_k} + (1 - \delta_{\alpha\beta})(1 - \delta_{\alpha\alpha_k}) \delta_{\alpha\beta_k}$  em que  $\delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{caso contrário} \end{cases}$

Note-se que o valor  $\Delta_{\alpha, \beta, \alpha_k, \beta_k}$  ( $0 \leq \Delta_{\alpha, \beta, \alpha_k, \beta_k} \leq 2$ ) vai indicar se se contabilizou mais do que uma vez as linhas de maior e menor índice.

Em cada fase k ( $k = 1, \dots, n$ ), o subespaço  $Q_k$  obtém-se recursivamente a partir de  $Q_{k-1}$  do seguinte modo:

$$Q_k = Q_{k-1} \cup \{(\mathcal{A}, \alpha, \beta) \oplus (\mathcal{A}_k, \alpha_k, \beta_k) : (\mathcal{A}, \alpha, \beta) \in Q_{k-1}\} \quad (2.4)$$

De modo a obter-se uma definição coerente para  $\oplus$ , considera-se  $g(\emptyset) = (0, m + 1, 0)$  e toma-se para valor inicial  $Q_0 = \{g(\emptyset)\} = \{(0, m+1, 0)\}$ .

O valor óptimo do problema relaxado será dado por  $f_n(m, 1, m)$ . Este valor corresponde a "cobrir" com custo mínimo  $m$  linhas do problema, correspondendo à primeira linha coberta, a linha com índice 1 e à última, a linha com índice  $m$ . O termo "cobrir" é usado entre aspas porque algumas linhas podem estar contabilizadas mais do que uma vez enquanto outras podem não ser cobertas.

Em conclusão, a solução obtida pode não ser admissível para o PCC, pois como já se referiu, o modo como a operação  $\otimes$  foi definida permite a existência de linhas contabilizadas mais do que uma vez e outras não cobertas.

### 3. Melhoramentos Adicionais

Mesmo considerando as reduções conseguidas por aplicação da técnica da relaxação de espaço de estados, os problemas relaxados atingem ainda, dimensões consideráveis. Assim, existe todo o interesse em estabelecer critérios de dominância de alguns estados face a outros. Um estado dominado por outro pode ser eliminado porque não conduz de certeza, ao estado correspondente à solução óptima do problema relaxado.

Estabeleceu-se [Paias, 1991] o seguinte critério de dominância de estados:

#### Propriedade 3.1:

O estado  $(\mathcal{A}_1, \alpha_1, \beta_1)$  com custo  $f(\mathcal{A}_1, \alpha_1, \beta_1) = c_1$  domina o estado  $(\mathcal{A}_2, \alpha_2, \beta_2)$  de custo  $c_2$  se

$$\mathcal{A}_1 \geq \mathcal{A}_2 + \bar{\delta}_{\alpha_1 \alpha_2} + \bar{\delta}_{\beta_1 \beta_2}$$

$$\alpha_1 \leq \alpha_2$$

$$\beta_1 \geq \beta_2$$

$$c_1 \leq c_2$$

com  $\bar{\delta}_{ij} = 1 - \delta_{ij} = \begin{cases} 1 & \text{se } i \neq j \\ 0 & \text{caso contrário} \end{cases}$

Nos casos reais que se pretendem resolver, o número de colunas é muito elevado, o que contribui para a geração de um número exagerado de estados. Esta técnica de relaxação permite também derivar testes que permitem eliminar, algumas colunas.

#### Propriedade 3.2:

Um limite inferior para o custo de uma solução em que  $x_j=1$  (coluna  $j$  pertence à solução) é dado por:

$$t_j = c_j + f_n(g(M - M_j)) = c_j + \min_{q \in T_j} f_n(q)$$

em que  $T_j = \{q \in Q: q \otimes g(M_j) = (m, 1, m)\}$

Isto é,  $t_j$  é dado pelo custo da coluna  $j$ , adicionado a um limite inferior para o custo de "cobrir" as  $(m - |M_j|)$  linhas não cobertas pela coluna  $j$ .

Seja  $LS$  um limite superior para o  $v(PCC)$ . Se existir uma coluna  $j$  para a qual  $t_j \geq LS$ , conclui-se que qualquer solução admissível com melhor valor do que  $LS$  não contém a coluna  $j$  ( $x_j = 0$ ). Assim, a coluna  $j$  pode ser removida do problema.

No entanto, o cálculo do valor de  $t_j$  envolve a pesquisa de todos os elementos de  $T_j$  e a comparação de  $|T_j|$  valores. Em muitas situações a pesquisa dos elementos  $T_j$  é demasiado demorada e portanto o cálculo de  $t_j$  para cada  $j$  pode envolver consumo de tempo exagerado. Assim, em alternativa ao subespaço  $T_j$ , pode-se considerar um subespaço  $\tilde{T}_j$  que o contenha, e portanto mais fácil de identificar, constituído pelos estados  $q = (\mathcal{A}, \alpha, \beta) \in Q$  com  $\mathcal{A} \geq m - |M_j| + \Delta_{\alpha, \beta, \alpha_j, \beta_j}$  e  $\alpha, \beta$  quaisquer.

### Propriedade 3.3:

Um limite inferior  $\tilde{t}_j$  para  $t_j$  é dado por:

$$\tilde{t}_j = c_j + \min f_n(\mathcal{A}, \alpha, \beta), \forall \alpha, \beta \text{ e para } \mathcal{A} = m - |M_j|, \dots, \beta - \alpha + 1.$$

Note-se que  $\min f_n(\mathcal{A}, \alpha, \beta), \forall \alpha, \beta$  e para  $\mathcal{A} = m - |M_j| + \Delta_{\alpha, \beta, \alpha_j, \beta_j}, \dots, \beta - \alpha + 1$ , indica o custo mínimo de se cobrir, com as  $n$  colunas, pelo menos  $m - |M_j|$  linhas e, portanto, esse valor é um limite inferior para  $f_n(g(M - M_j))$ .

Tendo em conta que  $f_n(m, 1, m)$  é um limite inferior válido para  $c_j + f_n(g(M - M_j))$ , o limite anterior pode ainda ser fortalecido para:

$$\tilde{t}_j = \max[ f_n(m, 1, m), c_j + \min f_n(\mathcal{A}, \alpha, \beta), \forall \alpha, \beta \quad \text{e} \\ \mathcal{A} = m - |M_j| + \Delta_{\alpha, \beta, \alpha_j, \beta_j}, \dots, \beta - \alpha + 1 ]$$

## 4. Relaxação lagrangeana

Considere-se o problema  $(PCCR_\lambda)$ , obtido a partir do problema  $(PCC)$  relaxando à "maneira" lagrangeana as restrições (1.2):

$$(PCCR_\lambda) \quad \min \quad \sum_{j=1}^n c_j x_j + \sum_{i=1}^m \lambda_i \left( 1 - \sum_{j=1}^n a_{ij} x_j \right) = \\ = \sum_{j=1}^n \left( c_j - \sum_{i \in M_j} \lambda_i \right) x_j + \sum_{i=1}^m \lambda_i \quad (4.1)$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n \quad (4.2)$$

em que  $\lambda_i \geq 0$  é o multiplicador de Lagrange (penalidade) associado à  $i$ -ésima restrição do PCC.

Sejam  $N^- = \{j \in N : \bar{c}_j < 0\}$  e  $N^+ = \{j \in N : \bar{c}_j \geq 0\}$  em que  $\bar{c}_j = c_j - \sum_{i \in M_j} \lambda_i$ .

É fácil verificar que uma solução óptima para o  $PCCR_\lambda$  pode ser obtida fazendo:

$$\begin{cases} x_j = 1 & \text{se } j \in N^- \\ x_j = 0 & \text{se } j \in N^+ \end{cases} \quad \text{e, conseqüentemente, o valor óptimo da relaxação lagrangeana}$$

$$v(PCCR_\lambda) = \sum_{j \in N^-} \bar{c}_j + \sum_{i \in M} \lambda_i.$$

Dos resultados gerais estabelecidos para a relaxação lagrangeana [Geoffrion, 1974], sabe-se que  $v(\text{PCCR}_\lambda) \leq v(\text{PCC})$ , qualquer que seja o conjunto escolhido de multiplicadores, desde que  $\lambda_i \geq 0, i = 1, \dots, m$ . A melhor escolha dos multiplicadores, será aquela que conduzirá a um maior valor do limite inferior para  $v(\text{PCC})$ . Assim, pretende-se encontrar o vector de multiplicadores  $\lambda^*$  tal que  $v(\text{PCCR}_{\lambda^*}) = \max_{\lambda \geq 0} v(\text{PCCR}_\lambda)$ .

Uma aproximação de  $\lambda^*$  pode ser obtida através do método de optimização por subgradiente [Shapiro, 1979].

**4.1. Penalidades**

A obtenção de um limite inferior para  $v(\text{PCC})$ , atrás descrita, pode ser acelerada mediante uma redução de colunas obtida através da análise dos custos reduzidos. De facto, dado um vector de multiplicadores de Lagrange,  $\lambda_i \geq 0 (i \in M)$ , o custo reduzido,  $\bar{c}_j$ , associado à variável  $x_j$  pode ser obtido do modo seguinte:

$$\bar{c}_j = c_j - \sum_{i \in M_j} \lambda_i$$

Seja, agora, LS um limite superior conhecido para  $v(\text{PCC})$ . Então para uma dada coluna  $j$  se  $\bar{c}_j + v(\text{PCCR}_\lambda) \geq \text{LS}$ , então essa coluna pode ser removida do problema, pois não figurará em nenhuma solução com melhor valor do que LS.

**5. Obtenção de melhores limites inferiores**

Como já foi referido, a solução obtida pela relaxação de espaço de estados (2.3)-(2.4) pode não ser admissível para o PCC. No entanto, de modo análogo ao que é feito na relaxação lagrangeana, é possível penalizar o valor da função objectivo através das restrições violadas.

Se se acrescentarem as seguintes restrições redundantes à formulação (1.1) - (1.3) :

$$\alpha = \min_{\{j: x_j=1\}} \alpha_j = 1 \qquad \beta = \max_{\{j: x_j=1\}} \beta_j = m \qquad \delta = \sum_{j=1}^n \delta_j x_j \geq m$$

em que  $\alpha_j$  = índice da 1ª linha coberta pela coluna  $j$

$\beta_j$  = índice da última linha coberta pela coluna  $j$

$\delta_j$  = número de linhas cobertas pela coluna  $j$

e  $\alpha$  = índice da 1ª linha coberta na solução óptima

$\beta$  = índice da última linha coberta na solução óptima

$\delta$  = número de linhas "cobertas" na solução óptima

é possível derivar uma relaxação mais forte do que a desenvolvida a partir de (1.1) - (1.3).

Relaxando o conjunto das restrições (1.2), como se fez na secção 4., obtém-se o seguinte problema relaxado:

$$(\text{REE}_\lambda) \min \sum_{j=1}^n (c_j - \sum_{i \in M_j} \lambda_i) x_j + \sum_{i=1}^m \lambda_i \tag{5.1}$$

$$\alpha = \min_{x_j=1} \alpha_j = 1 \tag{5.2}$$

$$\beta = \max_{x_j=1} \beta_j = m \quad (5.3)$$

$$\mathcal{A} = \bigoplus_{j=1}^n \mathcal{A}_j x_j \geq m \quad (5.4)$$

$$x_j \in \{0,1\} \quad j = 1, \dots, n \quad (5.5)$$

O problema (5.1) - (5.5) com  $\lambda = 0$  corresponde precisamente ao problema que se resolve considerando a relaxação de espaço de estados (2.3) - (2.4).

Para cada vector de multiplicadores  $\lambda$  ( $\lambda_j \geq 0$ ), calcula-se um limite inferior  $f_n^\lambda(m,1,m) = v(\text{REE}_\lambda)$  que ainda é um limite inferior para  $v(\text{PCC})$ . A exemplo do que acontecia no ponto anterior, pode-se obter uma aproximação dos multiplicadores óptimos através do método de optimização por subgradiente [Shapiro, 1979]. Em cada iteração do método do subgradiente, podem ser aplicados tanto o teste de dominância de estadosbem como as penalidades que foram propostos na secção 3.

## 6. Combinação da relaxação lagrangeana e da relaxação de espaço de estados

Recorde-se que a formulação (5.2) - (5.5) resulta da formulação (4.1) - (4.2), através da inclusão das seguintes três restrições redundantes:

$$\alpha = \min_{\{j: x_j=1\}} \alpha_j = 1 \quad \beta = \max_{\{j: x_j=1\}} \beta_j = m \quad \mathcal{A} = \bigoplus_{j=1}^n \mathcal{A}_j x_j \geq m$$

Pode-se, então, concluir que para um determinado vector  $\lambda$ ,  $F(\text{PCCR}_\lambda) \supseteq F(\text{REE}_\lambda)$  e  $v(\text{PCCR}_\lambda) \leq v(\text{REE}_\lambda)$  em que  $F(P)$  representa o conjunto das soluções admissíveis do problema  $P$ . Ou seja, para o mesmo conjunto de multiplicadores, o limite inferior obtido pela relaxação de espaço de estados,  $v(\text{REE}_\lambda)$ , é no pior caso, igual ao fornecido pela relaxação lagrangeana,  $v(\text{PCCR}_\lambda)$ .

No entanto, tal como já foi referido, a solução óptima para  $\text{PCCR}_\lambda$  pode ser facilmente obtida fazendo apenas:

$$\begin{cases} x_j = 1 & \text{se } j \in N^- \\ x_j = 0 & \text{se } j \in N^+ \end{cases}$$

Facilmente se verifica que se  $x_j = 1$  na solução óptima de  $\text{PCCR}_\lambda$  então  $x_j = 1$  na solução óptima de  $\text{REE}_\lambda$ , ou seja,  $\forall j \in N^-, x_j = 1$  na solução óptima de  $\text{REE}_\lambda$ .

Assim, como é fácil obter a solução óptima para  $\text{PCCR}_\lambda$ , pode-se tomar como estado inicial para a relaxação de espaço de estados, o estado  $(\mathcal{A}_0, \alpha_0, \beta_0) = \bigoplus_{j \in N^-} (\mathcal{A}_j, \alpha_j, \beta_j)$  com  $f(\mathcal{A}_0, \alpha_0, \beta_0) = v(\text{PCCR}_\lambda)$ , em vez do estado  $(0, m+1, 0)$ .

Uma vez que o número de estados gerados aumenta exponencialmente com o número de linhas do problema e atendendo à relação existente entre a solução óptima da relaxação lagrangeana e a relaxação de espaço de estados, é possível obter um processo globalmente

melhor para a resolução do PCC, combinando as duas relaxações e que pode, por exemplo, ser descrito do seguinte modo [Paixão & Christofides, 1986] ;

$$\begin{aligned} \text{se } \Delta_0 \geq \frac{2}{3}m & \quad \text{resolve-se } REE_\lambda, \text{ a partir do estado inicial } (\Delta_0, \alpha_0, \beta_0) = \\ & \quad \bigoplus_{j \in N^-} (\Delta_j, \alpha_j, \beta_j) \text{ e considerando apenas as colunas } j \text{ com } j \in N^+. \\ \text{se } \Delta_0 < \frac{2}{3}m & \quad \text{toma-se } v(PCCR_\lambda) \text{ como limite inferior para o } v(PCC). \end{aligned}$$

De facto, quanto maior é o valor limite assumido por  $\Delta_0$  menor será o número de operações necessárias para resolver  $REE_\lambda$ .

Em cada iteração do método do subgradiente em que se opta por  $v(PCCR_\lambda)$  aplicam-se as penalidades apresentadas na secção 4.1.

### 7. Determinação de uma cobertura a partir da solução do problema relaxado

A solução óptima do problema relaxado, pode não ser admissível para o PCC. Nesse caso, existem linhas que não são cobertas. No entanto, a solução pode ser completada de forma a obter-se uma cobertura.

Para cada linha ainda não coberta, selecciona-se para pertencer à cobertura, a coluna que a cobre com menor custo. Este processo é repetido até que todas as linhas estejam cobertas.

É possível que na solução admissível assim obtida, existam colunas redundantes, pelo que se reduz a solução a uma cobertura prima.

Em cada iteração do método de optimização do subgradiente aplica-se este método para determinar uma cobertura a partir da solução relaxada, tentando-se assim melhorar o valor corrente do limite superior.

### 8. Duas ordenações para as colunas do problema

Nesta secção apresentam-se dois procedimentos que tiram partido de reordenações particulares das colunas do problema no sentido de melhorar a qualidades dos limites inferiores, bem como acelerar a sua obtenção. Desenvolvem-se também dois testes que permitem ignorar algumas colunas e estados, num determinado estágio da recursão dinâmica.

#### 8.1. Ordenação das colunas considerando primeiro todas as colunas com $\alpha_j = 1$

As colunas do problema podem ser ordenadas de modo a considerar primeiro todas as colunas com  $\alpha_j = 1$ . Note-se que na solução óptima para o problema relaxado, tem-se sempre que  $\alpha = \min_{\{j: x_j = 1\}} \alpha_j = 1$ , o que significa que é obrigatório incluir na solução pelo menos uma coluna com  $\alpha_j = 1$ .

No procedimento de geração dos estados no espaço de estados relaxado e depois de se considerarem todos os estágios correspondentes a  $\alpha_j = 1$ , obriga-se a que os novos estados a gerar sejam obtidos por combinação com estes. Isto pode ser feito eliminando o estado inicial

$(0, m+1, 0)$ . Assim, garante-se em termos de problema original, que todos os estados  $S$  que venham a ser gerados contêm a linha 1.

Este melhoramento vai obviamente provocar uma grande diminuição na dimensão do espaços de estados, passando-se de  $O(m^3)$  para  $O(m^2)$ . Os estados gerados desta maneira são todos do tipo  $(\lambda, 1, \beta)$ , com  $\lambda \leq \beta$ , pois por construção  $\lambda = \min(\lambda', \beta - \alpha + 1)$ .

Recorde-se que um limite inferior para o custo de uma solução em que  $x_j = 1$  (coluna  $j$  pertence à solução) é dado por:

$$t_j = c_j + \min_{q \in T_j} f_n(q) \text{ e o subespaço } T_j = \{q \in Q: q \otimes g(M_j) = (m, 1, m)\}.$$

Para qualquer coluna  $j$  com  $\alpha_j \neq 1$ , os elementos  $q$  de  $T_j$  têm obrigatoriamente  $\alpha = 1$ , pois só assim é possível obter o estado  $(m, 1, m)$ . Então, para uma coluna  $j$  nessas condições, tem-se que os estados  $q \in T_j$  são da forma  $(\lambda, 1, \beta)$  com  $\beta = \begin{cases} m & \text{se } \beta_j \neq m \\ \text{qualquer caso contrário} \end{cases}$  e  $(m - |M_j| + \Delta_{1, \beta, \alpha_j, \beta_j}) \leq \lambda \leq \beta$ .

Pelas razões já referidas na secção 3, pode e deve-se optar por calcular um limite inferior  $\bar{t}_j = \max[f_n(m, 1, m), c_j + \min f_n(\lambda, 1, \beta)]$ , sendo  $\beta$  qualquer e com  $\lambda$  definido como acima.

Note-se que para as colunas com  $\alpha_j = 1$ , as penalidades  $\bar{t}_j$  não são válidas. Isto deve-se a que nestes casos, os elementos  $q$  de  $T_j$  não têm obrigatoriamente que ter  $\beta = 1$ , visto que  $\min(\alpha_j = 1, \alpha) = 1$ , qualquer que seja  $\alpha$ . Assim, e uma vez que os estados com  $\alpha \neq 1$  não foram considerados na geração, não se dispõe de toda a informação necessária para a determinação de  $T_j$ , pelo que não se aplicam penalidades a estas colunas.

## 8.2. Ordenação das colunas considerando linhas independentes

Sejam  $i$  e  $i'$  os índices correspondentes a duas linhas do problema, para as quais se tem  $N_i \cap N_{i'} = \emptyset$ . Então as linhas  $i$  e  $i'$  dizem-se independentes. Um conjunto de linhas  $i_0, \dots, i_t$  dizem-se independentes se forem independentes duas a duas.

Seja  $IIND_1 = \{i_0, \dots, i_t\}$ , um conjunto de linhas independentes tal que  $i_0 = 1$ .

As colunas do problema podem ser ordenadas, considerando em primeiro lugar e sequencialmente os blocos de colunas que cobrem cada uma das linhas que pertencem a  $IIND_1$  e depois, caso existam, as restantes colunas ( $\bar{N}$ ).

A nova ordenação de colunas, pode ser ilustrada da seguinte maneira:

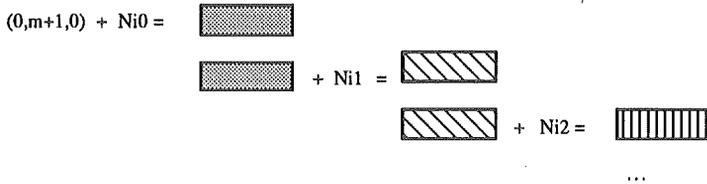


A solução óptima do problema relaxado pode não cobrir todas as linhas. No sentido de se "caminhar" na direcção da admissibilidade, obriga-se a que qualquer solução do problema relaxado inclua pelo menos uma coluna de cada  $N_i$  para todo o  $i \in IIND_1$ .

Assim, na geração do espaço de estados, o estado inicial  $(0, m+1, 0)$  é eliminado após se considerarem as colunas de  $N_{i_0}$ . Os estados então gerados serão combinados com os correspondentes às colunas de  $N_{i_1}$ . Os novos estados resultantes serão por sua vez, combinados com os correspondentes às colunas de  $N_{i_2}$ , tendo-se previamente eliminado os de

$N_{i0}$  (se fosse possível combinar  $N_{i0}$  com  $N_{i2}$  directamente sem incluir colunas de  $N_{i1}$  não se podia garantir que a linha  $i1$  fosse coberta) e assim sucessivamente.

Esquemáticamente :



Note-se que o novo valor óptimo obtido ainda é um limite inferior para  $v(\text{PCC})$ , porque o novo problema ainda é uma relaxação de PCC. Para verificar este facto, considere-se a formulação (5.1) - (5.5) juntamente com as restrições (1.2) e dupliquem-se estas restrições para  $\forall i \in \text{IIND}_1$ , ou seja, acrescentem-se a (5.1) - (5.5) e (1.2) as seguintes restrições:

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad i \in \text{IIND}_1 \quad (8.1)$$

Relaxando as restrições (1.2), obtem-se um problema muito semelhante a  $\text{REE}_\lambda$ , mas agora com as restrições adicionais (8.1).

Seguindo a ideia mencionada na secção 6, também é possível, neste caso, iniciar a geração dos estados a partir do estado  $(\mathcal{A}_0, \alpha_0, \beta_0) = \oplus_{j \in N^-} (\mathcal{A}_j, \alpha_j, \beta_j)$  com  $f(\mathcal{A}_0, \alpha_0, \beta_0) = v(\text{LPCC}_\lambda)$  e  $N^- = \{j \in N : \bar{c}_j < 0\}$ , visto que estas colunas estarão de certeza na solução óptima do problema relaxado em estudo. Assim e como se pretende incluir na solução do problema relaxado, pelo menos uma coluna de cada  $N_i$ , ( $i \in \text{IIND}_1$ ), sempre que se encontra um  $i \in \text{IIND}_1$  tal que  $N_i \cap N^- = \emptyset$ , obriga-se a inclusão de uma coluna pertencente a  $N_i$  na solução final. Isto é feito de modo análogo ao ilustrado no esquema anterior.

Note-se que a aproximação  $\bar{t}_j$  para  $t_j$  referida na secção 3, não é válida neste caso, pela mesma razão que não é válida para colunas com  $\alpha_j = 1$ . De facto, considerando a nova ordenação proposta,  $\min f_n(\mathcal{A}, \alpha, \beta)$  para todo  $\alpha, \beta$  e  $\mathcal{A} = m - |M_j|, \dots, m$  não é um limite inferior válido para o problema em que se pretendem "cobrir"  $m - |M_j|$  das  $m$  linhas do problema original, porque se fixaram, neste ordenação, as linhas pertencentes a  $\text{IIND}_1$ .

### 9. Resultados Computacionais

As duas ordenações descritas nas secções anteriores foram computacionalmente testadas num PC com processador 80286, tendo-se utilizado o compilador Pro Fortran-77 (versão iid 1.1). Assim, considerem-se os dois códigos seguintes:

PROG1 - versão em que se ordenam as colunas por ordem crescente de  $\alpha_j$ .

PROG2 - versão em que se ordenam as colunas considerando o subconjunto de linhas  $\text{IIND}_1$ .

Em ambos os programas foram considerados: o critério de dominância de estados apresentado na secção 3, a combinação da relaxação de espaço de estados com a relaxação lagrangeana descrita na secção 6, as penalidades referidas na secção 4.1 e a heurística descrita na secção 7. No programa PROG1 foram ainda introduzidas as penalidades apresentadas na secção 8.

Sempre que não se dispõe de um conjunto inicial de multiplicadores para iniciar o método do subgradiente, considera-se como conjunto inicial o conjunto de multiplicadores nulos e resolve-se obrigatoriamente a relaxação de espaço de estados. Nas iterações seguintes é aplicado o teste da secção 6 de forma a escolher qual das relaxações se resolve.

Ambas as versões foram testadas para dados reais, obtidos na CARRIS (exemplos 1 a 8) e na Rodoviária Nacional (RN) (exemplos 9 a 19). Estes últimos referem-se às zonas de Queluz de Baixo, Torres Vedras, Carcavelos, e Figueira da Foz.

Na tabela 1, apresentam-se para cada exemplo, o número de linhas (|M|), o número de colunas (|N|), o número de '1's na matriz das restrições (NJI), a densidade do problema ( $D = \frac{NJI}{|N|*|M|} * 100$ ) e um limite superior conhecido para o mesmo (LS). Os valores assinalados com um asterisco indicam valores óptimos.

Exemplo	M	N	NJI	D	LS
1	91	1223	5356	4.8	60275
2	76	678	2404	4.7	61244
3	72	739	4171	7.8	45726
4	103	580	3009	5.0	60002
5	78	686	3521	6.6	53673
6	80	1710	8713	6.4	57137
7	76	611	2219	4.8	61305
8	38	67	420	16.5	17992
9	27	328	1284	14.5	21784
10	24	246	907	15.4	21112
11	31	520	2239	13.9	21286
12	24	293	1038	14.8	20816*
13	28	400	1590	14.2	21622
14	13	43	99	17.7	21274
15	25	282	1069	15.2	21286
16	30	227	470	6.9	65527
17	46	785	2429	6.7	61508
18	58	1480	5765	6.7	62944
19	47	810	2560	6.7	62323

Tabela 1. Características das ocorrências testadas

Na tabela 2, apresentam-se, para cada versão ( $j = 1, 2$ ), o valor do limite inferior (LIPG<sub>j</sub>), o tempo de CPU medido em segundos (CPG<sub>j</sub>) e o número de colunas eliminadas (E<sub>j</sub>), correspondentes a 100 iterações do método do subgradiente. Convém recordar que no PROG2 apenas se eliminam colunas quando se resolve a relaxação lagrangeana já que nesta versão, não se utilizam as penalidades derivadas a partir da relaxação de espaço de estados.

Exemplo	LIPG1	LIPG2	CPG1	CPG2	E1	E2
1	59389	59001	1086	16042	186	2
2	60056	59606	3394	7155	1	0
3	44646	44724	656	2552	0	0
4	57556	57633	1428	7027	0	0
5	51039	51022	438	842	0	0
6	54442	54611	955	2487	0	0
7	59796	59664	2637	7007	0	0
8	17896*	17896*	14	15	34	43
9	20376	20266	68	91	61	42
10	20082	20085	52	61	81	69
11	20014	19994	90	155	260	126
12	19884	19630	64	85	95	65
13	20147	20001	124	174	74	189
14	20453	20427	6	6	1	0
15	20060	19865	71	92	74	30
16	61979	62346	337	136	0	0
17	58796	58596	1225	1382	28	5
18	56770	56737	1756	1996	0	0
19	59493	59361	4312	1647	10	0

Tabela 2. Limites inferiores obtidos com multiplicadores iniciais nulos (100 iterações do subgradiente).  
PW300, DOS 3.10, compilador Pro Fortran (processador 80286)

Nos exemplos 11 e 14 foi possível, por aplicação da heurística descrita na secção 7, melhorar o limite superior inicial respectivamente de 21286 para 20950 e de 21274 para 21112. No primeiro exemplo o melhoramento ocorreu na execução de PROG1 enquanto no segundo ocorreu na execução de PROG2.

Nos casos em que houve eliminação de colunas e/ou melhoramento do limite superior tem interesse repetir a execução, considerando o problema reduzido ou o novo problema mas com o novo limite superior. Este processo pode ser repetido iterativamente. Assim, para os exemplos nas condições mencionadas no parágrafo anterior, foram realizadas execuções adicionais até um máximo de quatro. As colunas indicadas por  $LSPG_j$ , indicam se o limite superior disponível foi melhorado. Considerando os resultados obtidos na tabela 2, apresentam-se na tabela 3, os resultados para as 3 restantes execuções.

Ex	LSPG1	LSPG2	LIPG1	LIPG2	CPG1	CPG2	E1	E2
1	-	-	59768	59462	169	188	375	164
	-	-	59823	59666	176	169	43	288
	-	-	59857	59774	151	163	31	89
2	-	-	60271	60324	170	97	35	11
	-	-	60317	60334	145	93	32	1
	-	-	60317	60339	144	114	1	1
9	-	-	20537	20417	26	29	27	34
	-	-	20537	20459	24	27	1	7
	-	-	20537	20481	24	30	0	3
10	-	-	20424	20280	16	19	28	34
	20788	-	20493	20375	14	16	28	1
	-	-	20555	20433	12	16	2	6
11	20788	-	20336	20305	28	44	92	132
	-	-	20430	20404	21	32	0	19
	-	-	-	20457	-	29	-	12
12	-	-	20203	20018	23	24	24	35
	-	-	20328	20148	19	21	7	12
	-	-	20417	20242	19	19	3	10
13	-	20788	20382	20189	37	24	53	68
	-	-	20345	20415	33	17	9	15
	21112	-	20397	20474	28	17	94	5
14	-	-	20645	-	5	-	0	-
15	-	-	20292	20149	23	21	19	34
	20788	-	20373	20248	18	20	68	16
	-	-	20440	20308	16	21	2	2
17	-	-	60175	59402	93	120	154	68
	-	-	60665	59862	87	108	17	75
	-	-	60884	60255	68	99	20	38
19	-	-	60105	-	121	-	65	-
	-	-	60306	-	102	-	36	-
	-	-	60463	-	123	-	34	-

Tabela 3. Limites inferiores obtidos nas execuções seguintes, para os exemplos da tabela 2 em que foi possível eliminar colunas e ou melhorar o limite superior (100 iterações do método do subgradiente).

PW300, DOS 3.10, compilador Pro Fortran (processador 80286).

Em seguida, na tabela 4, apresenta-se os resultados para os mesmos exemplos, mas em que o conjunto inicial de multiplicadores é obtido após a execução prévia de algumas iterações do método do subgradiente para a relaxação lagrangeana. Tal como anteriormente, caso o limite superior seja actualizado e/ou alguma coluna eliminada, repete-se a execução do programa até um máximo de 4 execuções.

Ex	LSPG1	LSPG2	LIPG1	LIPG2	CPG1	CPG2	E1	E2
1	-	-	59536	59567	154	178	329	321
	-	-	59728	59749	180	163	178	185
	-	-	59816	59805	194	146	90	74
	-	-	59835	59839	222	233	14	27
2	-	60810	60315	60315	181	104	44	294
	-	-	60315	60315	178	104	0	294
	-	-	-	60324	-	97	-	11
	-	-	-	60339	-	114	-	1
3	-	-	45451	45451	134	137	0	0
4	-	-	59683	59683	189	121	0	0
5	-	-	53500	53500	143	113	0	0
6	-	-	56771	56774	301	281	21	10
	-	-	56777	56777	493	295	6	12
	-	-	56777	56780	334	246	12	20
	-	-	56777	56780	330	232	0	3
7	-	-	60392	60392	132	117	0	0
8	-	-	17896*	17896*	11	16	46	43
9	20788	-	19151	19338	35	37	53	8
	-	-	19690	19874	28	38	60	19
	-	-	19887	20146	26	31	46	25
	-	-	20064	20292	19	28	40	22
10	20788	-	19232	19351	26	27	45	28
	-	-	19629	19703	21	24	20	14
	-	-	19852	19930	20	24	30	21
	-	-	20039	20063	16	21	28	17
11	-	21112	19212	18975	57	55	47	18
	21112	-	19564	19676	50	55	59	18
	20788	20950	19781	19930	44	42	140	137
	-	-	20002	20049	33	32	60	42
12	-	-	19153	19144	30	29	31	35
	-	-	19551	19526	29	26	23	12
	-	-	19783	19740	25	27	33	30
	-	-	19934	19872	23	25	13	16
13	-	-	19283	19295	41	42	15	111
	-	-	19641	19610	43	33	22	38
	21112	-	19912	19805	40	30	101	42
	20950	-	20055	19979	28	24	63	42
14	-	-	20950	20950	6	5	3	3
	-	-	20950	20950	6	4	0	0
15	-	-	19207	19170	31	31	15	13
	-	-	19604	19604	28	31	10	11
	-	-	19883	19862	28	28	20	16
	-	-	20086	20040	28	27	26	23
16	-	-	-	-	-	-	-	-
17	-	-	61206	61241	108	100	319	332
	-	-	61405	61341	54	53	109	42
	-	-	61439	61366	68	78	13	36
	-	-	61451	61372	51	101	8	5
18	-	-	55678	55380	174	176	0	0
19	-	-	60535	60632	93	83	199	199
	-	-	60767	60757	67	79	1	0
	-	-	60810	-	66	-	0	-

Tabela 4. Limites inferiores obtidos tomando como multiplicadores iniciais os fornecidos pela relaxação lagrangeana (100 iterações do método do subgradiente).  
PW300, DOS 3.10, compilador Pro Fortran (processador 80286).

Na tabela 5, apresentam-se para cada um dos exemplos, os melhores limites superior (LS) e inferior (LI) obtidos bem como o desvio correspondente ( $\text{Gap} = 100 \cdot (\text{LS} - \text{LI}) / \text{LI}$ ).

Ex	LS	LI	Gap
1	60275	59857	0.70
2	60810	60339	0.78
3	45726	45451	0.60
4	60002	59683	0.53
5	53673	53500	0.32
6	57137	56780	0.62
7	61305	60392	1.51
8	17896	17896	0.00
9	20788	20537	1.22
10	20788	20555	1.13
11	20788	20457	1.61
12	20816	20417	1.95
13	20788	20474	1.53
14	21112	20950	0.77
15	20788	20440	1.70
16	65527	62346	5.10
17	61508	61451	0.09
18	62944	56770	10.87
19	62323	60810	2.49

Tabela 5. Melhores limite superior, limite inferior e desvio obtidos para cada um dos exemplos.

Quando se utilizam multiplicadores iniciais nulos, o método 1 (PROG1) revela-se mais eficiente, quer em termos da qualidade dos limites inferiores obtidos, quer em termos do tempo de CPU necessário à sua obtenção e, ainda, em termos do número de colunas que é possível eliminar. A discrepância observada entre os tempos de CPU obtidos na primeira execução e seguintes deve-se ao facto de naquela se resolver obrigatoriamente a relaxação de espaço de estados na primeira iteração do método do subgradiente.

Quando se dispõe de um conjunto inicial de multiplicadores, obtidos após algumas iterações do método do subgradiente para a relaxação lagrangeana, o comportamento dos dois métodos é semelhante no que diz respeito à qualidade dos limites obtidos e aos tempos de CPU observados. Quanto ao número de colunas eliminadas, continua a observar-se uma maior eficácia do método 1, o que é explicado tendo em conta que no método 2 apenas se aplicam penalidades quando se resolve a relaxação lagrangeana. No entanto, a diferença observada no número de colunas eliminadas pelos dois programas não é, neste caso, tão acentuada.

Observe-se ainda que a qualidade dos desvios obtidos depende naturalmente da qualidade do limite superior. Os limites superiores iniciais foram obtidos por heurísticas do tipo greedy, e crê-se que a sua qualidade não é "tão boa" quanto se desejaria, tendo sido possível nalguns casos melhorar este limite superior, partindo da solução do problema relaxado.

À excepção de três instâncias testadas (16,18 e 19), os valores finais obtidos para os limites, podem considerar-se de boa qualidade. De facto, há que ter em conta que se tratam de ocorrências reais e que o suporte informático utilizado é claramente de 'baixa gama'.

Relativamente aos problema teste identificados pelos números 16,18 e 19, é de crer que o menor sucesso nos resultados seja devido ao limite superior. Na verdade, estes problemas teste são muito semelhantes ao problema 17, na medida em que dizem respeito à mesma zona e apenas diferem na partição dos serviços de viaturas em peças de trabalho. E, para este caso, obtiveram-se valores bastante bons.

#### 10. Reduções num qualquer estágio $k$ do Método 1

Nesta secção apresentam-se reduções que podem ser incorporadas em cada estágio do processo de geração de estados. As reduções vão ser apresentadas no contexto do método 1, ou seja, as colunas são ordenadas por valor crescente de  $\alpha$ .

Apesar de todas as reduções na dimensão dos problemas que foram conseguidas devido às ordenações, às penalidades e às dominâncias introduzidas, o processo de geração dos estados ainda é consideravelmente demorado para problemas de maior dimensão.

No processo de geração dos estados seria útil poder dizer à priori ou mediante comparações simples se determinada coluna é, ou não, importante para a geração. Ou por outras palavras, se todos os estados gerados em determinado estágio seriam ou não todos dominados.

Assim, considere-se a ordenação de colunas descrita na secção 8.1 e inicie-se a geração dos estados, considerando um subconjunto de colunas inicial (por exemplo, as  $(k-1)$  primeiras colunas). Apenas se exige que essas  $(k-1)$  colunas "contenham" o estado  $(m,1,m)$ . Se tal não acontecer, vão-se considerando novas colunas até tal acontecer.

Seja agora o valor MAXC tal que

$$\text{MAXC} = f_{k-1}(m,1,m) - \min_{\forall(\delta,1,\beta)} f_{k-1}(\delta,1,\beta)$$

Este valor pode ser considerado como o "maior acréscimo no valor da função custo, que se está disposto a pagar para obter uma solução tão boa como a solução corrente".

O resultado seguinte permite estabelecer um critério para a exclusão de uma determinada coluna no processo de geração dos estados não dominados.

##### Propriedade 10.1

Considere-se um qualquer estágio  $k$  no processo de geração de estados.

Se o custo da coluna  $k$ ,  $c_k$ , for tal que  $c_k \geq \text{MAXC}$  então todos os estados que resultam da combinação dessa coluna com um estado previamente gerado serão dominados pelo estado  $(m,1,m)$ .

O resultado anterior conduz então ao seguinte teste de eliminação temporária de colunas:

**Teste 1** - Se num determinado estágio  $k$ , se tem que  $c_k \geq \text{MAXC}$  então a coluna  $k$  pode ser ignorada no processo de geração de estados.

Este teste pode e deve ser aplicado em cada iteração do método do subgradiente. Mas note-se que o facto se concluir que se pode ignorar uma coluna numa dada iteração do método do subgradiente não significa que essa coluna não venha a ser necessária no processo de geração

relativo a uma iteração posterior do método do subgradiente. Como tal, a informação produzida pelo teste 1 apenas pode ser utilizada na iteração corrente desse método.

Pelo resultado anterior conclui-se que caso se tenha  $c_k < \text{MAXC}$ , podem existir estados que se obtêm como combinação da coluna  $k$  com qualquer outro e que podem vir a ser necessários para a solução final. Isto é, vai ser necessário combinar a coluna  $k$  com alguns dos estados previamente gerados. Mesmo assim, é possível identificar alguns desses estados que não vão ser necessários na combinação com esta coluna, já que os estados resultantes dessas combinações levam necessariamente a estados dominados.

Para tal, defina-se VALOR tal que

$$\text{VALOR} = f_{k-1}(m,1,m) - c_k$$

em que  $f_{k-1}(m,1,m)$  denota o custo corrente do estado  $(m,1,m)$ .

O resultado seguinte mostra como identificar estados nessas condições.

### Propriedade 10.2

Considere-se um qualquer estágio  $k$  no processo de geração de estados e seja  $(\mathcal{A}, 1, \beta)$  um estado já obtido pelo processo de geração. Se  $f_{k-1}(\mathcal{A}, 1, \beta) \geq \text{VALOR}$  então o estado resultante da combinação de  $(\mathcal{A}, 1, \beta)$  com a coluna  $k$  é dominado pelo estado  $(m,1,m)$ .

O resultado anterior leva a concluir que no processo de geração, apenas é necessário combinar a coluna  $k$  com estados previamente gerados que tenham custo inferior a VALOR. Caso contrário, os estados obtidos serão dominados pelo estado  $(m,1,m)$ . Isto leva a

**Teste 2** - Se num determinado estágio  $k$ , se tem que  $f_{k-1}(\mathcal{A}, 1, \beta) \geq \text{VALOR}$  para um determinado estado  $(\mathcal{A}, 1, \beta)$  previamente gerado, então o estado resultante da combinação de  $(\mathcal{A}, 1, \beta)$  com a coluna  $k$  pode ser ignorado no processo de geração.

De modo análogo ao teste 1, a informação produzida por este teste apenas pode ser utilizada numa iterada do método do subgradiente.

## **10.2. Resultados computacionais**

A eficiência dos testes 1 e 2, deduzidos na secção anterior, foi testada para os dados reais, apresentados na secção 9. Neste sentido desenvolveu-se um código, PROG3, muito semelhante a PROG1, acrescentado dos testes já referidos.

Nas tabelas seguintes apresentam-se os tempos de CPU (segundos) obtidos com este programa, e os já apresentados para o PROG1, no sentido de facilitar a sua comparação. Note-se que os limites inferiores obtidos são obviamente idênticos.

Os resultados apresentados na tabela 6, foram obtidos partindo de um conjunto de multiplicadores nulos.

Exemplo	CPG1	CPG3	Exemplo	CPG1	CPG3
1	1086	1148	2	3394	3483
	169	137		170	124
	176	131		145	100
	209	151		144	101
3	656	694	4	1428	1501
5	438	477	6	955	1070
7	2637	2719	8	14	15
9	68	68	10	52	53
	26	25		16	16
	24	23		14	14
	24	24		12	12
11	90	91	12	64	64
	28	28		23	21
	21	21		19	19
	-	-		19	18
13	124	126	14	6	6
	37	34		5	5
	33	31		-	-
	28	26		-	-
15	71	72	16	337	342
	22	21		-	-
	18	18		-	-
	16	15		-	-
17	1225	1241	18	1756	1790
	93	76		-	-
	87	66		-	-
	69	58		-	-
19	1790	1822			
	121	94			
	102	83			
	123	94			

Tabela 6. Tempos de cpu obtidos com multiplicadores iniciais iguais a zero (100 iterações do subgradiente).

PW300, DOS 3.10, compilador Pro Fortran (processador 80286)

Tendo em conta os valores indicados nas tabelas 6 verifica-se que o PROG1 é mais rápido quando não se dispõe à partida de um conjunto de multiplicadores (primeira execução de cada exemplo referido na tabela 5). Quando se dispõe de um conjunto de multiplicadores, como é o caso das restantes execuções, inverte-se a situação sendo o PROG3 mais rápido. Este facto é comprovado em [Paias, 1991], onde se fez um estudo comparativo dos tempos de cpu dos dois programas considerando os mesmos exemplos mas tomando como multiplicadores iniciais, os obtidos após a execução prévia de algumas iterações do método do subgradiente para a relaxação lagrangeana, à semelhança do que foi apresentado na tabela 4.

Estas observações sugerem que numa primeira abordagem se use preferencialmente o PROG1 e que nas sucessivas execuções se utilize o PROG3.

**11. Estudo computacional da eficiência das penalidades aproximadas  $\tilde{t}_j$**

Uma das razões que levou à determinação de penalidades aproximadas nas secções 3. e 8., em detrimento das penalidades exactas  $t_j$ , foi o facto de a pesquisa de um determinado estado ser bastante demorada. No entanto, teria todo o interesse comparar a eficiência relativa dos dois tipos de penalidades, ou seja, saber se pelo facto de se usarem penalidades mais fracas se eliminam menos colunas.

Na implementação efectuada para o método 1, PROG1, os estados gerados são ordenados por ordem crescente de cardinal. Recorde-se que todos os estados obtidos no processo de geração de PROG1 são da forma  $(\mathcal{A}, 1, \beta)$ . Assim dados dois estados  $(\mathcal{A}_1, 1, \beta)$  e  $(\mathcal{A}_2, 1, \beta)$ , se  $\mathcal{A}_1 \leq \mathcal{A}_2$  então na lista que representa o espaço de estados na implementação efectuada, o estado  $(\mathcal{A}_1, 1, \beta)$  aparece antes do estado  $(\mathcal{A}_2, 1, \beta)$ .

Neste tipo de implementação, a procura de um determinado estado é efectuada percorrendo, para o respectivo cardinal, a lista que representa o espaço de estados.

O interesse em conhecer a eficiência das penalidades utilizadas, levou ao desenvolvimento de uma representação alternativa para o espaço de estados, onde a pesquisa de um determinado estado fosse efectuada imediatamente. Uma forma de o fazer, consiste em considerar uma lista com dimensão igual ao número máximo de estados que é possível gerar, correspondendo cada posição da lista a um determinado estado  $(\mathcal{A}, 1, \beta)$ . Assim, todos os estados potenciais são apresentados por ordem crescente de  $\beta$  e, por sua vez, para cada  $\beta$  são apresentados por ordem crescente de  $\mathcal{A}$ . Não é necessário referenciar o índice da primeira linha, uma vez que este é fixo igual a 1.

A lista considerada pode ser ilustrada do modo seguinte:

$\beta = 1$	$\beta = 2$	$\beta = 3$	$\dots$	$\beta = m$
$(1, 1, 1)$	$(2, 1, 2)$	$(2, 1, 3)$	$(3, 1, 3)$	$(2, 1, m)$
			$\dots$	$(3, 1, m)$
				$\dots$
				$(m-2, 1, m)$
				$(m-1, 1, m)$
				$(m, 1, m)$

A dimensão da lista é dada por  $(m - 1) * (m - 2) / 2 + m$ , correspondendo ao estado  $(\mathcal{A}, 1, \beta)$  a posição  $(\beta - 1) * (\beta - 2) / 2 + \mathcal{A}$ .

Obviamente que numa representação deste tipo se dispense muita memória, até porque como já foi observado repetidas vezes, a dimensão do espaço de estados é ainda bastante elevada, mesmo considerando apenas os estados com  $\alpha = 1$ .

Utilizando esta representação para o espaço de estados, a determinação das penalidades exactas  $t_j = c_j + \min_{q \in T_j} f_n(q)$ , é simplificada uma vez que o subespaço  $T_j = \{q \in Q: q \oplus g(M_j) = (m, 1, m)\}$  é facilmente identificável.

Recorde-se que na secção 8., se caracterizaram os estados  $q = (\mathcal{A}, 1, \beta)$  pertencentes a  $T_j$ , como sendo aqueles que verificam as seguintes condições:

$$m - |M_j| + \Delta_{1, \beta, \alpha_j, \beta_j} \leq \mathcal{A} \leq \beta \text{ e } \beta = \begin{cases} m & \text{se } \beta_j \neq m \\ \text{qualquer caso contrário} \end{cases}$$

Assim,

se  $\beta_j \neq m$  então vem que  $\beta = m$  e  $m - |M_j| + \delta_{\alpha_j 1} \leq \mathcal{A} \leq m$

Neste caso, para o cálculo de  $t_j$ , não vale a pena determinar todo o subespaço  $T_j$ , uma vez que o primeiro estado a encontrar, ie o de menor cardinal, será aquele a que corresponde um menor custo. Se assim não fosse, este estado seria dominado por outro e teria sido removido. Resumindo, o primeiro estado a encontrar,  $q^*$ , com  $\beta = m$  é tal que  $q^* = \min_{q \in T_j} f_n(q)$ .

se  $\beta_j = m$ , interessam os estados  $m - |M_j| + \delta_{\alpha_j 1} \leq \mathcal{A} \leq \beta$  para os valores de  $\beta$  dados por  $m - |M_j| + \delta_{\alpha_j 1} \leq \beta \leq (\alpha_j - 1)$  e  $(\alpha_j + 1) \leq \beta \leq (m-1)$ . Para  $\beta = \alpha_j$  interessam os estados  $m - |M_j| + 1 \leq \mathcal{A} \leq \beta$  e para  $\beta = m$  interessam os estados  $m - |M_j| + \delta_{\alpha_j 1} + 1 \leq \mathcal{A} \leq \beta$ . Note-se que neste caso ( $\beta=m$ ), o estado com menor cardinal é o estado ao qual corresponde um menor custo, caso contrário teria sido dominado.

Como já foi referido, ao estado  $(\mathcal{A}, 1, \beta)$  corresponde a posição  $(\beta - 1) * (\beta - 2) / 2 + \mathcal{A}$  da lista que representa o espaço de estados. Para cada  $j$ , mostrou-se acima como identificar os estados pertencentes ao correspondente conjunto  $T_j$ . Com a representação utilizada é fácil de identificar um qualquer estado na lista que representa o espaço de estados. Isto é confirmado pelo algoritmo a seguir descrito.

#### Algoritmo (Determinação de $t_j$ para cada coluna $j$ )

##### Passo 0) inicialização

$$\text{CARDINAL} = m - |M_j| + \delta_{\alpha_j 1}$$

$$\text{CONST} = (m - 1) * (m - 2) / 2$$

$$t_j = +\infty$$

$$\text{CMIN} = +\infty$$

##### Passo 1)

Se  $\beta_j = m$  segue para o passo 2)

senão :

para  $s = \text{CARDINAL}$  até  $m$  fazer,

posição =  $\text{CONST} + s$

se  $\text{CMIN} > \text{CUSTO}(\text{posição})$  então

$\text{CMIN} = \text{CUSTO}(\text{posição})$

vai para o passo 4)

##### Passo 2)

se  $\text{CARDINAL} \leq \alpha_j$  segue para o passo 3)

senão :

para  $b = \text{CARDINAL}$  até  $(m-1)$  fazer,

$$\text{pos} = (b - 1) * (b - 2) / 2$$



## 12. Conclusões

Neste trabalho estuda-se a utilização de técnicas de relaxação de espaço de estados de um programa dinâmico associado a uma classe de problemas de cobertura. No entanto, tanto o programa dinâmico como a relaxação de espaço de estados têm um carácter geral.

Os resultados obtidos para as aplicações reais consideradas, permitem concluir que os métodos descritos podem levar à obtenção de 'gaps' contendo o óptimo, de dimensão razoável.

Os testes derivados para eliminação de colunas permitiram obter reduções significativas nas dimensões dos problemas. Essas reduções são mais acentuadas quando é possível, a partir da solução do problema relaxado, melhorar o valor corrente do limite superior.

Os testes computacionais efectuados permitem ainda concluir que a eficácia das penalidades aproximadas face às penalidades exactas, é de cerca de 100%, uma vez que na quase totalidade dos exemplos testados se eliminou igual número de colunas. Para os exemplos onde tal não se passou, a diferença observada entre o número de colunas eliminadas pelos dois tipos de penalidades foi de uma coluna no final dos 4 "runs" sucessivos.

Como investigação futura, teria interesse estabelecer regras que permitam, face à informação obtida a partir de um subconjunto de colunas, decidir à priori se cada uma das restantes colunas é ou não importante para a geração dos estados, no espaço de estados relaxado. Estas regras serão importantes para a resolução de instâncias de dimensão extremamente elevada bem como para a elaboração de um processo de geração implícita de colunas.

## 13. Bibliografia

- [1] Christofides, N., Paixão, J., Algorithms for Large Scale Set Covering Problems, Internal Report Department of Management Science Imperial College of Science Technology London, 1984.
- [2] Christofides, N., Mingozzi, A., Toth, P., State-Space Relaxation Procedures for the Computation of Bounds to Routing Problems, *Networks*, 11(1981) 145-164.
- [3] Daduna, J. and Wren, A.(eds.), *Computer-Aided Transit Scheduling*, Springer-Verlag, Berlin, 1988.
- [4] Geoffrion, A.M., Lagrangean Relaxation for Integer Programming; *Mathematical Programming Study* 2 (1974) 82-114.
- [5] Paias, A. M., Determinação dos Horários de Pessoal Tripulante: Uma abordagem baseada numa relaxação de espaço de estados; Tese de Mestrado, Universidade de Lisboa, 1991.
- [6] Paias, A.M., Sistema interactivo para geração dos horários de condutores de viaturas, *Investigação Operacional*, 8(1988) 47-54.
- [7] Paixão, J.P., Transit crew scheduling on a personal workstation (MS/DOS) in "*Operational Research '90*", H. Bradley (eds), Pergamon Press, Oxford, 1990.
- [8] Paixão, J.P., Eusébio, J.M. and Amado L., Generation of Bus Driver Schedules, Nota nº12, CEAUL, Faculdade de Ciências de Lisboa, 1985.
- [9] Paixão, J. , Christofides, N., State Space Relaxation for the Set Covering Problem, Nota nº19, CEAUL, Faculdade de Ciências de Lisboa. Lisboa/INIC, 1986.
- [10] Shapiro, J.F., A survey of Lagrangean Techniques for Discrete Optimization, *Annals of Discrete Mathematics* 5 (1979) 113-138.
- [11] Rousseau, J.-M.(eds.), *Computer Scheduling of Public Transport 2*, North-Holand, Amsterdam, 1985.
- [12] Wren, A.(eds.), *Computer Scheduling of Public Transportation: Urban Passenger Vehicle and crew scheduling*, North Holland, Amsterdam, 1981.



# DIFERENTES ABORDAGENS AO PROBLEMA DE LOCALIZAÇÃO DE SERVIÇOS SEMIOBNÓXIOS

M. Conceição Fonseca  
D.E.I.O. / C.E.A.U.L.  
Faculdade de Ciências de Lisboa  
Av. 24 de Julho, 134-5º  
1300 Lisboa

## Resumo

Apresentam-se várias abordagens possíveis para o estudo do problema de localização de serviços semiobnoxios numa rede. Designam-se de semiobnoxios os serviços que, embora sendo necessários às populações, exercem um efeito desagradável ou mesmo nocivo sobre as mesmas quando situados muito próximo. A localização de serviços semiobnoxios é considerada como um problema de localização discreta que, devido às características conflituosas dos serviços, pode ser encarado sob diversas perspectivas. Foram estudadas quatro abordagens a este problema, originando vários problemas de localização discreta que diferem nos objectivos que se pretende atingir. Dois desses problemas são problemas de localização com dois objectivos enquanto que os outros são problemas de localização com uma função objectivo e restrições adicionais de distância máxima e/ou mínima ao serviço mais próximo. Calcularam-se limites superiores e inferiores utilizando heurísticas e técnicas de relaxação. Apresentam-se resultados computacionais obtidos com exemplos gerados aleatoriamente.

## Abstract

We present some approaches for locating semiobnoxious facilities in a given network. Semiobnoxious are facilities that provide disservices as well as services. Hence, when locating semiobnoxious facilities, two conflicting objectives must be attained. We studied four approaches to this problem which led to different discrete location problems differing in the objectives to attain. Two of them are biobjective location problems while the others are location problems with one objective function and additional distance constraints. We assure that the demand points are neither too close to the nearest facility, in order to avoid the obnoxious effect, nor too far away so that they can be properly served. Lower and upper bounds were obtained for each approach making use of heuristic and relaxation methods. We report some computational results for randomly generated problems.

## Keywords:

Discrete location, Biobjective problem, Heuristic and relaxation methods.

## 1. Introdução

Equipamentos ou serviços semiobnoxios são serviços que sendo por um lado úteis e necessários às comunidades, por outro lado, devido à sua natureza, exercem um efeito desagradável ou mesmo nocivo sobre essas comunidades. São serviços semiobnoxios, por exemplo, as estações de tratamento de lixo e algumas fábricas poluentes.

O problema de localização de serviços semiobnoxios é considerado como um problema de localização discreta numa rede. Não faz sentido considerar a localização dos serviços no espaço

contínuo, visto que os locais para a sua instalação não se encontram disponíveis em qualquer ponto do plano. Geralmente os locais existentes são em pequeno número e correspondem a áreas bem delimitadas. Sendo assim, os potenciais locais de instalação destes serviços são considerados como vértices de uma rede assim como as comunidades que os utilizam.

A determinação dos locais de instalação de serviços deste tipo deve ser feita de modo a minorar o efeito obnóxico exercido sobre as comunidades e simultaneamente maximizar a acessibilidade destas aos serviços instalados que estão mais próximos.

Estes objectivos contraditórios que caracterizam o problema de localização de serviços semiobnóxios podem ser encarados sob diversas perspectivas.

Neste trabalho consideram-se quatro possíveis abordagens a este problema propondo-se e descrevendo-se métodos de resolução para cada um dos correspondentes problemas de localização. Os testes computacionais referem-se a exemplos gerados aleatoriamente, uma vez que não se dispunha de casos reais nem se encontraram na literatura exemplos ou resultados computacionais relativos a este problema. Tentou-se, dentro do possível, que os exemplos gerados traduzissem a realidade.

Na segunda secção formula-se o problema de localização de serviços semiobnóxios como um problema de localização de  $p$  serviços em que se minimiza a soma ponderada das distâncias entre as comunidades e os serviços, assegurando que toda a comunidade está a uma distância considerada razoável do serviço mais próximo. Calculam-se limites superiores utilizando várias heurísticas e determinam-se limites inferiores através de uma heurística dual e de técnicas de relaxação lagrangeana. Na terceira secção considera-se como objectivo primordial do problema em estudo a minimização do efeito obnóxico exercido pelos serviços. Determinam-se limites inferiores utilizando vários métodos heurísticos e calculam-se limites superiores usando técnicas de relaxação linear. Na quarta secção encara-se o problema de localização de serviços semiobnóxios como um problema de localização com dois objectivos. Apresentam-se duas abordagens que diferem nas funções objectivo consideradas propondo-se métodos para a sua resolução. Na quinta secção apresentam-se e analisam-se os resultados computacionais obtidos. Na sexta secção referem-se algumas conclusões.

## 2. Maximização da Acessibilidade aos Serviços

Nesta abordagem ao problema de localização de serviços ou equipamentos semiobnóxios o objectivo primordial é instalar os equipamentos de modo a que em média as comunidades estejam o mais próximo possível deles podendo assim usufruir efectivamente dos seus serviços. No entanto, a fim de minimizar o efeito obnóxico dos serviços sobre as comunidades, deve exigir-se para todas as comunidades que a distância ao serviço instalado mais próximo seja maior ou igual a determinado valor pré-estabelecido. Este problema pode então formular-se como um problema de localização discreta de  $p$  serviços em que o objectivo é minimizar a soma ponderada das distâncias entre as comunidades e os serviços com uma restrição adicional de

distância mínima ao serviço mais próximo. Em Programação Linear Inteira o problema formula-se do seguinte modo:

$$(P1) \quad \min \sum_{j \in J} \sum_{i \in I} c_{ij} x_{ij}$$

$$\text{s.a:} \quad \sum_{j \in J} x_{ij} = 1 \quad i \in I \quad (2.1)$$

$$\sum_{j \in J} y_j = p \quad (2.2)$$

$$x_{ij} \leq y_j \quad i \in I, j \in J \quad (2.3)$$

$$d_{ij} \geq D y_j \quad i \in I, j \in J \quad (2.4)$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J \quad (2.5)$$

sendo  $J = \{1, 2, \dots, m\}$  o conjunto dos potenciais locais de instalação dos serviços,  $I = \{1, 2, \dots, n\}$  o conjunto das comunidades a servir,  $a_i$  o peso associado a cada comunidade  $i \in I$ ,  $d_{ij}$  a distância entre a comunidade  $i$  e o serviço a instalar em  $j$ ,  $c_{ij} = a_i d_{ij}$  e  $D$  um parâmetro que designa o valor da distância mínima permitida entre cada localidade e o serviço que se encontra mais próximo. As variáveis de decisão são

$$y_j = \begin{cases} 1 & \text{se o serviço é instalado em } j \\ 0 & \text{caso contrário} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{se a comunidade } i \text{ é servida pelo serviço instalado em } j \\ 0 & \text{caso contrário} \end{cases}$$

As restrições (2.1) asseguram que toda a comunidade é servida por um único serviço, a equação (2.2) impõe a instalação de  $p$  serviços, as condições (2.3) garantem que só há afectação para serviços instalados e as restrições (2.5) definem o domínio de variação das variáveis do problema. Estas restrições são as que habitualmente aparecem num problema de localização de  $p$  serviços. As restrições (2.4) garantem que toda a comunidade se encontra a uma distância maior ou igual a  $D$  do serviço mais próximo.

Seja  $J' = \{j \in J : d_{ij} \geq D, \forall i \in I\}$ . O problema **P1** é equivalente ao problema cuja formulação em Programação Linear Inteira é:

$$(P1') \quad \min \sum_{j \in J'} \sum_{i \in I} c_{ij} x_{ij}$$

$$\text{s.a:} \quad \sum_{j \in J'} x_{ij} = 1 \quad i \in I \quad (2.6)$$

$$\sum_{j \in J'} y_j = p \quad (2.7)$$

$$x_{ij} \leq y_j \quad i \in I, j \in J' \quad (2.8)$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J' \quad (2.9)$$

As restrições (2.6), (2.7), (2.8) e (2.9) têm, respectivamente, o mesmo significado que as restrições (2.1), (2.2), (2.3) e (2.5) do problema **P1**.

A equivalência entre estes dois problemas é fácil de provar e encontra-se em [3]. Uma vez que estes dois problemas são equivalentes e que se torna mais fácil determinar limites

superiores e limites inferiores para o valor óptimo do problema  $P1'$ , os limites para o valor óptimo do problema  $P1$  serão aqui obtidos através do problema  $P1'$ .

Para transformar o problema  $P1$  no problema  $P1'$  basta eliminar os potenciais locais para instalação de serviços que não verificam a restrição de distância mínima a alguma comunidade. Assim, salvo indicação em contrário,  $J$  designará os locais para instalação de serviços que estão a uma distância maior ou igual a  $D$  de qualquer comunidade.

Para calcular limites superiores para o problema  $P1'$ , adaptaram-se várias heurísticas existentes para o problema da  $p$ -mediana.

Fez-se uma adaptação da heurística apresentada em [2] por *Cornuejols et al.* para o problema de localização simples. É uma heurística "greedy" que consiste em determinar, em cada iteração, qual o serviço que, ao ser instalado, conduz a um maior melhoramento no valor da função objectivo. Esse serviço é seleccionado para a solução heurística. O processo termina quando na solução já existem  $p$  elementos.

Utilizaram-se também métodos de melhoramentos por trocas locais.

Um dos métodos é uma adaptação da heurística de Teitz & Bart [9] para o problema da  $p$ -mediana. Este método consiste em trocar um elemento que não pertence à actual solução por um elemento da solução, sempre que de tal troca resulte uma diminuição do valor da função objectivo. O algoritmo termina quando já não é possível melhorar o valor da função objectivo efectuando este tipo de trocas. O outro método, que designaremos por Pesquisa Local, consiste em retirar um elemento da solução e substituí-lo, de forma admissível, por um elemento não pertencente à solução. Efectua-se a troca que corresponde ao maior melhoramento produzido na função objectivo e recomeça-se o processo com a nova solução admissível assim obtida. Se não houver melhoramento repõe-se o elemento na solução e recomeça-se o processo seleccionando outro elemento da solução. Os elementos da solução inicial são seleccionados pela mesma ordem pela qual foram inseridos na solução quando esta foi construída. Tal advém do facto de a escolha dos primeiros elementos para constituírem a solução ser feita com menos informação disponível.

Os passos dos três métodos anteriores são semelhantes aos descritos em [1] para a  $p$ -mediana.

Adaptou-se também para o problema de localização de serviços semiobnóxios uma heurística apresentada por Maranzana para o problema da  $p$ -mediana [7]. A partir de uma solução inicial admissível constituída pelos serviços  $j_1, j_2, \dots, j_p$  constrói-se uma partição de cada um dos conjuntos  $J$  e  $I$  em  $p$  subconjuntos que são, respectivamente,  $P_{j_1}, \dots, P_{j_p}$  e  $Q_{j_1}, \dots, Q_{j_p}$ . Cada conjunto  $Q_{j_i}$  ( $i = 1, 2, \dots, p$ ) é constituído pelas comunidades  $i \in I$  afectas ao serviço  $j_i$ , isto é, as comunidades que estão mais próximas de  $j_i$  do que de qualquer outro elemento da solução. Cada serviço vai pertencer ao conjunto  $P_{j_r}$  ( $r = 1, \dots, p$ ) para o qual a soma das distâncias entre esse serviço e as comunidades de  $Q_{j_r}$  ( $r = 1, \dots, p$ ) é menor. Determina-se o centro de gravidade de cada conjunto  $P_{j_r}$ , isto é, o serviço de  $P_{j_r}$  que minimiza

a soma das distâncias entre as comunidades de  $Q_{j_r}$  e um dos elementos de  $P_{j_r}$ . Trocam-se os elementos da solução actual pelos centros de gravidade. O processo termina quando todos os elementos que constituem a solução actual coincidem com os centros de gravidade determinados.

Desenvolveram-se ainda duas heurísticas que se baseiam no cálculo de poupanças. Estas heurísticas são uma adaptação, para este problema, das duas heurísticas apresentadas por Khumawala em [6] para o problema da p-mediana com restrição de distância máxima.

No método designado por Delta calcula-se para cada serviço  $j \in J$ , a poupança mínima,  $\Delta_j$ , que se obtém quando esse serviço é instalado. Em cada iteração impede-se a instalação do serviço com menor poupança, de entre os serviços ainda livres. O facto de se impedir a instalação de determinado serviço leva a alterações das poupanças dos restantes serviços livres, pelo que estas terão que ser recalculadas.

No método Omega calcula-se a poupança total, em relação aos serviços já instalados, que cada serviço originaria se fosse instalado. Instala-se, em cada iteração, o serviço a que corresponde maior poupança. Dado que a poupança total,  $\Omega_j$ , para cada serviço  $j \in J$  é calculada relativamente aos serviços já instalados, é necessário escolher qual o serviço que se instala em primeiro lugar. Selecciona-se para tal o serviço  $j \in J$  a que corresponde maior valor de  $\Delta_j$ . O processo termina quando o número de serviços instalados é p.

A descrição pormenorizada das três últimas heurísticas é apresentada em [3]

Para obter limites inferiores para o valor óptimo do problema **P1** utilizou-se um método heurístico para o dual da relaxação linear de **P1'**. A partir da solução obtida para o dual e utilizando as relações de complementaridade é fácil construir uma solução admissível para o problema primal **P1'**. Obtém-se deste modo um limite superior para o valor óptimo de **P1**. Os métodos utilizados para obter limites inferiores, através da heurística dual, e limites superiores com base na solução do dual são semelhantes aos descritos em [1].

Determinou-se ainda outro limite inferior com base numa relaxação Lagrangeana do problema **P1'** considerando a relaxação das restrições de igualdade (2.6) e (2.7). A optimização do dual Lagrangeano é feita pelo método do subgradiente [4], [5]. Embora do ponto de vista teórico a relaxação linear do problema **P1'**,  $\bar{\mathbf{P1}}$ , e esta relaxação Lagrangeana produzam limites inferiores de igual qualidade, dada a grande dimensão do problema  $\bar{\mathbf{P1}}$  para resolver em Programação Linear, optou-se por obter limites inferiores resolvendo o dual lagrangeano pelo método do subgradiente.

### 3. Minimização do Efeito Obnóxio dos Serviços

O problema de localização de serviços semiobnóxios pode também formular-se como um problema de localização discreta de serviços em que o objectivo é maximizar a soma ponderada das distâncias entre as comunidades e os serviços considerando restrições adicionais de distância máxima e mínima das comunidades ao serviço instalado mais próximo. Pretende-se

com estas restrições garantir que a distância a que cada comunidade fica do equipamento mais próximo seja tal que essa comunidade possa efectivamente utilizar os seus serviços sem estar demasiadamente exposta aos efeitos nocivos do mesmo. Nesta abordagem, o número de serviços a instalar é o estritamente necessário para garantir que a solução obtida seja admissível, visto que a instalação de mais serviços para além deste número conduz a um menor valor para a função objectivo.

O problema de localização discreta correspondente a esta abordagem pode formular-se em Programação Linear Inteira do seguinte modo:

$$(P2) \quad \max \sum_{j \in J} \sum_{i \in I} c_{ij} x_{ij}$$

$$\text{s.a:} \quad \sum_{j \in J} x_{ij} = 1 \quad i \in I \quad (3.1)$$

$$d_{ij} x_{ij} \leq d y_j \quad i \in I, j \in J \quad (3.2)$$

$$d_{ij} \geq D y_j \quad i \in I, j \in J \quad (3.3)$$

$$d_{ij} (y_j - x_{ij}) \leq d_{ik} (1 - x_{ik}) \quad i \in I; j, k \in J \quad (3.4)$$

$$j \neq k, d_{ij} < d_{ik}$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J \quad (3.5)$$

onde a notação utilizada e a definição das variáveis de decisão é a mesma da secção 2. O parâmetro  $d$  designa o valor da distância máxima permitida entre cada comunidade e o serviço instalado mais próximo.

As restrições (3.1), (3.3) e (3.5) têm um significado semelhante, respectivamente às restrições (2.1), (2.4) e (2.5). As restrições (3.2) garantem que só há afectações para serviços instalados respeitando a distância máxima, enquanto as restrições (3.4) garantem que a afectação de cada comunidade é feita ao serviço instalado mais próximo. Definindo  $J' = \{ j \in J : d_{ij} \geq D, \forall i \in I \}$  prova-se que o problema  $P2$  é equivalente ao problema  $P2'$ , cuja formulação em Programação Linear Inteira é:

$$(P2') \quad \max \sum_{j \in J'} \sum_{i \in I} c_{ij} x_{ij}$$

$$\text{s.a:} \quad \sum_{j \in J'} x_{ij} = 1 \quad i \in I \quad (3.7)$$

$$d_{ij} x_{ij} \leq d y_j \quad i \in I, j \in J' \quad (3.8)$$

$$d_{ij} (y_j - x_{ij}) \leq d_{ik} (1 - x_{ik}) \quad i \in I; j, k \in J' \quad (3.9)$$

$$j \neq k, d_{ij} < d_{ik}$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J' \quad (3.10)$$

As restrições (3.7), (3.8), (3.9) e (3.10) têm o mesmo significado que as correspondentes restrições (3.1), (3.2), (3.4) e (3.5) do problema  $P2$ .

Tal como para o problema  $P1$ , apresentado na secção anterior, também para  $P2$  se torna mais fácil usar  $P2'$  para determinar limites superiores e inferiores para o seu valor óptimo. Assim, os potenciais locais de instalação de serviços são apenas os elementos de  $J'$ .

Os limites superiores para o problema P2 foram obtidos utilizando a relaxação linear [8]. Apesar de os problemas a resolver serem de grandes dimensões, optou-se por esta técnica uma vez que a utilização de técnicas de Relaxação Lagrangeana originava problemas tão difíceis de resolver como o problema original ou conduzia a um elevado número de multiplicadores.

Para a determinação de limites inferiores para o problema P2' utilizaram-se diversos métodos heurísticos.

Para o cálculo destes limites considerou-se a matriz de custos constituída pelos elementos  $c_{ij}$  ( $i \in I$ ,  $j \in J'$ ) definidos do seguinte modo:

$$c_{ij} = \begin{cases} a_i d_{ij} & \text{se } d_{ij} \leq d \\ -M & \text{caso contrário} \end{cases}$$

onde  $M$  é um valor suficientemente grande.

Não se permite assim, a afectação de uma comunidade a um serviço que fica instalado a uma distância superior a  $d$ .

É necessário verificar se toda a comunidade é alcançável a partir de pelo menos um serviço, isto é, se existe pelo menos um serviço a uma distância menor ou igual a  $d$  dessa comunidade. Caso tal não se verifique, não é possível determinar uma solução admissível para o problema para aquele valor do parâmetro  $d$ . Sempre que possível, reduz-se o número de potenciais locais de instalação de serviços, eliminando da solução os locais inacessíveis a qualquer comunidade, isto é, os locais  $j \in J'$  tais que  $d_{ij} > d$ , para todo o  $i \in I$  e os locais acessíveis a uma única comunidade que pode ser afectada a outros locais originando um maior valor para a função objectivo. Quando existem comunidades apenas acessíveis a um único equipamento reduz-se o problema determinando o conjunto mínimo de equipamentos necessários para servir essas comunidades.

Um dos métodos utilizados para a obtenção de limites inferiores é uma heurística do tipo "greedy". Nesta heurística escolhe-se em cada iteração para fazer parte da solução o serviço que ao ser instalado produz uma menor diminuição no valor da função objectivo. O processo termina logo que se obtenha uma solução admissível para o problema.

No algoritmo 3.1 apresenta-se uma descrição dos passos deste método heurístico.

### Algoritmo 3.1

#### Passo 1. Inicialização

$t = 0$

$S = \emptyset$

$I = \{1, \dots, n\}$  comunidades a servir

$J = \{1, \dots, m\}$  potenciais locais de instalação de serviços

$F = J \setminus S$

$u_i^t = +\infty$   $i \in I$

Para toda a comunidade  $i \in I$  para a qual existe um único serviço  $k \in J$  tal que  $c_{ik}$  é finito fazer:

$t = t + 1$

$$S = S \cup \{k\}$$

$$u_i^t = \begin{cases} \min\{u_i^{t-1}, c_{ik}\}, & \text{se } c_{ik} \text{ finito} \\ u_i^{t-1} & , \text{ caso contrário} \end{cases}$$

$$F = F \setminus \{k\}$$

$$\sigma(S) = \sum_{i \in I} u_i^t$$

Se  $S = \emptyset$  calcular  $z_j = \sum_{i \in I} c_{ij} \quad \forall j \in F$

Escolher  $z_k = \max_{j \in J} z_j$

Fazer  $S = S \cup \{k\}$

$$t = t + 1$$

$$F = F \setminus \{k\}$$

Para  $i \in I$  fazer

$$u_i^t = \begin{cases} \min\{u_i^{t-1}, c_{ik}\}, & \text{e } c_{ik} \text{ finito} \\ u_i^{t-1} & , \text{ caso contrário} \end{cases}$$

$$\sigma(S) = \sum_{i \in I} u_i^t$$

### Passo 2. Teste

Se a solução obtida é admissível, isto é, se  $u_i^t$  finito  $\forall i \in I$ , então  $S$  é a aproximação heurística deste problema com valor  $\sigma(S)$ . STOP.

### Passo 3. Iteração $t$

Para cada  $j \in F$  calcular  $\rho_j(S) = \sum_{i \in I} u_i^t - \sum_{i \in I} \min_{\ell \in S \cup \{j\}} c_{i\ell}$   
 $c_{i\ell}$  finito

em que  $\min_{\substack{\ell \in S \cup \{j\} \\ c_{i\ell} \text{ finito}}} c_{i\ell} = 0$  se  $\{c_{i\ell} : c_{i\ell} \text{ finito para } \ell \in S \cup \{j\}\} = \emptyset$

Seja  $\rho_k(S) = \min_{j \in F} \rho_j(S)$

Fazer  $S = S \cup \{k\} \quad F = F \setminus \{k\}$

Para  $i \in I$  fazer

$$u_i^{t+1} = \begin{cases} \min\{u_i^t, c_{ik}\}, & \text{se } c_{ik} \text{ finito} \\ u_i^t & , \text{ caso contrário} \end{cases}$$

$$\sigma(S) = \sum_{i \in I} u_i^{t+1}$$

$$t = t + 1$$

Voltar ao Passo 2.

Desenvolveram-se dois métodos heurísticos de melhoramentos que se baseiam em trocas admissíveis entre elementos da solução corrente e elementos não pertencentes à solução. Para os aplicar é necessário dispôr de uma solução admissível inicial. Tal solução pode ser obtida escolhendo arbitrariamente elementos de  $J$  que originem uma solução admissível ou é resultado de aplicar alguma das outras heurísticas disponíveis para este problema.

Num dos métodos troca-se, em cada iteração, um elemento que não pertence à actual solução por um elemento da solução desde que a nova solução continue admissível e de tal resulte um aumento para o valor da função objectivo [3]. No outro método selecciona-se um elemento da actual solução e troca-se por um elemento que não pertence à solução sempre que de tal resulte uma solução admissível a que corresponde um melhor valor para a função objectivo. A troca efectua-se com o elemento, nas condições referidas, que produz o maior melhoramento na função objectivo. O processo termina assim que todos os elementos da solução inicial tiverem sido pesquisados e analisados quanto à possibilidade de produzirem um melhoramento no valor da função objectivo ao serem trocados por outro elemento que não pertence à solução. A escolha dos elementos da solução é feita pela mesma ordem pela qual foram inseridos na solução no caso em que a solução inicial é obtida a partir da heurística "greedy" [3].

Consideraram-se também duas heurísticas em que o critério para instalar ou não serviços em determinados locais está de acordo com a diminuição ou aumento sofrido pelo valor da função objectivo em resultado de instalar ou não o serviço nesse local.

Seja  $\Delta_j$  ( $j \in J$ ) o aumento produzido no valor da função objectivo caso não se instale um serviço no local  $j \in J$ . Em cada iteração da heurística descrita no Algoritmo 3.2 impede-se a instalação do serviço a que corresponde maior valor de  $\Delta_j$ . Em qualquer iteração, se existir uma comunidade que só pode aceder a um único serviço, esse serviço  $j \in J$  tem que ser obrigatoriamente instalado, caso contrário obter-se-ia uma solução não admissível para o problema.

O processo pára quando a solução corrente é admissível ou quando a não instalação de mais um serviço conduziria a uma solução não admissível.

Relativamente à heurística descrita no Algoritmo 3.3, defina-se  $\Omega_j$  ( $j \in J$ ) como sendo a poupança total que cada serviço originaria se fosse instalado, relativamente aos serviços já instalados. Como  $\Omega_j$  ( $j \in J$ ) é calculado relativamente aos serviços já instalados, é necessário dispôr de pelo menos um serviço instalado para iniciar o processo. Assim, calcula-se  $\Delta_j$  ( $j \in J$ ) tal como é feito no Algoritmo 3.2. Instalam-se todos os serviços para os quais existe uma comunidade que só tem acesso a esse serviço, se existir algum nestas condições. Caso contrário, instala-se em primeiro lugar o serviço a que corresponde menor valor de  $\Delta_j$ . Em cada uma das iterações seguintes, instala-se o serviço a que corresponde menor valor de  $\Omega_j$ .

O processo pára assim que a solução obtida for admissível.

Apresenta-se seguidamente uma descrição pormenorizada dos algoritmos 3.2 e 3.3.

<b>Algoritmo 3.2</b>
----------------------

**Passo 1. Inicialização**

$I = \{1, \dots, n\}$  comunidades a servir  
 $J = \{1, \dots, m\}$  potenciais locais de instalação de serviços  
 $C = \{\text{serviços fechados}\} = \emptyset$   
 $S = \{\text{serviços instalados}\} = \emptyset$   
 $F = J$

Para  $i \in I$  fazer:

$$v_i = -\infty$$

$$u_i^1 = \min_{j \in F} c_{ij} = c_{ij_1^1} \quad \text{para } c_{ij} \text{ finito}$$

$$u_i^2 = \begin{cases} \min_{\substack{j \in F \\ j \neq j_1^1}} c_{ij} = c_{ij_1^2} & , \text{ se } c_{ij} \text{ finito} \\ -\infty & , \text{ caso contrário} \end{cases}$$

Construir

$$J_r^1 = \{i \in I : j_i^1 = r\} \quad r \in F$$

$$J_r^2 = \{i \in I : j_i^2 = r\} \quad r \in F$$

Calcular

$$\Delta_j = \sum_{i \in J_j^1} (u_i^2 - u_i^1) \quad j \in F$$

$$\text{Fazer } \begin{cases} S = \{j \in F : \exists i \in J_j^1 \text{ tal que } u_i^2 = -\infty\} \\ F = F \setminus S \end{cases}$$

$$v_i = \begin{cases} \min_{j \in S} c_{ij} & , \text{ se } c_{ij} \text{ finito} \\ -\infty & , \text{ caso contrário} \end{cases} \quad i \in I$$

**Passo 2. Teste**

Se a solução  $S$  obtida é admissível, isto é, se  $v_i$  é finito para todo o  $i \in I$ , então  $S$  é a aproximação heurística deste problema com valor  $\sigma(S) =$

$$\sum_{i \in I} v_i \cdot \text{STOP.}$$

**Passo 3.** Seja  $k$  tal que  $\Delta_k = \max_{j \in F} \Delta_j$ 

$$\text{Fazer } \begin{cases} C = C \cup \{k\} \\ F = F \setminus \{k\} \end{cases}$$

Para  $i \in J_k^1$  fazer

$$u_i^1 = u_i^2 \quad j_i^1 = j_i^2$$

$$u_i^2 = \begin{cases} \min_{\substack{j \in F \cup S \\ j \neq j_i^1}} c_{ij} = c_{ij_1^2} & , \text{ se } c_{ij} \text{ finito} \\ -\infty & , \text{ caso contrário} \end{cases}$$

Para  $i \in J_k^2$  fazer

$$u_i^2 = \begin{cases} \min_{\substack{j \in F \cup S \\ j \neq j_i^1}} c_{ij} = c_{ij_i^2}, & \text{se } c_{ij} \text{ finito} \\ -\infty & , \text{ caso contrário} \end{cases}$$

Actualizar  $J_r^1$  e  $J_r^2$  para cada  $r \in F$

$$\text{Calcular } \Delta_j = \sum_{i \in J_j^1} (u_i^2 - u_i^1) \quad j \in F$$

Se para algum  $j \in F$  existe  $i \in J_j^1$  tal que  $u_i^2 = -\infty$

$$\text{então } \begin{aligned} S &= S \cup \{j\} \\ F &= F \setminus \{j\} \end{aligned}$$

$$v_i = \begin{cases} \min_{j \in S} c_{ij}, & \text{se } c_{ij} \text{ finito} \\ & i \in I \\ -\infty & , \text{ caso contrário} \end{cases}$$

Voltar ao Passo 2.

### Algoritmo 3.3

#### Passo 1. Inicialização

$I = \{1, \dots, n\}$  comunidades a servir

$J = \{1, \dots, m\}$  potenciais locais de instalação de serviços

$S = \emptyset$

$F = J$

$t = 1$

$$\Delta_j = \sum_{i \in J_j^1} (c_{ij_i^2} - c_{ij_i^1}) \quad j \in J$$

onde  $J_j^1 = \{i \in I : c_{ij} = \min_{k \in J} c_{ik}\}$   
 $c_{ik}$  finito

com  $c_{ij_i^1} = \min_{j \in F} c_{ij}$  para  $c_{ij}$  finito

$$c_{ij_i^2} = \begin{cases} \min_{\substack{j \in F \\ j \neq j_i^1}} c_{ij}, & \text{se } c_{ij} \text{ finito} \\ -M & , \text{ caso contrário} \end{cases}$$

Fazer  $S = \{j \in J : \exists i \in J_j^1 \text{ tal que } c_{ij_i^2} = -M\}$   
 $F = J \setminus S$

Se  $S = \emptyset$  seja  $\Delta_k = \min_{j \in F} \Delta_j$

Fazer  $S = \{k\}$   
 $F = F \setminus \{k\}$

$$\text{Fazer } u_i^1 = \begin{cases} \min_{j \in S} c_{ij}, & \text{se } c_{ij} \text{ finito} \\ & i \in I \\ +\infty & , \text{ caso contrário} \end{cases}$$

**Passo 2. Teste**

Se a solução  $S$  obtida é admissível, isto é, se  $u_i^t$  é finito para todo o  $i \in I$ , então  $S$  é a aproximação heurística deste problema com valor  $\sigma(S) =$

$$\sum_{i \in I} u_i^t . \text{ STOP.}$$

**Passo 3. Iteração  $t$** 

Para cada  $j \in F$  calcular  $\Omega_j = \sum_{i \in I} u_i^t - \sum_{\substack{\ell \in S \cup \{j\} \\ c_{i,\ell} \text{ finito}}} \min c_{i,\ell}$

em que  $\min_{\substack{\ell \in S \cup \{j\} \\ c_{i,\ell} \text{ finito}}} c_{i,\ell} = 0$  se  $\{c_{i,\ell} : c_{i,\ell} \text{ finito para } \ell \in S \cup \{j\}\} = \emptyset$

Seja  $\Omega_k = \min_{j \in F} \Omega_j$

Fazer  $S = S \cup \{k\}$   
 $F = F \setminus \{k\}$

$$u_i^{t+1} = \begin{cases} \min \{ u_i^t, c_{ik} \}, & \text{se } c_{ik} \text{ finito} \\ u_i^t & , \text{ caso contrário} \end{cases} \quad i \in I$$

$t = t + 1$

Voltar ao **Passo 2.**

**4. Abordagens com dois Objectivos**

A localização de serviços semiobnóxios pode ser encarada como um problema de localização com duas funções objectivo. Pretende-se maximizar a acessibilidade das comunidades aos serviços e, simultaneamente, minimizar o efeito obnócio exercido pelos serviços. Assim, a forma como tratamos cada um dos objectivos conduz a modos diferentes de encarar o problema.

Uma das formas será considerar os dois objectivos como:

- (i) minimizar a soma ponderada das distâncias entre as comunidades e os serviços;
- (ii) maximizar a distância mínima entre as comunidades e o serviço mais próximo.

Pretende-se, deste modo, assegurar que em média as comunidades ficam próximas do serviço a que estão afectas, garantindo que a comunidade mais próxima de um serviço instalado fica à maior distância possível desse serviço.

No outro modo de encarar o problema, as duas funções objectivo são:

- (iii) maximizar a soma ponderada das distâncias entre as comunidades e os serviços;
- (iv) minimizar a distância máxima entre as comunidades e o serviço mais próximo ao qual estão afectas.

Aqui garante-se que em média os serviços ficam o mais longe possível das comunidades que servem, mas suficientemente perto para que seja viável utilizá-los em tempo útil.

Designando por **P3** o problema a que correspondem as funções objectivo (i) e (ii) e por **P4** o problema com as funções objectivo (iii) e (iv), a formulação destes problemas utilizando a notação apresentada na secção 2 é a seguinte:

$$(P3) \quad \min \sum_{j \in J} \sum_{i \in I} c_{ij} x_{ij}$$

$$\max_{S \subseteq J} \min_{\substack{i \in I \\ j \in S}} d_{ij} y_j$$

$$\text{s.a:} \quad \sum_{j \in J} x_{ij} = 1 \quad i \in I \quad (4.1)$$

$$\sum_{j \in J} y_j \leq p \quad (4.2)$$

$$x_{ij} \leq y_j \quad i \in I, j \in J \quad (4.3)$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J \quad (4.4)$$

onde  $S$  designa qualquer subconjunto de  $J$  constituído pelos locais escolhidos para instalar os serviços. As restrições deste problema são as características de um problema de localização de  $p$  serviços.

$$(P4) \quad \max \sum_{j \in J} \sum_{i \in I} c_{ij} x_{ij}$$

$$\min_{S \subseteq J} \max_{\substack{i \in I \\ j \in S}} d_{ij} x_{ij}$$

$$\text{s.a:} \quad \sum_{j \in J} x_{ij} = 1 \quad i \in I \quad (4.1)$$

$$\sum_{j \in J} y_j \leq p \quad (4.5)$$

$$x_{ij} \leq y_j \quad i \in I, j \in J \quad (4.3)$$

$$d_{ij} (y_j - x_{ij}) \leq d_{ik} (1 - x_{ik}) \quad i \in I; j, k \in J \quad (4.6)$$

$$j \neq k, d_{ij} < d_{ik}$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J \quad (4.4)$$

Para a resolução dos problemas **P3** e **P4** podemos considerar dois métodos.

Um dos métodos consiste em considerar como função objectivo do problema a combinação convexa das duas funções objectivo. Determinam-se os pesos  $\alpha$  e  $1 - \alpha$ , com  $0 \leq \alpha \leq 1$ , a atribuir a cada uma das funções e, para cada valor de  $\alpha$ , obtém-se a solução correspondente. Este método não foi estudado no presente trabalho sendo, no entanto, um dos objectivos de trabalhos futuros.

O outro método de resolução consiste em incorporar uma das funções objectivo como restrição do problema, ficando apenas a outra como função objectivo. Assim, a resolução de um problema com dois objectivos vai resultar na resolução de uma sequência de problemas com uma só função objectivo.

Se no problema **P3** incorporarmos a função (ii) como restrição e considerarmos como função objectivo a função (i) resulta o problema **P5(D)** cuja formulação é:

$$\begin{aligned}
 \text{(P5(D))} \quad & \min \sum_{j \in J} \sum_{i \in I} c_{ij} x_{ij} \\
 \text{s.a:} \quad & \sum_{j \in J} x_{ij} = 1 \quad i \in I \quad (4.1)
 \end{aligned}$$

$$\sum_{j \in J} y_j = p \quad (4.2)$$

$$x_{ij} \leq y_j \quad i \in I, j \in J \quad (4.3)$$

$$\min_{\substack{i \in I \\ j \in S}} d_{ij} y_j \geq D \quad S \subseteq J \quad (4.7)$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J \quad (4.4)$$

O problema P5(D) coincide com o problema P1 para o qual já se apresentaram métodos para determinar limites superiores e inferiores [3]. O problema P5(D) deve ser resolvido para diferentes valores do parâmetro D com vista a ter uma ideia do que seria resolver P3.

Se no problema P3 considerarmos a função (i) como restrição e a função (ii) como função objectivo tem-se o problema que designaremos por P6(R) e cuja formulação é:

$$\begin{aligned}
 \text{(P7(R))} \quad & \max_{S \subseteq J} \min_{\substack{i \in I \\ j \in S}} d_{ij} y_j \\
 \text{s.a:} \quad & \sum_{j \in J} x_{ij} = 1 \quad i \in I \quad (4.1)
 \end{aligned}$$

$$\sum_{j \in J} y_j \leq p \quad (4.2)$$

$$x_{ij} \leq y_j \quad i \in I, j \in J \quad (4.3)$$

$$\sum_{j \in J} \sum_{i \in I} c_{ij} x_{ij} \leq R \quad (4.8)$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J \quad (4.4)$$

A restrição (4.8) garante que em média toda a comunidade está a uma distância menor ou igual a  $R / \sum_{i \in I} a_i$  do serviço mais próximo a que fica afectada, sendo R um parâmetro dado. Este

problema deve ser resolvido para diferentes valores do parâmetro R.

Determinam-se limites inferiores para este problema, obtendo soluções admissíveis para ele e calculando o correspondente valor da função objectivo. Os métodos utilizados para obter soluções admissíveis para P6(R) são do tipo "greedy" [3].

O problema P4 dá origem, tal como o problema P3, a dois problemas distintos, consoante a função objectivo que se incorpora como restrição. Assim, se a função objectivo (iv) passar a ser uma restrição do problema e (iii) continuar como função objectivo, tem-se o problema P7(d), cuja formulação é a seguinte:

$$\begin{aligned}
 \text{(P7(d))} \quad & \max \sum_{j \in J} \sum_{i \in I} c_{ij} x_{ij} \\
 \text{s.a:} \quad & \sum_{j \in J} x_{ij} = 1 \quad i \in I \quad (4.1)
 \end{aligned}$$

$$\sum_{j \in J} y_j \leq p \quad (4.5)$$

$$x_{ij} \leq y_j \quad i \in I, j \in J \quad (4.3)$$

$$d_{ij} (y_j - x_{ij}) \leq d_{ik} (1 - x_{ik}) \quad i \in I; j, k \in J \quad (4.6)$$

$$j \neq k, d_{ij} < d_{ik}$$

$$\max_{\substack{i \in I \\ j \in S}} d_{ij} x_{ij} \leq d \quad S \subseteq J \quad (4.9)$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J \quad (4.4)$$

As restrições (4.9) garantem que toda a comunidade está a uma distância máxima do serviço a que fica afecto, que não excede o valor do parâmetro d. As restrições (4.3) conjuntamente com as restrições (4.9) são equivalentes às restrições

$$d_{ij} x_{ij} \leq d y_j \quad i \in I, j \in J \quad (4.10)$$

Se em P7(d) substituirmos (4.3) e (4.9) por (4.10) obtém-se o problema P2 apresentado na secção 3.

Se, no problema P4, consideramos (iv) como função objectivo e (iii) como restrição, resulta o problema P8(T) cuja formulação é:

$$(P8(T)) \quad \min_{S \subseteq J} \max_{\substack{i \in I \\ j \in S}} d_{ij} x_{ij}$$

$$\text{s.a:} \quad \sum_{j \in J} x_{ij} = 1 \quad i \in I \quad (4.1)$$

$$\sum_{j \in J} y_j \leq p \quad (4.5)$$

$$x_{ij} \leq y_j \quad i \in I, j \in J \quad (4.3)$$

$$d_{ij} (y_j - x_{ij}) \leq d_{ik} (1 - x_{ik}) \quad i \in I; j, k \in J \quad (4.6)$$

$$j \neq k, d_{ij} < d_{ik}$$

$$\sum_{j \in J} \sum_{i \in I} c_{ij} x_{ij} \geq T \quad (4.11)$$

$$x_{ij}, y_j \in \{0,1\} \quad i \in I, j \in J \quad (4.4)$$

A restrição (4.11) garante que em média, toda a comunidade está a uma distância maior ou igual a  $T / \sum_{i \in I} a_i$  do serviço mais próximo a que fica afecto, onde T é um parâmetro dado. Tal

como no caso anterior devem experimentar-se diferentes valores para o parâmetro T. Dado que o objectivo é minimizar a maior distância entre cada comunidade e o equipamento que a serve, a afectação de cada comunidade é feita ao equipamento instalado mais próximo, pelo que as restrições (4.6) são redundantes.

Os métodos utilizados para obter soluções admissíveis para P8(R) são do tipo "greedy" [3].

### 5. Resultados computacionais

Foram feitos vários testes computacionais com problemas gerados aleatoriamente. Geraram-se no plano os pontos relativos às comunidades e aos potenciais locais de instalação de serviços, ficando estes numa área exterior à delimitada pelo conjunto de comunidades, uma

vez que se admitiu que, nos casos reais, não faz sentido localizar este tipo de serviços no meio das comunidades. Constrói-se seguidamente com base nestas comunidades e potenciais locais de instalação de serviços uma rede com uma densidade pretendida.

A descrição da geração dos problemas é feita detalhadamente em [3]. Geraram-se 43 tipos de exemplos que diferem no valor dos parâmetros  $m$  (número de potenciais locais de instalação de instalação de serviços) e  $n$  (número de comunidades). Para cada um destes tipos geraram-se cinco exemplos. Consideraram-se sempre redes de densidade 0.6. Para uma maior facilidade de leitura dos resultados computacionais obtidos agruparam-se os problemas em nove grupos de acordo com a seguinte tabela:

Grupo I	$m = 3, 4, 5$	$n = 10, 20$
Grupo II	$m = 6, 7, 8$	$n = 10, 20$
Grupo III	$m = 9, 10$	$n = 10, 20$
Grupo IV	$m = 3, 4, 5$	$n = 30, 50$
Grupo V	$m = 6, 7, 8$	$n = 30, 50$
Grupo VI	$m = 9, 10$	$n = 30, 50$
Grupo VII	$m = 3, 4, 5$	$n = 75, 100$
Grupo VIII	$m = 6, 7, 8$	$n = 75, 100$
Grupo IX	$m = 9, 10$	$n = 75, 100$

Todos os resultados computacionais relativos às quatro abordagens propostas para o problema de localização de serviços semiobnóxios foram obtidos com estes exemplos num microcomputador Data General com processador 286, 12 Mhz. Todos os métodos apresentados para a resolução deste problema foram implementados em linguagem Fortran. Os tempos de cpu são em segundos e não incluem o tempo de leitura dos dados e escrita de resultados.

Para a abordagem ao problema de localização de serviços semiobnóxios descrita na secção 2, apresentam-se nas tabelas 5.1 e 5.2 os valores mínimos, médios e máximos dos desvios obtidos com cada uma das heurísticas primais ou duais quando se consideram, respectivamente, os valores de  $p = 2$  e  $p = 3$ . Na 1ª coluna destas tabelas encontra-se o número do grupo e na 3ª, 4ª, 5ª, 6ª, 7ª e 8ª coluna estão os desvios percentuais respeitantes, respectivamente, à heurística "greedy", à heurística de Maranzana, à heurística Omega, à heurística Delta, à heurística Primal Dual e à heurística Dual. O desvio percentual em relação ao valor óptimo é calculado, para cada exemplo, no caso dos limites superiores pela fórmula  $\frac{LS - v(P)}{v(P)} \times 100$  onde LS representa o limite superior e  $v(P)$  o valor óptimo do problema P.

No caso dos limites inferiores o desvio percentual é dado por  $\frac{v(P) - LI}{v(P)} \times 100$  onde LI representa o limite inferior.

A relaxação lagrangeana forneceu em todos os casos a solução óptima. Os resultados computacionais para as heurísticas de melhoramentos locais foram obtidos a partir de uma

solução inicial construída pela heurística "greedy". Na heurística de Maranzana a solução inicial é constituída pelos p primeiros potenciais locais de instalação de serviços.

Grupo		Greedy	Maranz.	Omega	Delta	P.Dual	Dual
I	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.08	17.23	0.10	0.00	0.08	0.03
	Max	1.70	72.80	2.20	0.00	1.70	0.50
II	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.17	20.23	0.47	0.10	0.10	0.33
	Max	1.70	75.00	6.90	0.90	0.90	2.60
III	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	1.05	17.35	0.55	0.15	0.05	0.10
	Max	7.20	62.00	5.30	1.90	0.90	0.70
IV	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.43	2.75	0.25	0.00	0.20	0.20
	Max	8.80	38.70	6.00	0.00	4.00	3.00
V	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.13	13.17	1.63	0.08	0.12	0.20
	Max	2.60	78.80	20.80	1.60	1.80	3.40
VI	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.54	24.38	1.42	0.24	0.24	0.53
	Max	8.30	112.00	15.90	3.90	3.90	3.90
VII	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.02	2.23	0.02	0.05	0.05	0.13
	Max	0.40	18.30	0.40	1.00	1.00	3.00
VIII	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.12	8.85	1.82	0.46	0.60	0.53
	Max	4.70	51.00	16.90	6.00	6.00	7.20
IX	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.00	13.33	1.65	0.34	0.34	0.20
	Max	0.00	67.00	20.00	7.70	7.70	1.50

Tabela 5.1 - Desvios correspondentes aos problemas testados para  $p = 2$

Quanto à qualidade dos limites superiores verifica-se que com as heurísticas de melhoramentos locais em quase todos os casos se obteve a solução óptima. Existiram apenas quatro excepções para as quais o maior desvio médio foi de 0.3%. Em segundo lugar aparece a heurística Delta e depois a heurística Primal Dual. Um pouco mais abaixo temos as heurísticas Omega e "greedy" originando desvios médios bastante semelhantes, embora em média a "greedy" seja ligeiramente melhor. Finalmente, o método que dá piores limites superiores é a heurística de Maranzana com desvios médios entre 2.23% e 35.48% e com desvios máximos que em alguns casos são de 108% e 112%. No que diz respeito aos tempos de execução verifica-se que todas as heurísticas para determinar limites superiores são muito rápidas. O

maior tempo médio de execução foi de 0,39 segundos. A heurística Dual é mais rápida que a relaxação lagrangeana desde que o número médio de iterações desta seja superior a 4.

Grupo		Greedy	Maranz.	Omega	Delta	P.Dual	Dual
II	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.30	26.70	0.30	0.00	0.00	0.01
	Max	2.40	76.00	2.40	0.00	0.05	0.50
III	Min	0.00	0.30	0.00	0.00	0.00	0.00
	Med	0.45	30.00	0.10	0.00	0.10	0.04
	Max	4.00	89.00	1.20	0.00	1.00	0.50
V	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.17	25.48	0.27	0.05	0.05	0.04
	Max	3.90	66.00	6.00	1.60	1.60	1.00
VI	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.30	35.48	0.70	0.00	0.00	0.04
	Max	4.00	108.00	7.40	0.60	0.40	0.50
VIII	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.24	15.55	0.58	0.25	0.10	0.09
	Max	3.00	57.80	7.80	2.50	2.50	2.00
IX	Min	0.00	0.00	0.00	0.00	0.00	0.00
	Med	0.08	29.13	0.68	0.00	0.00	0.05
	Max	1.10	71.00	10.00	0.00	0.00	0.80

Tabela 5.2 - Desvios correspondentes aos problemas testados para  $p = 3$

Para a abordagem que corresponde a minimizar o efeito obnóxico dos serviços sobre as comunidades, que é descrita na secção 3, calcularam-se limites inferiores para todos os exemplos gerados, utilizando as várias heurísticas descritas. A determinação de limites superiores recorrendo à relaxação linear só foi possível para os exemplos de menores dimensões ( Grupo I e IV e exemplos dos grupos II e V com  $m = 6$  ), sendo impraticável para os restantes em termos de tempo de cpu ( mais que 5h ). Os limites superiores assim obtidos são de má qualidade variando o desvio médio entre 7% e 39.6%. Para os exemplos de maiores dimensões, fez-se um estudo comparativo das várias heurísticas com base nos desvios relativamente ao melhor limite inferior disponível. O cálculo deste desvio é feito pela fórmula  $\frac{v(P) - v(P)^*}{v(P)^*} \times 100$  onde  $v(P)$  representa o limite inferior obtido pela heurística em análise

e  $v(P)^*$  o melhor limite inferior. Na tabela 5.3 apresentam-se os resultados assim obtidos. Na 1ª coluna desta tabela encontra-se o número do grupo e na 3ª, 4ª, 5ª, 6ª e 7ª coluna estão os desvios relativos aos limites inferiores obtidos, respectivamente, pela heurística "greedy", heurísticas de melhoramentos locais, algoritmo 3.2 e algoritmo 3.3.

Grupo		Greedy	ML1	ML2	Alg. 3.2	Alg. 3.3
II	Min	0.00	0.00	0.00	0.00	0.00
	Med	4.05	1.75	1.75	0.95	5.40
	Max	13.70	13.60	13.60	10.90	17.40
III	Min	0.00	0.00	0.00	0.00	0.00
	Med	7.95	2.80	2.95	1.30	4.95
	Max	23.60	12.80	12.80	7.40	23.40
V	Min	0.00	0.00	0.00	0.00	0.00
	Med	5.65	2.18	2.25	0.45	3.48
	Max	30.80	10.40	13.10	5.60	15.40
VI	Min	0.00	0.00	0.00	0.00	0.00
	Med	5.73	3.30	3.30	1.07	3.88
	Max	25.60	20.10	20.10	12.80	22.40
VII	Min	0.00	0.00	0.00	0.00	0.00
	Med	2.40	0.91	0.91	0.40	0.61
	Max	12.80	8.60	8.60	5.00	8.60
VIII	Min	0.00	0.00	0.00	0.00	0.00
	Med	5.67	3.37	3.48	0.55	4.03
	Max	25.30	17.80	17.80	7.10	25.60
IX	Min	0.00	0.00	0.00	0.00	0.00
	Med	11.95	5.73	5.80	0.58	9.95
	Max	40.80	29.70	29.70	5.20	30.60

Tabela 5.3 - Desvios correspondentes aos problemas testados

Quanto à qualidade dos limites inferiores fornecidos, verifica-se que a heurística que produz melhores limites é a descrita no algoritmo 3.2. Seguidamente estão as heurísticas de melhoramentos locais. A heurística "greedy" é a que origina maiores desvios médios na maioria dos casos sendo o maior desvio médio para esta heurística de 11.95%. Todas heurísticas são bastante rápidas. O maior tempo médio de execução corresponde à heurística descrita no algoritmo 3.3 e é de 1.45 segundos.

Relativamente às abordagens com dois objectivos determinaram-se limites inferiores para o problema P6(R) e limites superiores para o problema P8(T). Em termos de tempo de cpu, os métodos utilizados para a obtenção destes limites são bastante rápidos.

Não foi possível determinar limites superiores para o problema P6(R), pelo que, não se pode inferir nada sobre a qualidade dos limites obtidos no que diz respeito ao desvio relativamente ao valor óptimo. Pode, no entanto, ter interesse calcular estes limites, uma vez que, é possível compará-los com o valor da menor das distâncias entre uma comunidade e o serviço instalado mais próximo, respeitantes a uma solução do problema de localização de serviços semiobnóxios obtida por um dos processos descritos na secção 2. Assim, para determinados valores de  $p$  e de  $R$  correspondentes a uma solução calculada por algum dos métodos da secção 2, podem comparar-se as soluções obtidas pelos dois processos e escolher

aquela em que a menor das distâncias entre as comunidades e o serviço instalado mais próximo é maior. Verificou-se para os exemplos testados que, relativamente a este critério, a solução obtida para P6(R) é, em 22% dos casos, melhor que a correspondente solução determinada por algum dos métodos descritos na secção 2.

Também para o problema P8(T) não foi possível obter limites inferiores, pelo que, não se podem tirar conclusões quanto à qualidade dos limites obtidos no que diz respeito ao desvio em relação ao valor óptimo. Tal como já foi referido, no caso do problema P6(R), tem interesse calcular limites superiores para o problema P8(T), uma vez que se podem comparar as soluções obtidas por este processo, para determinados valores de  $p$  e  $T$ , com as soluções a que correspondem os valores de  $p$  e  $T$  utilizados e que são fornecidos por algum dos métodos descritos na secção 3. Escolhe-se a solução para a qual a maior distância entre uma comunidade e o equipamento que a serve é a menor possível. Relativamente a este critério, verificou-se que com os exemplos testados, em 19% dos casos, a solução obtida para P8(T) é melhor que a correspondente solução determinada por algum dos métodos descritos na secção 3.

Todos os resultados computacionais relativos às várias abordagens ao problema de localização de serviços semiobnóxios são apresentados detalhadamente em [3].

## 6. Conclusões

O problema de localização de serviços semiobnóxios é um problema de localização discreta em que, dado as características específicas dos serviços, se pretende atingir dois objectivos contraditórios.

Das diferentes perspectivas de encarar estes objectivos resultaram quatro problemas distintos de localização para os quais se apresentaram formulações e se propuseram métodos de resolução. Em cada uma das quatro abordagens efectuaram-se testes computacionais com exemplos gerados aleatoriamente, simulando dentro do possível a realidade.

Os métodos de obtenção de limites para o problema de localização apresentado na secção 2 revelaram-se muito rápidos. A determinação de limites inferiores para a abordagem ao problema de localização de serviços semiobnóxios descrita na secção 3 tal como o cálculo dos limites respectivos para os problemas referidos na secção 4 são bastante rápidos. Assim, perante um problema de localização de serviços semiobnóxios devem considerar-se as várias abordagens. Isto permite-nos comparar os resultados obtidos para os diferentes modos de encarar o problema e, ao mesmo tempo, utilizar os limites determinados para uma das abordagens como parâmetros de outra abordagem. Esta interacção conduz à obtenção de melhores resultados no que diz respeito a atingir os objectivos que caracterizam este problema.

### **Bibliografia**

- [1] Captivo, M.E., Algoritmos para o Problema da p-mediana, Tese de Doutoramento, Universidade de Lisboa, (1988).
- [2] Cornuejols, G., Fisher, M.L., Nemhauser, G.L., Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms, *Management Science*, 23 (1977) 789 - 810.
- [3] Fonseca, M.C., Localização de Serviços Semiobnóxios. Algumas Abordagens, Tese de Mestrado, Universidade de Lisboa, (1991).
- [4] Geoffrion, A.M., Lagrangean Relaxation for Integer Programming, *Mathematical Programming Study* 2 (1974) 82 - 114.
- [5] Held, M., Wolfe, P., Crowder, H.P., Validation of Subgradient Optimization, *Mathematical Programming* 6 (1974) 62 - 88.
- [6] Khumawala, B.M., An Efficient Algorithm for the p-Median Problem with Maximum Distance Constraints, *Geographical Analysis* 5 (1973) 309 - 321.
- [7] Maranzana, F.E., On the Location of Supply Points to Minimize Transport Costs, *Operations Research Quarterly* 15 (1964) 261 - 270.
- [8] Singhal, J., Marsten, R.E., Morin, T.L., Fixed Order Branch and Bound Methods for Mixed Integer Programming: the ZOOM System, *ORSA Journal on Computing* 1 (1989) 44 - 51.
- [9] Teitz, M.B., Bart, P., Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph, *Operations Research* 16 (1968) 955 - 961.

**Fotografia, Montagem**  
**Impressão e Acabamentos**  
Tip.Nocamil  
COIMBRA

## ÍNDICE

<i>J. F. Sousa e R. C. Guimarães</i> , Estudo da evolução previsível do processo de substituição de veículos ligeiros no mercado português .....	3
<i>M. A. Matos e L. M. Proença</i> , Método de amplificações sucessivas .....	21
<i>J. A. Mesa</i> , Location of extensive facilities: potential applications .....	35
<i>M. Vaz Pato</i> , Linear and Lagrangean penalties for variable bounding in ILP: an application to a covering problem .....	43
Prémios APDIO – SCHNEIDER 92	
Nota Introdutória .....	51
<i>F. Vieira</i> , Um sistema informático para apoio ao sequenciamento de tarefas ...	53
<i>A. Paias</i> , Escalonamento de pessoal tripulante: Utilização de uma técnica de relaxação de espaço de estudos para problemas de cobertura .....	71
<i>M. C. Fonseca</i> , Diferentes abordagens ao problema de localização de serviços semiobnóxios .....	95



Associação Portuguesa para o Desenvolvimento  
da Investigação Operacional

CÉSUR - Instituto Superior Técnico - Avenida António Pais  
1000 Lisboa - Telef. 80 74 55