

# INVESTIGAÇÃO OPERACIONAL

Junho 1988

Número 1

Volume 8

Publicação Científica da

**apodlo**

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 8 - nº 1 - Junho 1988

Publicação semestral

Director : J.M. Pinto Paixão  
(Fac. Ciências - Universidade de Lisboa)

## Comissão Editorial

Mordecai Avriel	(Israel)	A. Simões Monteiro	(NORMA)
João A. Branco	(IST - Univ. Técn. Lisboa)	Mohamed Najim	(ENSIAS - Argélia)
Josep Casanovas	(UPC - Espanha)	J. Manuel Oliveira	(EFACEC)
J. Dias Coelho	(FE - Univ. Nova Lisboa)	Fernando Pacheco	(Univ. Católica)
Nuno Crato	(NORMA - Açores)	A. Gouvêa Portela	(IST - Univ. Técn. Lisboa)
J.A. Romão Eusébio	(CIMPOR)	M. Baptista Rodrigues	(Partex - CPS)
A. Sousa Ferraria	(Petrogal)	A. Guimarães Rodrigues	(Univ. Minho)
D. V. Gokhale	(Estados Unidos)	Bernard Roy	(LAMSADE - França)
J. Borges Gouveia	(FE - Univ. Porto)	C. Moreira da Silva	(FE - Univ. Porto)
R. Campos Guimarães	(FE - Univ. Porto)	L. Valadares Tavares	(IST - Univ. Técn. Lisboa)
Masao Iri	(TU - Japão)	Isabel H. Themido	(IST - Univ. Técn. Lisboa)
Joaquim J. Júdice	(FC - Univ. Coimbra)	B. Calafate Vasconcelos	(FE - Univ. Porto)
A. Rinnoy Kan	(EU - Holanda)	José M. Viegas	(IST - Univ. Técn. Lisboa)
Nelson Maculan	(UF RJ - Brasil)	Andres Weintraub	(UC - Chile)

Este Volume é inteiramente dedicado à publicação de trabalhos apresentados ao

**III CONGRESSO DA APDIO**  
Coimbra - 1987

e 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 (DEIOC- Faculdade de Ciências de Lisboa).

A Revista "INVESTIGAÇÃO OPERACIONAL" está registada na Secretaria de Estado da Comunicação Social sob o nº108335.

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.

Assinatura : 3000\$00

## MODELOS DE INVESTIGAÇÃO OPERACIONAL PARA APOIAR A TOMADA DE DECISÃO EM SUPERMERCADOS

Isabel M<sup>a</sup> Hall Themido

I.S.T. / CESUR

### Resumo

Neste artigo apresentam-se alguns modelos de I.O. que permitem apoiar a tomada de decisões em supermercados.

Os modelos de vendas e gestão de stocks foram concebidos para bens de grande consumo objecto de forte actividade promocional o que origina uma grande variabilidade de vendas e substituição entre marcas em situações de rotura de stocks. São apresentadas regras empíricas sugeridas pelos modelos teóricos que, pela facilidade de implementação, são mais úteis à loja de média dimensão.

### Abstract

This paper introduces some OR models designed to support decision making in supermarkets.

The sales and inventory models presented were designed for items with strong promotional activity wich leads to high sales variability and substitution in stock-out situations. Some empirical rules suggested by the theoretical models are also presented.

### 1. INTRODUÇÃO

O caso de estudo que se apresenta neste trabalho resultou do projecto de investigação aplicado que serviu de base à tese de doutoramento do autor realizado na Universidade de Lancaster em Inglaterra. Por essa razão difere da grande maioria dos casos publicados na literatura que relatam estudos cujo objectivo é o de resolver um problema concreto ainda que, por vezes, mal definido.

O objectivo do projecto em causa era o de desenvolver e ensaiar modelos alternativos de apoio às decisões comerciais dos gestores de lojas do tipo "Supermercado". Não se pretendia portanto resolver os problemas particulares da loja que serviu de "figurino" ao estudo mas sim desenvolver modelos que pudessem vir a ser utilizados por lojas do mesmo ramo e dimensão. Uma das preocupações que esteve sempre presente foi a de encontrar modelos de apoio à decisão de pouca com-

plexidade adequados à frequência das decisões e ao nível dos recursos humanos e materiais disponíveis em supermercados.

Tipicamente um supermercado vende uns largos milhares de itens em relação aos quais toma decisões todas as semanas. Nomeadamente no que se refere à quantidade a encomendar, ao preço a que vai vender o artigo e à actividade promocional que este vai disfrutar (seja uma baixa de preço, a utilização de um expositor especial, a inclusão numa campanha promocional publicitada na imprensa ou TV etc.). Usualmente as decisões são tomadas com base em informação pouco fiável ou mesmo inexistente, apoiadas exclusivamente na experiência e intuição comercial dos gestores. Mesmo quando existe informação em relação a vendas os valores referentes a uma combinação particular de produto, marca e tamanho são frequentemente baixos e portanto mais sujeitos a flutuações aleatórias que dificilmente podem ser ignoradas.

Assim sendo estará o gestor condenado a continuar a gerir a sua loja de forma tradicional? Ou será que tirando partido dos equipamentos de recolha de informação acoplados a caixas registadoras (nomeadamente leitores de códigos de barras), será possível recolher a informação que de futuro virá a alimentar modelos adequados às capacidades do retalhista, permitindo-lhe assim tomar decisões baseadas em métodos quantitativos ?

Os leitores ópticos de códigos de barras são comuns nos USA, começam a vulgarizar-se na Europa Industrializada e surgirão em breve em Portugal acompanhando a modernização que se vem sentindo neste sector da ecónomia e de que a onda crescente de "Hiper-mercados" é apenas um reflexo. A velocidade de introdução desses equipamentos é condicionada pela adopção sistemática de códigos de barras pelos fabricantes de artigos. As economias de mão-de-obra e diminuição do tempo de espera junto às caixas registadoras resultantes da introdução de leitores são incentivo suficiente para a sua introdução, só travado pelo facto da grande maioria dos produtos ainda não possuir código de barras.

O projecto de investigação em causa permitiu demonstrar que partindo de modelos de elevada complexidade é possível, pela introdução de restrições operacionais e/ou recurso a

heurísticas ou soluções aproximadas, chegar a modelos simples que satisfazem os objectivos propostos. Esses modelos aproximados foram testados recorrendo a um modelo de simulação das operações do retalhista. Neste artigo apresentam-se alguns dos resultados obtidos.

## 2. APRESENTAÇÃO DO ESTUDO

O Sr. Cliente entra no supermercado, passa quase sem olhar pelos expositores que ladeiam o seu percurso e dirige-se à zona dos produtos de limpeza para comprar o detergente que a mulher lhe encomendou - "Skup" - tamanho grande. À primeira vista não encontra. Certifica-se junto dum empregado que o produto está esgotado. O que fazer agora? A mulher recomendou que não se esquecesse. Não há outra loja junto do seu emprego. Decide finalmente que o melhor é comprar doutra marca. Há uma grande pilha de Xau com reclame de promoção. Tem uma vaga recordação do produto ser anunciado na TV. É isso, vai levar Xau - tamanho grande.

A Sr<sup>a</sup> Poupança entra no Supermercado onde usualmente faz as compras da semana. Gosta de fazer todas as compras nesta loja porque há sempre lugar para parar o carro, têm muita variedade e tirando partido dos produtos promovidos consegue poupar "uns cobres". Dirige-se ao escaparate dos enlatados para cão. A sua lista diz - 6 latas. Procura a "promoção da semana". Pega depois em latas de duas marcas que compra com frequência e compara efectivamente os preços. Finalmente decide levar 3 da marca promovida e, mais 3 de outra marca a que está habituada e que custa apenas uns escudos mais. Dirige-se depois ao escaparate dos detergentes. Lembra-se que na folha promocional que recebeu na caixa do correio o "Skup", a sua marca preferida, vem anunciada. Procura e não encontra. Finalmente repara num anuncio e numa prateleira vazia. Pensa que as promoções duram normalmente três semanas e que se lavar pouca roupa o resto que tem em casa vai durar até ao fim da semana. Decide adiar a compra para a semana seguinte.

Estes são apenas dois das centenas de clientes do supermercado que ao longo de vários meses levaram aos gráficos de

vendas representados na fig. 1.

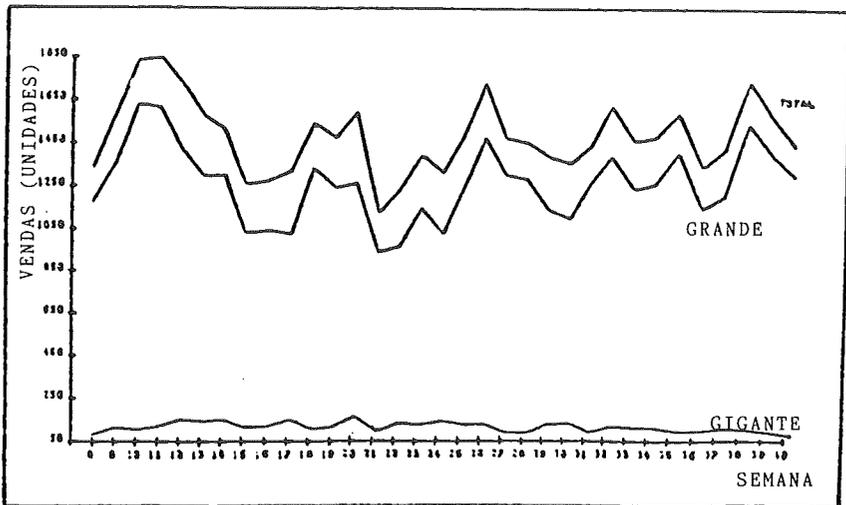
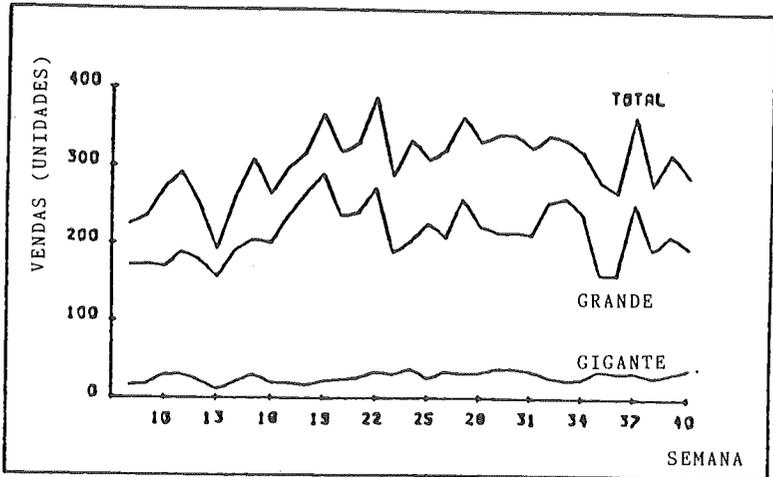


Fig. 1 - Vendas Semanais de duas Familias de Produtos

O que é surpreendente nos gráficos da figura 1 é a grande variabilidade das vendas totais do Grupo de Produtos, que agrega todas as marcas e tamanhos que constituem o grupo. Ambos os grupos de produtos são constituídos por artigos de grande consumo em que há substituição significativa entre marcas resultante quer de alterações do preço relativo das marcas em consequência de actividade promocional quer de roturas de stocks.

Essa grande interacção entre as marcas leva a que a formulação correcta do problema exija o tratamento do Grupo de Produtos como um todo, já que para o retalhista pouco importa se vende a marca A ou B desde que o seu volume global de vendas seja da mesma ordem de grandeza.

Assim, houve que considerar modelos de previsão de vendas que contemplam a interacção existente entre as marcas e adoptar como medida de performance para o Estudo o Lucro Bruto do Grupo de Produtos.

### 3. MODELO DE VENDAS

A análise dos valores das vendas semanais para a família de produtos apresentados na figura 1 permite concluir:

- a) Grande variabilidade
- b) Não haver evidência de substituição entre tamanhos (esta informação foi confirmada através de entrevistas a clientes)
- c) Os tamanhos menos populares têm vendas mais estáveis. Este efeito é o resultado de actividade promocional menos intensa.

O conhecimento do problema permitia supor que a grande variabilidade encontrada era resultado de alterações nas variáveis de Marketing.

A análise das vendas marca a marca permitiu concluir que os picos ocorrem durante semanas em que a marca está promovida.

Alguns ensaios preliminares bem como a observação do processo de comparação de preços "in-loco" efectuado por muitos clientes levou à adopção de modelos em que as vendas de uma marca dependem dos valores das variáveis de marketing de todas as marcas que constituem a família de produtos.

O modelo recomendado, para prevêr as vendas de uma marca é do tipo:

$$S_i = TS \times SH_i$$

onde:

$S_i$  = vendas da marca  $i$  durante a semana  $j$

$TS$  = vendas totais da família de produtos durante a semana  $j$

$SH_i$  = quota de mercado na loja, da marca  $i$  durante a semana  $j$

### 3.1 Modelo para as Vendas Totais

O modelo recomendado para as Vendas Totais da família é linear do tipo

$$TS = KT + \alpha PL$$

onde  $KT$  e  $\alpha$  são parâmetros e  $PL$  é um preço médio ponderado da seguinte maneira:

$$PL = \frac{m}{\sum_{i=1}^m} \bar{SH}_i P_i$$

onde:

$i$  = marca

$m$  = número de marcas na família

$\bar{SH}_i$  = quota média de mercado da marca  $i$  nos últimos 6 meses

$P_i$  = preço da marca  $i$

O modelo apresentado para  $TS$  é uma versão simplificada já que há ainda que tomar em consideração os aspectos seguintes que estão discutidos em pormenor em (1) :

- i) Ao ajustar a expressão a dados reais há que corrigir os valores de  $TS$  em relação às vendas perdidas resultantes de roturas de stocks.
- ii) O preço médio ponderado  $PL$  tem de ser ajustado em relação à inflação. Este ajustamento, ao nível da loja não é trivial.

O modelo proposto foi obtido depois de ensaiar vários outros modelos alternativos a par de um grande número de variáveis de marketing cujo poder explicativo é insignificante quando comparado com os bons resultados obtidos com o preço médio.

O modelo comporta-se tanto melhor quanto maior fôr a variabilidade da variável explicativa PL o que está associada a actividade promocional intensa.

### 3.2 Modelo para a Quota de Mercado

O modelo recomendado para a quota de mercado de cada marca é do tipo

$$SH_i = So_i \quad YP_i^{-wi}$$

onde  $So_i$  e  $w_i$  são parâmetros do modelo e  $YP_i$  é o preço relativo da marca  $i$  dado por:

$$YP_i = \frac{P_i (1 - \bar{SH}_i)}{\sum_{j \neq i} P_j \bar{SH}_j}$$

O preço relativo faz reflectir o efeito do preço das outras marcas nas vendas duma determinada marca. Esta técnica utilizada também por Frank, R.E. e Massy, W.F. (2) é a forma mais simples de modelar a interacção entre as marcas. Tem no entanto a desvantagem de não garantir que os valores de  $SH_i$  são consistentes, isto é que

$$\sum SH_i = 1$$

como seria desejável. Na prática, quando os modelos foram testados em condições mais severas do que se julga provável ocorrer na realidade, obtiveram-se desvios em relação à unidade pequenos. No entanto é recomendável que quando os modelos são utilizados com fins previsionais os valores obtidos para a quota de mercado de cada marca sejam posteriormente corrigidos dividindo pela soma das quotas obtidas para todas as marcas na família.

Foi portanto possível concluir que um modelo econométrico muito simples, baseado exclusivamente no preço de venda permite prever, com um grau de precisão adequado para a situação do retalhista, as vendas semanais de famílias de produtos consideradas de forma integrada. As dezenas de variáveis de marketing testadas não introduziam melhorias significativas.

Esta conclusão que vem reafirmar resultados propostos por Doyle e Gidengil's (3) é de grande importância para o retalhista, porque lhe permite prever vendas a partir de modelos simples alimentados exclusivamente por informação recolhida em caixas registadoras.

#### 4 - MODELO DE STOCKS

As características fundamentais dum modelo de stocks para o retalhista são:

- i) **Procura variável no tempo** - As vendas de cada marca são variáveis aleatórias com médias que variam consideravelmente de semana para semana em função da actividade promocional.
- ii) **Substituição** - A substituição entre marcas em situação de rotura de stocks é significativa em muitas famílias de produtos o que influencia as vendas perdidas e os níveis de serviço.
- iii) **Oportunidades de Comprar Stock a Preço Reduzido** - A grande maioria das promoções oferecidas por supermercados são da iniciativa do fabricante que reduz o preço de venda ao supermercado. Este por sua vez passa o desconto ao público. É prática corrente os supermercados comprarem nestas ocasiões mais do que necessitam. Esse stock é depois vendido ao preço normal aumentando assim as suas margens de venda. As promoções, ocorrem no tempo de forma aleatória e os retalhistas têm que decidir em cada caso quanto stock comprar. Este valor depende do desconto dado pelo fabricante, do custo de posse do stock e das restrições de armazenagem e capital que possam existir.

A modelação dum sistema de stockagem com estas caracteris-

ticas é muito mais complexa do que os modelos de stockagem triviais correntemente utilizadas. Por isso foi necessário procurar heurísticas e regras de gestão simples que pudessem ser implementadas. Essas regras foram depois testadas recorrendo a um modelo de simulação das operações do retalhista.

Partiu-se desde logo do pressuposto de que o modelo a utilizar será do tipo revisão cíclica com um período determinado não por critérios de optimalidade mas por condições operacionais do supermercado. O período de revisão na loja em estudo era de uma semana sendo a grande maioria dos produtos abastecidos ao mesmo tempo.

Este funcionamento é frequente noutras lojas. O abastecimento da loja, e o subsequente recarregar das prateleiras ocorre no início da semana, altura em que o movimento de clientes, na loja é menor. A maioria das lojas evita mesmo a todo o custo ter que reabastecer as prateleiras durante o período 6<sup>a</sup> / Sábado. Isto leva a que o volume de prateleira alocado a famílias de produtos de grande consumo seja grande.

Como é impensável que no dia imediatamente anterior ao reabastecimento das prateleiras estas se apresentem vazias somos levadas, por razões de imagem, a stocks de segurança elevados (factores de segurança superiores a 2, frequentemente 3). Estes valores são muito superiores aos que seriam recomendados por modelos que tomam em consideração formalmente a substituição entre marcas. Assim tendo deduzido modelos que tratam formalmente o problema da substituição (1) foi possível comparar os resultados obtidos com esses modelos com as restrições operacionais do retalhista podendo concluir-se que ao nível da loja não é realmente justificável a adopção de tais modelos.

#### 4.1 Políticas de Revisão Cíclica e Nível Variável

O modelo de revisão cíclica e nível variável, proposto para tratar a situação de procura variável, recomenda que o stock-em-mão mais em encomenda venha igual a um valor que varia de semana para semana. Utilizando a notação de Peterson e Silver (4) o stock-em-mão mais em encomenda é dado por

$$UT_j = d_{R+L_j} + K \sigma_{R+L}$$

onde:

$d_{R+L,j}$  é a procura média durante o período  $j$ , dado pelo modelo de vendas

$K$  é o factor de segurança

$\sigma_{R+L}$  é o desvio padrão da procura durante  $R+L$  (comprimento do ciclo + tempo de entrega).  $\sigma_{R+L}$  representa a variação aleatória da procura não explicada pelo modelo de vendas.

A implementação deste modelo, embora simples, exige o apoio de meios informáticos para prevêr as vendas. Por isso foram ensaiados dois outros modelos alternativos para averiguar em que condições é de facto justificável a utilização do modelo de revisão cíclica e nível variável. Esses modelos são:

1 - Revisão cíclica com dois níveis - nesta política são fixados dois níveis para UT, um mais elevado para as semanas em que a marca está promovida e outro mais baixo para as restantes semanas.

2 - Revisão cíclica - esta é a política usual de revisão cíclica em que a variabilidade da procura resultante de efeitos causais como por exemplo promoções é ignorada, sendo toda a variabilidade da procura tratada como aleatória. Para aplicar esta política não é necessário prever vendas semanais já que é apenas necessário conhecer a média e variância da procura.

Os melhores ou piores resultados obtidos com cada um dos três modelos alternativos dependem da variabilidade da procura. O modelo de revisão cíclica é tradicionalmente utilizado em situações de procura bastante estável. O modelo de revisão cíclica e nível variável deu muito bons resultados para os produtos seleccionados para este estudo e em particular para as duas famílias que apresentavam maior variabilidade. O modelo de revisão cíclica com dois níveis dará resultados satisfatórios quando não houver grandes diferenças no nível da procura dentro das semanas promovidas.

Uma das preocupações do estudo foi obter expressões heu-

rísticas que permitissem avaliar, em termos de redução dos custos variáveis de stockagem para a família de produtos, as situações em que seria aplicável cada um dos modelos. Essas expressões apresentadas em (1), pags 287 e 288 foram depois testadas para os produtos em estudo podendo concluir-se que produzem estimativas da mesma ordem de grandeza das economias efectivamente conseguidas.

O factor determinante na escolha do modelo é a variabilidade da família de produtos dada por:

$$\frac{\sum_{i=1}^m SD_i}{\bar{T}D}$$

onde  $m = n^o$  de marcas

$SD_i$  = desvio padrão dos valores de venda observadas para a marca  $i$

$\bar{T}D$  = Vendas médias da família de produtos (soma das médias de todas as marcas)

Sempre que a variabilidade da família de produtos é superior a .5 justifica-se a utilização de um dos modelos de nível variável. Para as famílias de produtos e supermercado que serviram de base ao estudo a diminuição dos custos de stockagem (custo de posse + custo de rotura) obtido foi sempre superior a 50% quando a variabilidade excedida .5.

#### 4.2 Compra de Stock no fim do Período Promocional

A determinação da quantidade óptima a encomendar no fim do período promocional resulta do equilíbrio entre o custo adicional que representa a posse desse stock e o benefício resultante de adquirir o material a preço mais baixo. Se ignorarmos todas as restrições que usualmente se aplicam a um problema deste tipo, que discutiremos depois, o stock a adquirir, expresso em número de semanas de consumo médio ( $n$ ) é dado por:

$$n = \frac{2+r}{2r} \frac{p_2}{p_1} - \frac{r+1}{r} \quad (1)$$

onde  $r$  = custo de posse por semana expresso como percentagem do custo de compra (% semana)

$p_1$  = custo unitário de compra durante o período promocional

$p_2$  = custo unitário de compra fora do período promocional ( $p_2 > p_1$ )

A determinação de  $n$  por esta via não toma em consideração algumas restrições importantes do problema:

i) Não é desejável (ainda que eventualmente fosse mais económico) ter em armazém 'stock' de uma promoção anterior quando uma nova promoção chega ao mercado.

ii) Restrições de espaço de armazenagem e capital que podem eventualmente ser excedidas caso esta política seja estendida a todas as famílias de produtos.

O modelo de simulação das operações do retalhista foi utilizado para estudar este problema. Os primeiros ensaios levaram a concluir que a restrição i) pode ser imposta de forma adequada se se fizer

$$n^* = \text{Min} ( n, \text{MIP} )$$

onde

$n^*$  =  $n^0$  de semanas de stock a adquirir

$n$  = valor teórico dado pela expressão (1)

MIP = Período interpromocional médio.

Posteriormente foram ensaiados casos em que a compra no fim do período promocional era dado por

$$\text{Min} ( n \times \text{FP} , \text{MIP} )$$

onde FP é um factor compreendido entre 0 e 1 fixado pelo utilizador.

O resultado obtido para uma das famílias de produtos está representado na figura 2. O primeiro ponto da curva corresponde a  $FP = 0$ . O último a  $FP = 1$ .

É fácil de ver que o lucro sobe rapidamente com o aumento do stock de segurança. Este aumenta por via da compra de stock no fim de todas as promoções. O lucro máximo obtém-se para valores do stock de segurança que correspondem a  $FP = .3$ . A partir daí os benefícios de comprar stock mais barato são anulados pelo custo adicional de posse deste stock.

O lucro máximo foi obtido para valores de  $FP$  cerca de  $.25/.3$  para as três famílias de produtos estudadas. Mais ensaios são necessários para verificar se esta conclusão se pode estender a outras lojas e grupos de produtos.

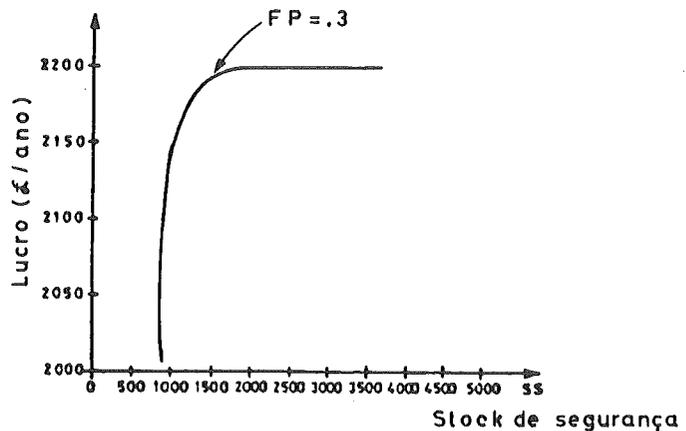


Fig. 2 - Lucro associado à compra de stock no fim dos períodos promocionais

## 5. CONCLUSÃO

Apresentam-se alguns modelos decisoriais especialmente concebidos para apoiar a tomada de decisões de lojas do tipo supermercado. É de realçar a simplicidade de muitos deles e em particular o facto da informação necessária para os alimentar poder ser recolhida directamente através de leitores ópticos acoplados a caixas registadoras.

## 6. BIBLIOGRAFIA

(1) Themido, Isabel Hall (1984). Pricing and Inventory Control in Retailing" - Tese de Doutoramento, Universidade de Lancaster.

(2) Frank, R.E and W.F. Massy (1965) "Market segmentation and the effectiveness of a brand's price and dealing policies", *Jornal of Business*, 38, April 1965, pp 186 - 200.

(3) Doyle, P. and B.Z. Gidengil (1977). "A review of instore experiments", *Jornal of Retailing*, 53(2), 1977.

(4) Peterson, Robert A. and E.A. Silver (1979) "Decision Systems for Inventory Management and Production Planning". John Wiley & Sons, New York.

# APLICAÇÃO DA PROGRAMAÇÃO DINÂMICA A OPTIMIZAÇÃO DE SISTEMAS DE TRATAMENTO

Paula Garcia Carranca\*  
João de Quinhones Levy\*\*

\* Instituto Superior de Tecnologia e Gestão (Faro)

\*\* Instituto Superior Técnico (Lisboa)

## Resumo

Atendendo ao elevado custo dos sistemas de tratamento de águas residuais e à multiplicidade de esquemas alternativos, apresenta-se um algoritmo de optimização para identificar a solução de mais baixo custo.

Este algoritmo utiliza a programação dinâmica como técnica de optimização tendo em conta que as condições de fronteira estão definidas à partida e que existem sucessivos estádios de decisão.

## Abstract

Considering the high costs associated to wastewater treatment facilities and the multiple number of existing alternatives to be considered, this paper presents an algorithm of optimization to identify the lowest cost solution.

This algorithm uses the Dynamic Programming as the optimization method, due to pre-established limits and also following decision stages.

**Keywords:** Cost, Wastewater treatment, Dynamic, Programming.

## 1 - INTRODUÇÃO

O fraco nível de atendimento das populações em termos de tratamento de águas residuais, leva a que seja urgente implementar a construção dos necessários sistemas de tratamento.

Sendo estas infraestruturas onerosas, há que estabelecer criteriosamente a solução mais adequada, em termos técnicos e económicos.

Uma estação de tratamento é constituída por várias unidades com características e funções próprias. Dadas as interações entre elas, a optimização do sistema global de tratamento implica a optimização do conjunto e não de cada uma das unidades.

Devido às múltiplas combinações possíveis entre as diversas unidades, a selecção do esquema de tratamento mais económico envolve a realização de numerosos cálculos.

Com vista a apoiar o agente de decisão nesta tarefa, desenvolveu-se um algoritmo de optimização que permite identificar, com rapidez, a solução de custo mais baixo.

## 2 - DESCRIÇÃO DE UM SISTEMA DE TRATAMENTO TIPO

Para melhor compreensão do algoritmo desenvolvido, apresenta-se, neste capítulo, uma breve descrição de um sistema de tratamento de águas residuais.

Um sistema, ou estação de tratamento, é constituído por diversas unidades. Estas, têm características e funções próprias e têm como objectivo, reduzir as cargas poluidoras afluentes, através de processos físicos, químicos e biológicos.

Para a mesma eficácia de tratamento existem muitos sistemas alternativos que se distinguem entre si, pelas unidades que o constituem e pela forma como estas estão associadas.

Como exemplo de um sistema de tratamento escolheu-se o sistema de lamas activadas. Este sistema é constituído por duas fases, a líquida e a de lamas. Na Figura 1 esquematizam-se estas fases:

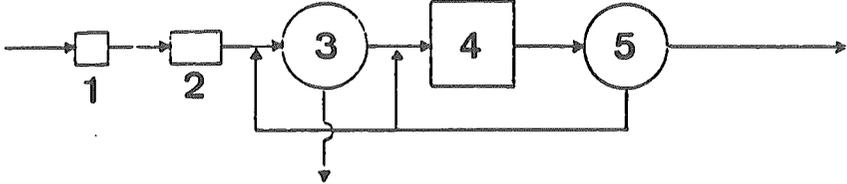
A fase líquida é constituída por:

- 1 - Câmara de grades
- 2 - Desarenador
- 3 - Decantador primário
- 4 - Tanque de arejamento
- 5 - Decantador secundário

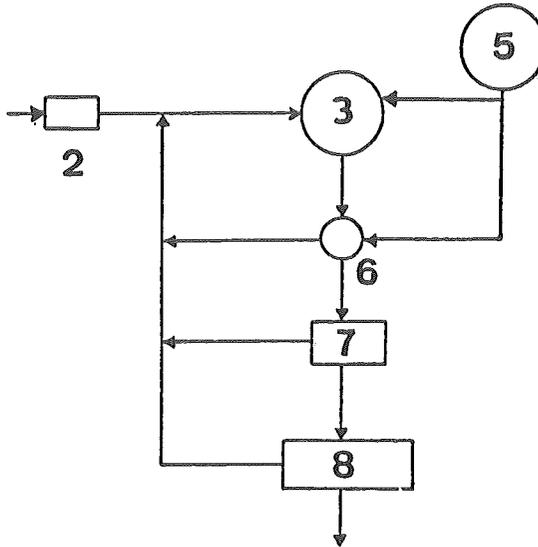
e a fase de lamas por:

- 6 - Espessador
- 7 - Digestor aeróbio
- 8 - Leitões de secagem

## FASE LÍQUIDA



## FASE DE LAMAS



## 3 - MÉTODO DE OPTIMIZAÇÃO

## 3.1 - DEFINIÇÃO DA FUNÇÃO OBJECTIVO

Estando definidos à partida os parâmetros de entrada e de saída, as alternativas de tratamento respeitam à forma de funcionamento das diversas unidades. Refere-se que não é uma análise custo-benefício visto que se considera que qualquer alternativa terá que garantir a mesma qualidade do efluente final.

Assim, o objectivo será o de determinar o esquema de tratamento mais económico que será conseguido pela minimização de uma função custo. Esta função custo será constituída pelas seis parcelas relativas ao custo associado a cada uma das unidades (3 a 8 na Figura 1).

Este custo associado, pode ser decomposto em dois elementos, o custo de primeiro estabelecimento (resultado do dimensionamento da unidade), e o custo de exploração (que resulta do seu modo de funcionamento).

O custo de exploração tem em conta a vida útil da estação pelo que considerando 20 anos, ter-se-ão 20 vezes o custo anual de exploração. Estes valores terão que ser reportados ao mesmo ano para poderem ser comparados de alternativa para alternativa. Escolhido o ano zero (de instalação) como ano de referência, o valor actualizado correspondente será,

$$C_E^* = \frac{C_E}{(1+i)^t}$$

em que:

- i - taxa de actualização
- t - ano considerado
- $C_E$  - custo de exploração não actualizado
- $C_E^*$  - custo de exploração actualizado ao ano zero

É assim imediato o cálculo do custo total reportado ao momento da instalação (ano zero):

$$C_T = C_I + C_E \cdot \frac{1}{(1+i)^t}$$

para uma certa taxa  $i$ , o valor do somatório pode ser facilmente obtido a partir de uma tabela financeira; assim para  $i=12.5\%$  esse valor será 7.47,

$$C_T = C_I + 7.47 C_E$$

Apresentam-se de seguida, as equações de custo para as várias unidades da cadeia de tratamento, indicando-se qual a variável que dará origem às várias alternativas de custo, bem como a fonte de cada uma das expressões.

- Decantador Primário, Decantador Secundário, Espessador de Lamas

$$C_I = 824 * 1.637 * A^{0.77}$$

$$C_E = (17.1 * 11.4 + 9.23 * 10.2 + 8.62 * 1.637) * A^{0.66}$$

Variável: Carga hidráulica superficial (m<sup>3</sup>/m<sup>2</sup>h)

Origem: TYTECA (1979) e CRAIG e col. (1978)

- Tanque de Arejamento e Digestor Aeróbio

$$C_I = 461 * 1.637 * V^{0.71} + 1000 * CA^{0.88}$$

$$C_E = 110 * 11.4 * CA^{0.54} + 45 * 10.2 * CA^{0.59} + 65 * 0.065 * CA$$

Variável: Tanque de Arejamento - Idade das lamas (dias)

Digestor Aeróbio - Teor de sólidos suspensos (mg/l)

Origem: CRAIG e col. (1978)

- Recirculação

$$C_I = 1710 * 1.637 * Q^{0.53}$$

$$C_E = 0.153 * 11.4 * Q + 0.0951 * 10.2 * Q + 6.11 * 1.637 * Q^{0.88} + 2.2 * Q * H$$

Origem: CRAIG e col. (1978)

- Leitos de Secagem

$$C_I = -5635 + 1636 * \ln(A)$$

$$C_E = 0.10 * C_I$$

Variável: Área (m<sup>2</sup>)

Origem: LEVY (1985)

Todas as expressões se encontram em \$ USA, excepto a referente ao Leito de Secagem cuja unidade é milhares de escudos.

## 3.2 - TÉCNICA ADOPTADA

A técnica adoptada para a optimização do sistema, minimização dos custos, foi a da Programação Dinâmica. A sua escolha deve-se ao facto de haver uma analogia entre os esquemas de tratamento e a sequência estrutural de um problema de multidecisão, não só devido ao facto de em ambos existirem sucessivos estádios decisórios, mas também por serem bem conhecidos os limites do pro-

blema em estudo. De facto, num sistema de tratamento está definido tanto o valor das cargas afluentes como o das cargas efluentes.

A Programação Dinâmica é uma técnica matemática que se adapta à resolução de problemas que exigem decisões, a serem tomadas em sequência e, que influenciarão a tomada de futuras decisões.

Ao contrário do que acontece com a Programação Linear, não existe um modelo matemático standard que se possa ajustar ao problema em causa; cada problema, é uma situação diferente a ser estudada. Se, por um lado, o facto de cada problema exigir decisões básicas de formulação constitui uma das dificuldades na aplicação da Programação Dinâmica, por outro, leva a que haja um conjunto mais variado de possíveis aplicações.

### 3.3 - EXEMPLO DE APLICAÇÃO

Ilustrar-se-á, de seguida, esta técnica, através da resolução de um exemplo muito simples.

Considere-se então o seguinte problema, em que o objectivo é o de minimizar o caminho a percorrer entre dois pontos A e B, da rede representada na Figura 2.

Como se pode observar, encontramos 4 estádios decisórios a que correspondem 4 decisões, que se influenciam sequencialmente. Os valores dos comprimentos entre cada nó encontram-se indicados nos quadros da Figura 3.

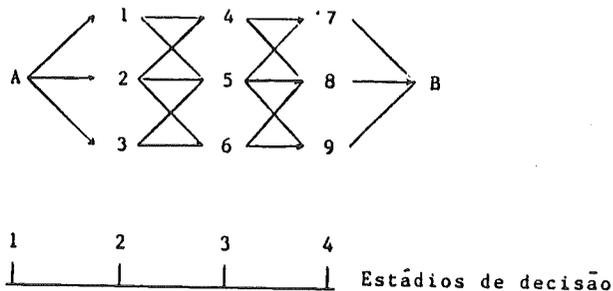


Fig. 2 - Estádios de decisão

	1	2	3		4	5	6		7	8	9		B
A	3	5	2	1	7	6	-	4	6	3	-	7	4
				2	4	3	5	5	7	5	4	8	3
				3	-	4	2	6	-	7	4	9	5

Fig.3 - Alternativas de cada  
estádio de decisão

Apresenta-se na Figura 4 a mesma rede mas agora referida como uma árvore de decisão, figurando todas as alternativas existentes.

Existem 17 caminhos possíveis a tomar para ir do ponto A a B.

A Programação Dinâmica divide o problema num conjunto de sub-problemas de mais fácil resolução, e analisa posteriormente os resultados obtidos. Há assim uma decomposição do problema.

O objectivo é então determinar L (distância mínima total), tal que:

$$L = \sum_i l_i$$

sendo  $l_i$  o comprimento associado à decisão tomada no estágio i.

Inicia-se a resolução do problema pelo último estágio de decisão, o estágio 4. Tem-se assim  $F4^*$  - valor óptimo associado a cada uma das alternativas - para o ponto  $X4^*$  respectivo.

P	$F4^*$	$X4^*$
7	4	B
8	3	B
9	5	B

No estágio 4 determinou-se qual o caminho a tomar, associando a cada um dos pontos do estágio. Passando para o estágio 3 (duas decisões remanescentes), a distância total a B é agora função do estágio anterior.

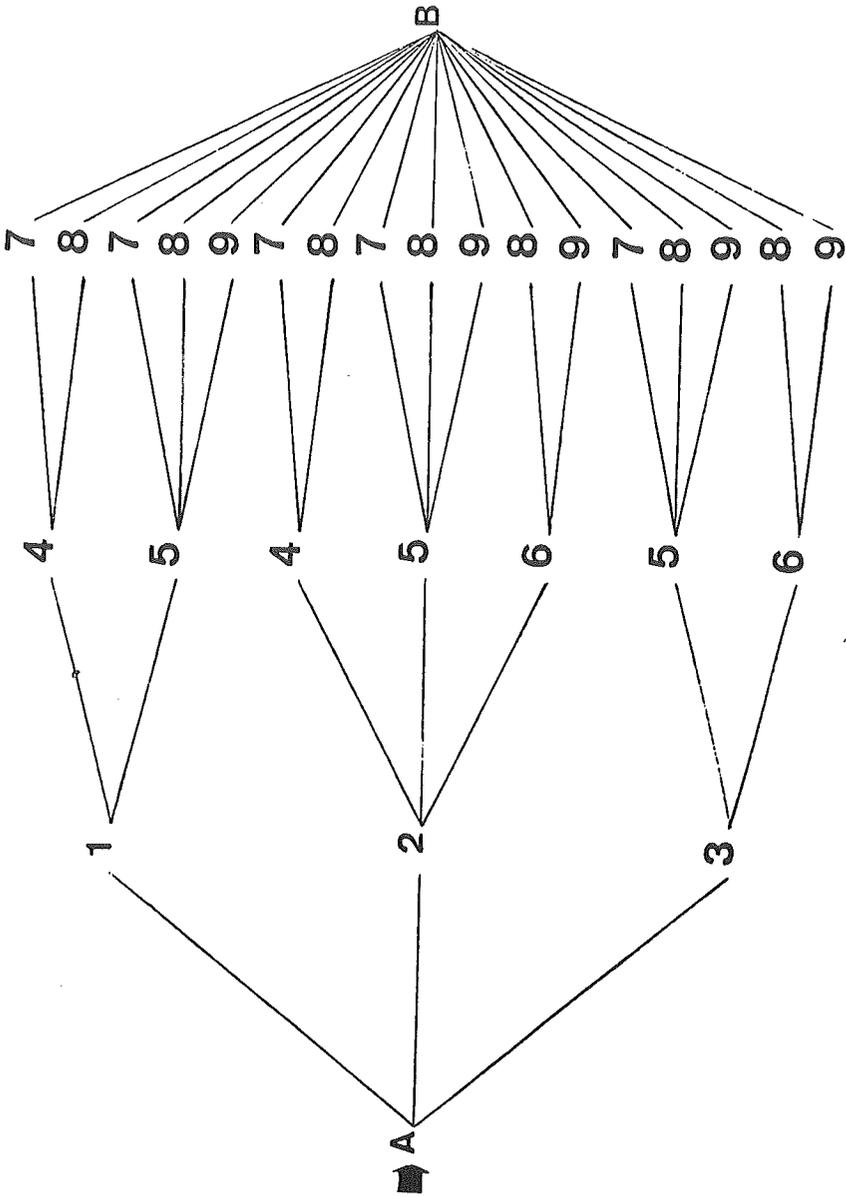


Figura 4 - Árvore de decisão

X3 P	F3			X3*	F3'	F4**	F3*
	7	8	9				
4	6	3	-	8	3	3	6
5	7	5	4	9	4	5	9
6	-	7	4	9	4	5	9

A solução do estágio 3 foi assim obtida dum modo idêntico. Determinou-se o valor de  $F3'$  para cada uma das alternativas desse estágio, adicionando depois o valor de  $F4^{**}$  associado ao ponto  $x3^*$ , obtendo-se assim o valor de  $F3^*$ .

Passando ao estágio 2, e seguindo o mesmo raciocínio,

X2 P	F2			X2*	F2'	F3**	F2*
	4	5	6				
1	7	6	-	5	6	9	15
2	4	3	5	5	3	9	12
3	-	4	2	6	2	9	11
3	-	4	2	6	2	9	11

Finalmente para o estágio 1, e considerando os valores até aí encontrados,

X1 P	F1			X1*	F1'	F2*	F1*
	1	2	3				
A	3	5	2	3	2	11	13

Assim, através de sucessivas decisões chega-se à solução final do problema. O caminho de A a B de menor distância é A-3-6-9-B, e o comprimento correspondente é de 13 unidades.

Evidenciou-se com este exemplo a importância da decisão tomada, sobre o resultado de todas as decisões futuras a que ela conduz. Demonstrou-se também um conceito básico desta técnica de optimização, que consiste na divisão de um problema por estádios, exigindo-se uma decisão em cada um deles.

Esta decisão está dependente não só das decisões já tomadas, mas também das condições de entrada para esse estágio. Resumidamente, para o estágio  $i$  a decisão no ponto  $P$  será

$$X_i^* \text{ tal que } F'(X_i^*) + F^*(X_{i+1}) = F_i^*$$

em que  $F_i^*$  é o menor valor da função nesse estágio.

### 3.4 - APLICAÇÃO DA PROGRAMAÇÃO DINÂMICA À OPTIMIZAÇÃO DE UM SISTEMA DE TRATAMENTO.

Numa estação de tratamento são conhecidas as seguintes condições limites

- cargas à entrada
- cargas à saída

ou seja, reportando ao exemplo apresentado anteriormente, nó inicial e nó final.

Seguidamente, para continuar com o estudo comparativo, tem-se a primeira unidade de tratamento - o decantador primário, que é o primeiro estágio a atingir. As diferentes formas de ele trabalhar dependem da carga hidráulica e constituem as alternativas deste estágio, que são afinal os diversos caminhos para o alcançar.

O segundo estágio é o tanque de arejamento onde as alternativas surgem agora sob a forma de concentração de sólidos suspensos voláteis. Sucessivamente no terceiro estágio encontra-se o decantador secundário.

Assim, cada unidade da cadeia tem associada a si um estágio decisório e pelo menos uma variável de decisão. Esta variável entra no cálculo do dimensionamento da unidade e, como tal, na determinação do seu custo.

Em resumo, as unidades de tratamento são os diversos níveis, e as diferentes alternativas de tratamento, os caminhos possíveis entre os estágios.

#### 4 - CONSTRUÇÃO DO MODELO

Como se referiu, a técnica escolhida para optimização dum sistema de tratamento, foi a Programação Dinâmica, tendo sido apontadas as analogias existentes e que tornaram possível a sua utilização.

Neste ponto definir-se-á o esquema de tratamento já como um conjunto de estádios decisórios, e os vários níveis de decisão alternativos, tendo em consideração os conceitos de Programação Dinâmica atrás referidos.

Associar-se-á a cada estádio decisório (unidade de tratamento) um custo, função das condições de entrada para esse estádio e, à semelhança do que se fez para a determinação do caminho mais curto, escolher-se-á agora o menor dos custos.

##### 4.1 - NÍVEIS DE DECISÃO ALTERNATIVOS

O sistema de tratamento escolhido como exemplo foi o apresentado no capítulo 2.

Indicam-se, de seguida, as alternativas consideradas para cada estádio decisório, e situa-se cada uma das unidades da cadeia no estádio correspondente.

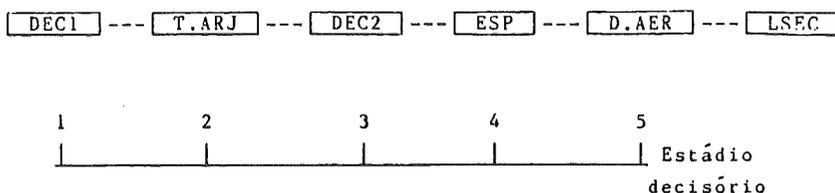


Fig.5 - Estádios decisórios

No decantador primário, estádio 1, consideram-se oito alternativas para a carga hidráulica superficial,  $Ch = 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75$  e  $2.00 \text{ m}^3/\text{m}^2.\text{d}$ . Cada uma destas alternativas conduzirá a uma solução de custo.

No tanqué de arejamento, estádio 2, as alternativas são seis e concernem à concentração em sólidos suspensos voláteis, SV6 - 2000, 2500, 2750, 3000, 3250 e 3500. Também a cada um destes valores corresponderá um custo para o tanque de arejamento.

O terceiro estádio é o decantador secundário. Nesta unidade adicionar-se-á ao custo do decantador a parcela referente ao custo da recirculação. As alternativas são oito e dizem respeito à concentração de sólidos, Str2 - 8000, 8500, 9000, 9500, 10000, 10500, 11000 e 12000 mg/l, o que originará oito valores para o custo desta unidade.

No quarto estádio, espessador de lammas, aparecem quatro alternativas em relação ao tempo das lammas no espessador, Te - 1, 2, 3, e 4 dias, dando assim origem a quatro valores de custos diferentes.

Por fim, no quinto estádio está o digestor aeróbio que tem três modos de funcionamento relativamente ao acréscimo da concentração máxima de sólidos suspensos totais. Este acréscimo é relativo à concentração afluyente Stii - 5000, 10 000 ou 15 000 mg/l. Existirão assim três custos alternativos para esta unidade.

A última unidade da cadeia de tratamento, os leitos de secagem, apresentam um único valor para custo, função da sua área.

Fazendo uma analogia entre este problema e o apresentado no ponto 3.3, é possível reformular a apresentação das várias alternativas de cada estádio tal como está indicado na Figura 3.

Estádio 1: (Ch - m<sup>3</sup>/m<sup>2</sup>h)

	TA 1	TA 2	TA 3	TA 4	TA 5	TA 6	TA 7	TA 8
DEC 1	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00

Estádio 2: (SV6 - mg/l)

	DEC21	DEC22	DEC23	DEC24	DEC25	DEC26
TA 1	2000	2500	2750	3000	3250	3500
TA 2	2000	2500	2750	3000	3250	3500
TA 3	2000	2500	2750	3000	3250	3500
TA 4	2000	2500	2750	3000	3250	3500
TA 5	2000	2500	2750	3000	3250	3500
TA 6	2000	2500	2750	3000	3250	3500
TA 7	2000	2500	2750	3000	3250	3500
TA 8	2000	2500	2750	3000	3250	3500

Estádio 3: (STR2 - mg/l)

	ESP 1	ESP 2	ESP 3	ESP 4	ESP 5	ESP 6	ESP 7	ESP 8
DEC21	8000	8500	9000	9500	10000	10500	11000	12000
DEC22	8000	8500	9000	9500	10000	10500	11000	12000
DEC23	8000	8500	9000	9500	10000	10500	11000	12000
DEC24	8000	8500	9000	9500	10000	10500	11000	12000
DEC25	8000	8500	9000	9500	10000	10500	11000	12000
DEC26	8000	8500	9000	9500	10000	10500	11000	12000

Estádio 4: (Te - dias)

	DA 1	DA 2	DA 3	DA 4
ESP 1	1	2	3	4
ESP 2	1	2	3	4
ESP 3	1	2	3	4
ESP 4	1	2	3	4
ESP 5	1	2	3	4
ESP 6	1	2	3	4
ESP 7	1	2	3	4
ESP 8	1	2	3	4

Estádio 5: (STii - mg/l)

	LSEC1	LSEC2	LSEC3
DA 1	+5000	+10000	+15000
DA 2	+5000	+10000	+15000
DA 3	+5000	+10000	+15000
DA 4	+5000	+10000	+15000

#### 4.2 - CARACTERÍSTICAS DO PROGRAMA DE CÁLCULO AUTOMÁTICO

A partir do modelo apresentado no ponto anterior, foi desenvolvido um programa de cálculo automático. Este programa foi escrito em linguagem PASCAL, sendo a sua listagem apresentada em anexo.

Tentou-se que o modelo adaptado fosse, em termos de execução, o mais rápido possível. O grande número de alternativas em cada estágio decisório, reflectiu-se em termos de programação num elevado número de ciclos a executar. Assim, abordou-se o problema de acordo com a árvore apresentada na Figura 6, (que ilustra somente parte das soluções possíveis).

No estágio 5 (a), e para as três alternativas, é determinado o mínimo, procedendo-se do mesmo modo para os casos (b), (c) e (d). Estes 4 casos - (a), (b), (c) e (d) - constituem por si as alternativas para o cálculo do custo no estágio 4 - Espessador (X1). Associa-se ao espessador (X1) o menor dos custos

$$(X1) + (a)$$

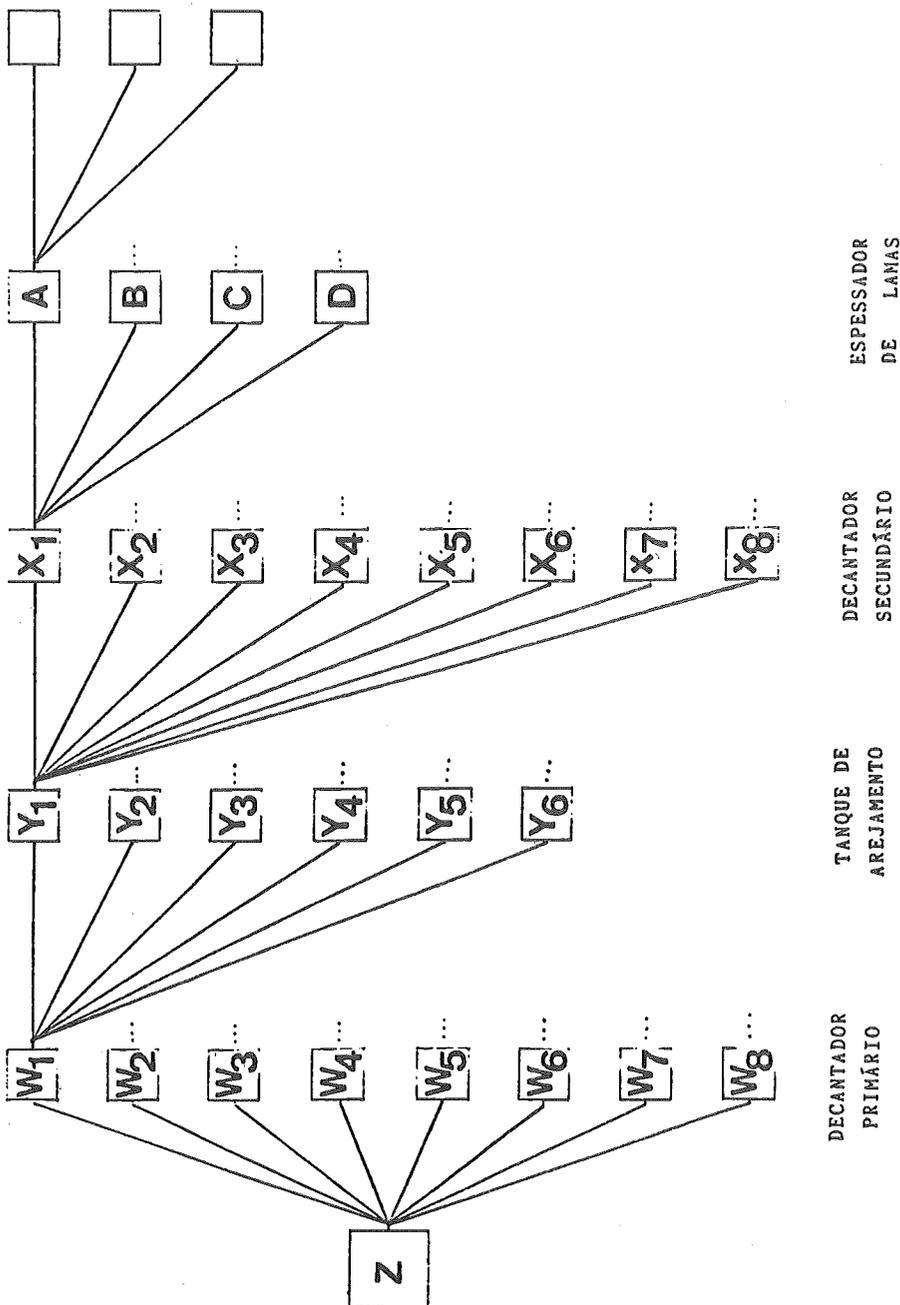
$$(X1) + (b)$$

$$(X1) + (c)$$

$$(X1) + (d)$$

obtendo-se o valor de  $(X1)^*$ .

Do mesmo modo, determinam-se os valores  $X2^*$ ,  $X3^*$ , até  $X8^*$ . Obtidos estes valores, pode-se agora escolher o menor deles e, juntamente com o custo do decantador secundário (Y1),



$$\begin{aligned} &(Y1) + (X1^*) \\ &(Y1) + (X2^*) \\ &(Y1) + (X3^*) \\ &(Y1) + (X4^*) \\ &(Y1) + (X5^*) \\ &(Y1) + (X6^*) \\ &(Y1) + (X7^*) \\ &(Y1) + (X8^*) \end{aligned}$$

determinar a solução de menor custo até ao estágio 3 ( $Y1^*$ ).

Este processo repetir-se-á no estágio 2 - Tanque de Arejamento - onde se comparam os seis custos correspondentes aos diferentes modos de funcionamento,

$$\begin{aligned} &(W1) + (Y1^*) \\ &(W1) + (Y2^*) \\ &(W1) + (Y3^*) \\ &(W1) + (Y4^*) \\ &(W1) + (Y5^*) \\ &(W1) + (Y6^*) \end{aligned}$$

para escolha do menor custo ( $W1^*$ ); finalmente para o estágio 1 - Decantador Primário,

$$\begin{aligned} &(Z) + (W1^*) \\ &(Z) + (W2^*) \\ &(Z) + (W3^*) \\ &(Z) + (W4^*) \\ &(Z) + (W5^*) \\ &(Z) + (W6^*) \\ &(Z) + (W7^*) \\ &(Z) + (W8^*) \end{aligned}$$

O menor dos custos obtido desta iteração, corresponderá à solução ótima para o sistema de tratamento, numa determinada situação, onde as cargas afluentes e efluentes são definidas à priori. Esta solução é apresentada não só em termos de custos mas também em termos de dimensionamento ótimo das unidades.

## 5 - CONCLUSÕES

Apresentou-se nesta comunicação um algoritmo para otimizar o dimensionamento de sistemas de tratamento de águas residuais.

Relativamente à técnica escolhida, a programação dinâmica, faz-se notar que embora implique uma simplificação nos ciclos de recirculação, consiste numa técnica de optimização de fácil utilização que torna bem clara as relações entre as diversas unidades de tratamento.

A finalizar, evidencia-se o interesse de técnicas simples de optimização que possam ser facilmente utilizadas na prática com recurso a pequenos microcomputadores.

## BIBLIOGRAFIA

Carranca, P., 1987: "Aplicação da Programação Dinâmica à Optimização de Sistemas de Tratamento", Cocite, Isma, Lisboa.

Hillier, F. and Lieberman, G., 1980: "Introduction to Operations Research", Holden-Dan Inc. 3<sup>rd</sup> Edition, 266-299.

Levy, J.Q., 1985: "Dimensionamento de Estações de Tratamento de Águas residuais Comunitárias - Modelo Interactivo", Tese de Doutoramento, I.S.T., Lisboa.

Shih, C. S. and De Filippi, John A., 1970: "System Optimization of Waste Treatment Plant Process Design", Journal of the SANITARY ENGINEERING DIVISION 96, 409-421.

Shih, C. S. and Krishnann, P., 1972: "System Optimization for Pulp and Paper Industrial Treatment Design", Water Research, Vol. 7, 1805-1820.

Sterling, 1976: "Compturized Algorithms Simplify Meeting Federal Effluent Standards", WATER & SEWAGE WORKS, Vol. 123, N<sup>o</sup> 10 68-72.

"Waste Treatment Selection", Vol. 41, n<sup>o</sup> 11, Part 1, 1846-1858, Journal WPCF, 1969.



EURO: the Association of European Operational  
Research Societies Within I.F.O.R.S.

Every year EURO organizes a summer institute  
for selected young scientists working in a particular field of O. R.

## SIXTH EURO SUMMER INSTITUTE

Madeira, Portugal, May 26 – June 10, 1989

### THEME FOR 1989: DECISION SUPPORT SYSTEMS

*Organization:* EURO and APDIO (The Portuguese O. R. Society), with the support of ORS (The British O. R. Society) and also of DTIM (Regional Association for the Development of Information Technologies in Madeira) and the University of Madeira.

*Activities:* The Institute will take place on Funchal, Madeira island. Accommodation will be offered by the EURO Summer Institute organization. Each participant will present a research paper on the above theme. Invited lectures by prominent speakers will be given. A special issue of EJOR (European Journal of Operational Research) will be prepared jointly during the stay.

*Application:* Admittance is limited to 20 promising young persons (age 25–35) from different countries. Applicants should send their curriculum vitae together with an unpublished paper in the field of Decision Support Systems to their national O.R. Society. These societies will select their national candidates and submit them before March 1 to the coordinator of the Sixth EURO Summer Institute. The final choice of the 20 participants will be made by the scientific committee.

*Further information:* available at your national O. R. Society.

*Coordinator:* Prof. J. Paixão; DEIOC – Faculty of Sciences, University of Lisbon; Avenida 24 de Julho, 134; 1300 Lisboa; Portugal.

*Scientific and Organizing Committee:* V. Belton, J.P. Brans, A.I. Cardoso, J. Clímaco, V. Miranda, J. Paixão, A. Rodrigues, I. Themido.

*Youth Committee:* A. Rodrigues, A.I. Barros, H. Lourenço.

## DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA AUTOMÁTICO DE ESCALAS DE AUTOCARROS E TRIPULAÇÕES NA RODOVIÁRIA NACIONAL

**J. Romão Eusébio, Lélío Amado, Lucinda Fragoso**  
RODOVIÁRIA NACIONAL  
Av. Columbano Bordalo Pinheiro, 86  
1096 Lisboa Codex

**José Paixão**  
FAC. CIÊNCIAS DE LISBOA - DEIOC  
Av. 24 de Julho, 134 - 5º  
1300 Lisboa

### **Resumo**

Com vista à elaboração de escalas de autocarros e tripulações foi desenvolvido na Rodoviária Nacional um sistema baseado em alguns modelos matemáticos e num subsistema de gestão de bases de dados, que permitem a realização daquele trabalho em microcomputador.

A presente comunicação descreve os modelos e a metodologia utilizados no desenvolvimento do sistema, bem como a correspondente estratégia de implementação, apresentando ainda resultados de algumas aplicações reais deste sistema.

### **Abstract**

This paper reports on the development and implementation of an automatic system for scheduling buses and crews at Rodoviária Nacional.

The system has been implemented on PC's and it is based on several mathematical models which are briefly described in this report.

The methodology adopted for the system as well as the corresponding implementation strategy and real life experience relative to its application are mentioned too.

**Keywords** System; Optimization; Methodology; Implementation.

### **1. INTRODUÇÃO**

A Rodoviária Nacional (RN) é a maior empresa de transporte rodoviário de passageiros em Portugal e uma das maiores da Europa. Foi fundada em 1976 em resultado da nacionalização de 93 empresas de transporte de passageiros e mercadorias.

A principal actividade da RN consiste no transporte rodoviário de passageiros por todo o País.

A empresa dispõe, à data, de uma frota superior a 3000 autocarros e emprega cerca de 12000 trabalhadores (5600 pessoal tripulante). As despesas com pessoal tripulante e com gasóleo representam respectivamente, 23 e 15% do total de custos (25300 milhões de escudos em 1986).

Desde 1977, a empresa tem vindo a desenvolver um sistema automático para elaboração de escalas de autocarros e de pessoal tripulante, com base em modelos de optimização. Numa primeira fase, o sistema foi concebido e desenvolvido por técnicos da RN e computacionalmente suportado em mainframe [9]. Mais tarde, em 1984, estabeleceu-se um processo de cooperação entre a empresa e a Universidade de Lisboa (Faculdade de Ciências), que tornou possível a introdução de novos algoritmos e a mudança de suporte para microcomputador. Somente nesta segunda fase se abriu o caminho à utilização do sistema de forma descentralizada e à sua aplicação generalizada a toda a empresa.

Entretanto, para além do trabalho I & D, o processo de implementação do sistema requereu outro tipo de actividades, de modo a assegurar todas as funções inerentes ao sistema.

Nos pontos 2 e 3 são dados pormenores sobre o sistema em computador usado na RN. No ponto 4 referem-se problemas surgidos na sua implementação. Finalmente, o ponto 5 relaciona-se com a experiência prática mostrando-se alguns valores relativos a poupanças conseguidas.

## **2. APRESENTAÇÃO DO SISTEMA**

Considerando um certo nível de oferta de transporte definida por uma procura conhecida, o objectivo do sistema consiste no planeamento racional dos meios de produção de forma a garantir a satisfação da oferta. Pretende-se para além disso, fornecer um suporte adequado para a gestão e controlo operacional.

Com vista a atingir aquele objectivo, o sistema foi concebido e desenvolvido com as seguintes propriedades:

- a) permitir manipular ficheiros de dados contendo informação sobre estradas, horários, autocarros, motoristas e depósitos.
- b) ser suficientemente flexível para que permita um bom interface com os utilizadores, em particular dando ao escalador a inteira possibilidade de intervir no sistema.
- c) ser utilizável noutras funções, particularmente fornecendo informação actualizada para efeitos estatísticos.

O sistema decorre em três fases que correspondem a:

- i) Determinar a dimensão necessária dos meios de produção, e a forma mais racional de os utilizar, de modo a cumprir a oferta de transporte (**PLANEAMENTO**).
- ii) Aplicar e manipular a solução obtida na fase de Planeamento (**GESTÃO**).
- iii) Registrar alterações e analisar desvios na aplicação prática (**CONTROLO**).

Como se ilustra na Fig. 1, as metas estão escalonadas por três fases relacionadas entre si:

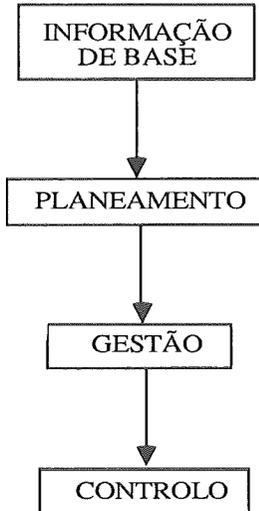


Fig. 1

O Planeamento aparece como a fase em que são definidas as escalas dos autocarros e das tripulações. Baseia-se em modelos de Investigação Operacional e é suportada em microcomputador. Utiliza um sistema de gestão de base de dados para efectuar o processamento da informação que alimenta os dados de I.O. e a manipulação das soluções por eles produzidas e ainda para efectuar o cálculo de estatísticas e a impressão automática de documentos de gestão que suportam a aplicação das soluções obtidas. Na Fig. 2 apresenta-se o esquema correspondente à primeira fase do sistema.

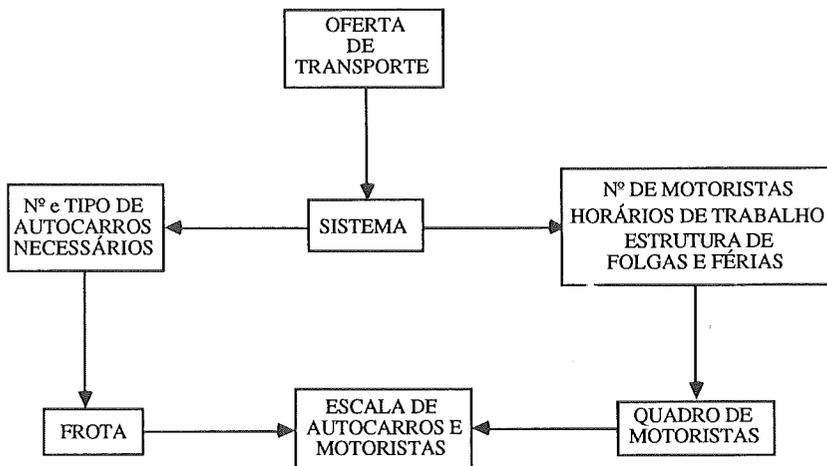


Fig. 2

A Gestão e o Controlo correspondem respectivamente à segunda e terceira fases do sistema. São elas que lhe garantem o dinamismo necessário, ajustando permanentemente as soluções à realidade e fornecendo informação de gestão para a empresa. Estas fases baseiam-se em vários procedimentos e normas de organização, tais como um adequado circuito de informação, fazendo uso dum sistema de base de dados.

### 3. METODOLOGIA

Nesta secção descrevem-se os modelos de I.O. usados na fase de Planeamento. Consistem em três programas integrados de computador cada um dos quais dedicado a uma função específica:

- (i) elaboração de escalas de autocarros;
- (ii) geração dos horários dos motoristas;
- (iii) definição dos dias de folga e férias das tripulações.

A sua identificação e interrelações resumem-se abaixo.

#### 3.1 ELABORAÇÃO DE ESCALAS DE AUTOCARROS

##### 3.1.1 Formalização Matemática

O problema da elaboração de escalas de autocarros consiste em encadear viagens de forma a constituir o serviço individual dos autocarros que operam em dada área. Isto é, dados os horários de um grande número de pequenas viagens, pretende-se construir o serviço dos autocarros que minimiza os custos inerentes à dimensão da frota, aos percursos efectuados em vazio e á utilização de tipologias não adequadas. O serviço dum autocarro corresponde a uma cadeia de viagens que inicia e termina na mesma garagem. Duas viagens consecutivas na mesma cadeia são tais que o local de fim da primeira é o local de início da segunda. Consideram-se, portanto, viagens sem transporte de passageiros.

Pode-se formalizar este problema do seguinte modo. Seja  $I = \{1, 2, \dots, n\}$  um conjunto de índices das viagens a serem operadas no intervalo de planeamento  $T$ ; cada viagem  $i \in I$  é caracterizada pela sua hora e local de início e de fim, linha a que pertence (a empresa opera num determinado conjunto de linhas e incorre em penalidade sempre que na cadeia, duas viagens consecutivas pertencem a linhas diferentes) e pelas tipologias de viaturas que lhe podem ser atribuídas.

Estas características permitem-nos definir uma matriz de custos  $C = [c_{ij}]$  ( $i, j = 1, \dots, n+1$ ) calculada quando se associa um par  $(i, j)$  de viagens, incluindo ainda um custo fixo  $D$  relativo cada autocarro utilizado na escala. Considerando o depósito como a viagem de ordem  $(n+1)$ , as variáveis de decisão são:

$$\begin{array}{ll}
 x_{ij} = & 1, \text{ se a viagem } i \text{ está associada com a viagem } j \\
 i, j \in I & 0, \text{ outros casos} \\
 \\
 x_{n+1, j} = & 1, \text{ se o depósito fornece directamente um autocarro para a viagem } j \\
 j \in I & 0, \text{ outros casos} \\
 \\
 x_{i, n+1} = & 1, \text{ se seguir à viagem } i \text{ o autocarro regressa ao depósito} \\
 i \in I & 0, \text{ outros casos}
 \end{array}$$

$x_{n+1, n+1}$  = número de autocarros que permanecem inactivos no depósito

O problema então será:

$$\min \quad z = \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} c_{ij} x_{ij} \quad (1)$$

$$\text{s.a} \quad \sum_{j=1}^{n+1} x_{ij} \quad i = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=1}^{n+1} x_{ij} = 1 \quad j = 1, 2, \dots, n \quad (3)$$

$$\sum_{j=1}^{n+1} x_{n+1, j} = n \quad (4)$$

$$\sum_{i=1}^{n+1} x_{i, n+1} = n \quad (5)$$

$$x \in \{0, 1\} \quad i, j = 1, 2, \dots, n + 1 \quad (6)$$

A restrição (2) implica que após cada viagem  $i$  o autocarro regressa ao depósito ou inicia outra viagem. A restrição (3) garante que a cada viagem está sempre associado um autocarro, seja a partir do depósito seja a partir de uma outra viagem. As igualdades (4) e (5) significam que os  $n$  autocarros ou permanecem inactivos no depósito ou operam numa cadeia de viagens.

Quando o autocarro após ter realizado a viagem  $i$  pode ser afectado á viagem  $j$ , o custo  $c_{ij}$  define-se:

$$c_{ij} = (K_1 \times L_{ij}) + (K_2 \times T_{ij}) + (K_3 \times V_{ij}) \quad (i, j = 1, \dots, n, i \neq j)$$

onde:

$$L_{ij} = \begin{cases} 0, & \text{se as viagens } i \text{ e } j \text{ pertencem á mesma linha} \\ 1, & \text{outros casos} \end{cases}$$

$$T_{ij} = \text{custo do percurso em vazio necessário para associar } i \text{ e } j$$

$$V_{ij} = \text{penalização por combinar viagens } i \text{ e } j \text{ a que correspondem diferentes tipos de autocarros}$$

e  $K_1, K_2$  e  $K_3$  são parâmetros definidos de acordo com critérios da empresa.

Sempre que a hora de fim da viagem  $i$  mais o tempo de condução necessário para associar  $i$  com  $j$  (calculado com uma margem fixa de segurança) excede a hora de início da viagem  $j$ , considera-se o custo  $c_{ij}$  infinito.

Os custos relativos ás ligações de ou para o depósito fixam-se como segue:

$$c_{i, n+1} = d = D/2 \quad i \in I$$

$$c_{n+1, j} = d = D/2 \quad j \in I$$

$$c_{n+1, n+1} = 0$$

Considerando as viagens ordenadas pelo valor crescente da hora de partida, a matriz de custos vem:

### 3.1.2 Algoritmos

O problema da elaboração de escalas de autocarros apresentado acima é um modelo peculiar de transportes onde todos os segundos membros, excepto dois, são iguais a 1; os dois restantes são iguais a n. Por conseguinte, deve ser usado um algoritmo especial em vez da versão corrente do algoritmo de stepping-stone. Isto foi feito, como se pode ver em [2] e [3], onde se apresenta um método primal especializado, testado para o problema das escalas de autocarros.

Contudo, o programa matemático (1) - (6) pode também ser visto como um modelo de afectação com duas restrições livres. De facto as restrições (4) e (5) são redundantes e o método húngaro pode ser adaptado ao problema da elaboração de escalas de autocarros de tal modo que somente se representam os custos associados ás ligações admíssíveis. Isto foi apresentado pela primeira vez em [4] e [5]. Mais extensões da abordagem do "quasi-assignment" descritas em [6] permitiram a consideração de casos práticos específicos tais como a fixação do número de veículos (incurrendo, eventualmente, em penalidades por viagens não realizadas), ou forçando algumas ligações entre viagens.

### 3.1.3 Resultados Obtidos

Atendendo á leve codificação e às reduzidas necessidades de espaço, o algoritmo de quasi-assignment pode ser facilmente implementado em microcomputador ([7]). Por exemplo, neste momento, na RN, é possível considerar nas escalas de autocarros um máximo de 450 viagens e 14000 associações admíssíveis de viagens num microcomputador IBM/PC-XT de 256Kb o que se tem revelado suficiente para a maioria dos problemas práticos tratados. Contudo, podem vir a fazer-se melhoramentos na codificação FORTRAN o que permitirá tratar dimensões maiores sem prejuízo dos tempos de computação que são já bastante bons como se pode ver pelo Quadro I. No mesmo quadro apresentamos alguns números relativos a aplicações reais tratadas num IBM/PC-XT (primeira parte do quadro I) e num SPERRY HT (parte final), em que ambas as máquinas dispõem de co-processor aritmético 8087. Mais pormenores sobre a experiência computacional em aplicações de escala de autocarros na RN podem ser vistos em [5] e [6].

Nº	PROBLEMA		SOLUÇÃO ÓPTIMA	
	Dimensão	Nº de ligações admíssíveis	Nº de veículos	tempos de computação (seg)
P <sub>1</sub>	210	3028	37	77.77
P <sub>2</sub>	318	4800	44	209.82
P <sub>3</sub>	318	8897	44	328.13
P <sub>4</sub>	318	12735	44	453.80
P <sub>5</sub>	356	3646	48	188.89
P <sub>6</sub>	412	3603	84	227.40
P <sub>7</sub>	258	5272	37	46.36
P <sub>8</sub>	258	11050	26	98.54
P <sub>9</sub>	281	8179	35	63.82
P <sub>10</sub>	356	9534	35	93.43
P <sub>11</sub>	412	2366	84	100.07
P <sub>12</sub>	412	7399	37	108.37

QUADRO I

### 3.2 GERAÇÃO DE HORÁRIOS DOS MOTORISTAS

#### 3.2.1 Formalização Matemática

O problema da elaboração de escalas para os motoristas consiste em determinar o custo mínimo de afectação de motoristas aos autocarros operando numa determinada área. O número de autocarros em movimento e os respectivos serviços são obtidos pela rotina de elaboração de escalas de autocarros e os serviços dos motoristas são determinados de tal forma que a cada autocarro se afecta um motorista e são respeitadas as regras do acordo colectivo de trabalho.

Este problema pode ser formalizado em programação inteira. Seja  $I$  o conjunto de índices das peças de trabalho em que o serviço dos autocarros pode ser partido, isto é, períodos contínuos de serviço em que não são permitidas mudanças de motorista. Um horário admissível de um motorista consiste numa sequência de peças de trabalho que não viole nenhuma das regras do contrato. Seja  $D$  o conjunto de todos os horários admissíveis, para cada um dos quais é calculado um custo de acordo com as regras contractuais dependentes da situação particular considerada. Na RN, o custo de um horário consiste num custo fixo por motorista, adicionado, se for caso disso, de custos relativos a horas extraordinárias, a trabalho nocturno e a refeições.

Podemos agora formalizar o problema em programação inteira, como segue:

$$\min \sum_{j \in D} c_j x_j \tag{7}$$

$$\text{s.a} \sum_{j \in D} a_{ij} x_j \geq 1, \quad i \in I \tag{8}$$

$$x_j \in \{0, 1\}, \quad j \in D \tag{9}$$

onde,

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

$$a_{ij} = \begin{cases} 1, & \text{se a peça } i \text{ de trabalho é coberta pelo horário } j \\ 0, & \text{caso contrário} \end{cases}$$

O problema acima apresentado é bem conhecido na literatura com o nome de "Set Covering Problem" (SCP) o qual, neste caso, é de grande dimensão na maior parte das situações práticas. De facto o número de linhas correspondentes ao número total de peças de trabalho para todos os autocarros e o número de colunas, atinge os milhares mesmo para um reduzido número de autocarros.

Apesar dos recentes melhoramentos introduzidos nos algoritmos do SCP, o problema associado às aplicações das escalas de motoristas só se resolve com um significativo esforço computacional. É possível, contudo, obter aproximações razoáveis considerando um modelo agregado definido como segue:

$$\min \quad z = \sum_{j \in D} c_j x_j \tag{10}$$

$$\text{s.a} \quad \sum_{j \in D} d_{tj} x_j \geq b_t, \quad t \in T \tag{11}$$

$$x_j \geq 0 \text{ e inteiro}, j \in D \quad (12)$$

Onde T é o conjunto de intervalos de tempo em que o período em estudo está dividido;  $b_t$  é o número de autocarros em circulação no intervalo de tempo  $t \in T$ . D estabelece o conjunto das combinações dos períodos de tempo que respeitam as regras contratuais. Os custos dos horários são calculados como anteriormente e

$$d_{ij} = \begin{cases} 1, & \text{se o período } t \text{ está coberta pelo horário } j \\ 0, & \text{caso contrário} \end{cases}$$

O modelo agregado é um modelo de programação inteira conhecido normalmente por "set covering problem" generalizado ([8]).

### 3.2.2 Algoritmos

Embora o conjunto D do modelo agregado seja muito mais simples de gerar que o anterior D, ele continua a ser muito grande na maior parte dos casos reais. Além disso, segundo a nossa experiência, é computacionalmente mais económico gerar as colunas (horários) como e quando necessárias e dar ao escalador operacional a possibilidade de utilizar a sua própria experiência. Dai que, para a escala de um dia de trabalho se considere primeiro um subconjunto  $D_0$  consistindo em:

- i) horários de período único
- ii) horários de dois períodos com início no princípio do dia
- iii) horários de dois períodos a terminarem tarde
- iv) horários com certos atractivos ditados pela experiência

O problema restrito que resulta da utilização do conjunto  $D_0$  no modelo agregado é aproximado, ou solucionado, por heurísticas greedy e programação linear. As heurísticas greedy ([8]) fornecem uma solução admissível ao problema restrito e, obviamente, ao modelo agregado. Então, consideramos somente os horários seleccionados pela heurística greedy, digamos  $D_1 \subset D_0$ , e a programação linear é usada para calcular uma solução "agregada" admissível (próxima da solução linear), bem como uma solução dual restrita. Esta última, não é necessariamente admissível para o modelo agregado visto que algumas das colunas não consideradas podem ter um custo reduzido negativo relativamente à solução dual mencionada.

Dai que, se procurem horários  $j \in D - D_1$  tais que:

$$s_j = c_j - \sum_{t \in T} d_{tj} u_t^0 < 0$$

Onde  $u_t^0$ , ( $t \in T$ ), é a solução dual admissível obtida para o problema t restrito. Essas colunas podem ser incluídas em  $D_0$  e o correspondente problema restrito é de novo considerado, com as heurísticas greedy à procura de melhores soluções admissíveis e a programação linear elaborando um novo par de soluções primal e dual.

Procurar todas as variáveis com  $s_j < 0$  pode corresponder a perder muito tempo, e será mais razoável uma vez mais considerar apenas as mais atractivas. Contudo, é dada possibilidade ao utilizador de gerar todas as variáveis com custo reduzido negativo para  $u_0$ . O processo continua até que seja encontrada uma solução satisfatória para o modelo agregado.

Convém acentuar que, em geral, esta solução é não admissível para o problema "set covering" original. Por isso, são necessários alguns ajustamentos para a tornar admissível dum ponto de vista operacional aumentando significativamente o custo da solução. Estes

ajustamentos anteriormente eram feitos à mão mas, neste momento, já se dispõe de um procedimento em computador que apoia esta fase da elaboração da escala do pessoal tripulante.

### 3.2.3 Resultados Obtidos

Conclui-se da exposição anterior, que o tempo gasto a resolver ou aproximar cada exemplo do problema restrito é significativo. Na realidade, para cada aplicação prática é necessário efectuar um número razoável de corridas de computador, facto que se ilustra no Quadro II onde se apresentam os resultados de computador relativos a aplicações reais do procedimento acima descrito.

	Nº de horários consider.	Nº de motoristas	custo	tempo de comput.
Heur. Greedy	372	28	99024.	134.23
Prog. Linear	53	28	97124.	23.07
Análise dos c. reduzidos	+1			
Prog. Linear	54	28	96700.	
Análise dos c. reduzidos	+6			
Heur. Greedy	378	28	98120.	114.15
Prog. Linear	54	28	96326.	18.95
Análise dos c. reduzidos	+0			
Solução Final		28	108615.	

Quadro II

Pela leitura do Quadro apresentado acima, podemos ver, em primeiro lugar, que as heurísticas Greedy são usadas para encontrar soluções admissíveis para o modelo agregado de entre um total de 372 horários. A melhor destas soluções envolve 28 motoristas com um custo de 99024 escudos. Dos 372 horários possíveis são seleccionados 53 para a Programação Linear que por sua vez dá uma solução admissível de custo inferior - 97124 escudos - para o mesmo número de motoristas. A análise de custos reduzidos produz um horário possível adicional que leva a um novo melhoramento no custo - 96700 escudos. Uma pesquisa mais exaustiva dos horários com custos reduzidos negativos leva-nos a considerar mais seis horários. As heurísticas, chamadas pela segunda vez, encontram uma solução de 98120 escudos e a Programação Linear (com 54 horários possíveis) dá a solução óptima para o modelo agregado: 28 motoristas com um custo de 96326 escudos. Esta solução, bem como todas as anteriores, foram tidas em conta para os ajustamentos manuais com vista à sua admissibilidade prática conduzindo assim a uma solução final que envolve o mesmo número de motoristas. Embora o custo desta tenha aumentado significativamente (108615 escudos), ficou demonstrado que a solução final obtida pelo sistema computadorizado é mais barata que a obtida de forma tradicional. O processo de a obter é definitivamente muito mais fácil do que anteriormente.

Os tempos de computação que se apresentam no Quadro II referem-se a um IBM/PC-XT e demonstraram uma estabilidade razoável em problemas de dimensão semelhante à do problema apresentado no Quadro. Actualmente, podemos dizer que o procedimento descrito nesta secção conduziu a reduções significativas quer em tempo quer no esforço dedicado à elaboração das escalas de pessoal tripulante na RN. Além disso, obtiveram-se também

poupanças significativas relativamente às soluções tradicionais conforme se ilustra na última secção desta comunicação.

### 3.3 ESTRUTURA DE FOLGAS E FÉRIAS

#### 3.3.1 Formalização Matemática

O modelo que se descreve nesta secção visa o problema da determinação do número total de motoristas necessários para garantir a realização dos serviços diários definidos pelo modelo da secção anterior. Para além disso, determina a correspondente estrutura de folgas e de férias.

O modelo considera que:

- Cada trabalhador tem dois dias consecutivos de folga por semana e um mês seguido de férias por ano.
- Uma semana tem sete dias tipo, cada dia eventualmente com diferentes necessidades de pessoal.
- Cada mês tem quatro semanas.

De acordo com o exposto acima, resolve-se o seguinte problema de programação linear:

$$\min \sum_{i=1}^7 \sum_{j=1}^{12} x_{ij} \quad (13)$$

$$\text{s.a.} \sum_{i=1}^7 \sum_{j=1}^{12} a_{i,j,k,l} x_{ij} \geq b_{k,l}, \quad (k=1, 2, \dots, 7) \quad (14)$$

$$x_{ij} \geq 0 \quad (15)$$

onde:

$x_{ij}$  é o número de trabalhadores com folgas do tipo  $i$  e férias no mês  $j$   
( $i=1, \dots, 7$ ;  $j=1, \dots, 12$ )

$a_{i,j,k,l} =$   
1, se o trabalhador com dia de folga do tipo  $i$  e mês de férias  $j$  estão de serviço no dia de semana  $k$  do mês  $l$   
0, caso contrário

$b_{k,l}$  é o número de trabalhadores necessários ao serviço no dia de semana  $k$  do mês  $l$ .

O principal inconveniente desta formalização é a necessidade de proceder a ajustamentos da solução óptima para uma solução inteira. Contudo, um técnico experiente não dispense mais de meia hora para o fazer usando um pequeno programa desenvolvido, para o efeito, em microcomputador.

## 4. ESTRATÉGIA DE IMPLEMENTAÇÃO

No início do processo de implementação, no final dos anos 70, o sistema estava ainda na primeira fase de desenvolvimento e era computacionalmente suportado em mainframe.

Dadas as dificuldades em usar um mainframe destinado a muitas outras tarefas na empresa, o planeamento automático das escalas era tarefa do departamento de I. O. deixando para os escaladores tradicionais apenas a tarefa de ajustamentos manuais das soluções obtidas. Entretanto, a aplicação de tal sistema tornara-se muito maçadora com longos tempos de espera sempre que havia necessidade de alterar os parâmetros de planeamento. Estas razões

contribuíram para que surgisse alguma resistência em usar computadores por parte dos operadores tradicionais, os quais, conhecendo perfeitamente a complexidade das tarefas envolvidas neste tipo de trabalho, duvidavam naturalmente da possibilidade de ele ser executado por um programa de computador sem que fossem negligenciadas regras contratuais ou operacionais.

A entrada em cena dos microcomputadores, em 1984, veio mudar este estado de coisas drasticamente. De facto tornou-se possível o uso descentralizado do sistema e daí a sua aplicação generalizada na empresa. O curto tempo de resposta do computador e a possibilidade de testar alternativas até se considerar satisfeito, conquistaram mesmo os antigos críticos do sistema.

Simultaneamente, desenvolveu-se um subsistema interactivo de gestão de bases de dados que permite criar, manipular e actualizar os ficheiros principais e além destas operações permite outras, tais como:

- modificar os serviços a efectuar pelos autocarros e/ou pelos motoristas;
- calcular todo o tipo de estatísticas relacionadas com a exploração operacional e salários;
- imprimir todo o tipo de documento relacionado com a escala.

Finalmente, a introdução na empresa dum novo processo de trabalho necessitou do desenvolvimento doutro conjunto de actividades relacionadas com:

**PSICOLOGIA** - Desenvolveram-se acções psicológicas, particularmente para demover algumas suspeitas e/ou resistências em alterar os métodos tradicionais de trabalho nos centros operacionais.

**FORMAÇÃO** -. Preparou-se um programa de formação tendo em conta os diferentes níveis hierárquicos e diferentes graus de participação no sistema. Até à data, já decorreram muitas acções de formação tendo sido preparada e distribuída documentação de apoio.

**ESTRUTURA** - A actividade de Organização não foi omitida. De facto, houve a preocupação de analisar todas as funções e de encontrar a estrutura adequada em cada centro operacional.

**SISTEMA DE INFORMAÇÃO DE GESTÃO** - Está também relacionado com a actividade de Organização. De facto há necessidade de obter estatísticas que permitam um controlo operacional actualizado e de fazer a ligação ao Sistema de Informação de Gestão da Empresa.

## **5. RESULTADOS OBTIDOS**

Neste momento, o nível de aplicações do sistema ronda os 50% e em grande parte das aplicações foi possível obter poupanças anuais significativas, especialmente em Lisboa e arredores conforme se apresenta no Quadro III.

Refira-se que, em muitas aplicações, se tem revelado difícil calcular as poupanças anuais obtidas com a introdução do sistema de elaboração automática de escalas por falta de informação referente ao passado. Mas, muito embora as poupanças sejam o principal atractivo para os gestores ao adoptarem o sistema, este pode proporcionar outros melhoramentos tais como maior rapidez, informação mais conveniente e melhor controlo das capacidades.

(preços correntes)

Área	Ano	Nº de Autocarros	Nº de Motoristas	Poupanças Anuais (contos)	
					%
Massamá	1982	9	13	600	9%
Brandoa	1983	15	20	1280	?
Montijo	1984	29	43	12000	31%
Amadora	1984	28	38	1007	10%
Cascais	1985	47	56	6000	?
Barreiro	1985	47	59	4950	11%
Laranjeiro	1986	85	118	8700	7%
Paio Pires	1986	68	82	3100	3%
C. Caparica	1987	60	81	5800	6%

QUADRO III

Por fim, podemos dizer que, com a estratégia seguida para a implementação do sistema, os técnicos envolvidos nas funções de elaboração de escalas se sentem entusiasmados com a utilização do computador e sabem que a sua própria experiência contribui extremamente para o bom êxito do processo.

### REFERÊNCIAS

- [1] Christofides, N. and Paixão, J., "Algorithms for Large Scale Set Covering Problems", comunicação apresentada ao 12<sup>th</sup> International Symposium on Mathematical Programming, Boston, USA, (1985).
- [2] Gavish, B., Schewitzer, P. and Shlifer, E., "Assigning Buses to Schedules in Metropolitan Areas", *Computers and Operations Research*, vol. 5, (1978).
- [3] Gavish, B. and Shlifer, E., "An Approach for Solving a Class of Transportation Scheduling Problems", *European Journal of Operational Research*, vol. 3, (1978).
- [4] Paixão, J. e Branco, I. M., "A New Assignment Algorithm for Bus Scheduling in an Urban Area", Nota Nº 9 do CEAUL, Faculdade de Ciências de Lisboa, paper apresentado ao Euro VII, Bolonha, Itália, (1985).
- [5] Paixão, J. e Branco, I. M., "A Quasi-Assignment Algorithm for Bus Scheduling", *Networks*, vol. 17 (3), (1987).
- [6] Paixão, J. e Branco, I. M., "Bus Scheduling with a Fixed Number of Vehicles", paper apresentado a Fourth Internacional Workshop on Computer-Aided Scheduling of Public Transport, Hamburgo, RFA, (1987).
- [7] Paixão, J., Branco, I. M., Captivo, E., Pato, M. V., Eusébio, R., Amado, L., "Bus and Crew Scheduling on a Microcomputer", *Proceedings of the Conference "O. R. Models on Microcomputers"*, Lisboa, Portugal, (1985). Editado por Coelho, J. D. e Tavares, L. V., North-Holland (1986).
- [8] Paixão, J. e Pato, M. V., "Greedy Heuristics for the Generalized Set Covering Problem", Nota Nº 8/85 do CEAUL, Faculdade de Ciências de Lisboa, (1985).
- [9] Romão Eusébio, J. A. e Amado, L., "Modelos de Planeamento/Gestão para Optimização de Viaturas e Tripulações numa Rede de Transportes", paper apresentado ao II Congresso da APDIO, Porto, Portugal, (1984).
- [10] Wren, A., "Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling", North-Holland Publishing Co., (1981).

## SISTEMA INTERACTIVO PARA GERAÇÃO DOS HORÁRIOS DE CONDUTORES DE VIATURAS

Ana Maria Paias  
D.E.I.O.C.  
Faculdade de Ciências de Lisboa

### Resumo

Na presente comunicação apresenta-se um método heurístico para a resolução do problema da determinação dos horários de condutores de viaturas.

Este problema pode ser formulado como um Modelo de Cobertura de Custo Mínimo de um Conjunto. No entanto em casos reais, tais como os tratados na RODOVIARIA NACIONAL (RN), esta formulação conduz a problemas com dimensão bastante elevada para serem resolvidos através de métodos exactos.

O modo como se particiona o serviço a realizar por cada viatura é um dos factores que mais contribui para a obtenção de dimensões tão elevadas.

O método implementado permite ao utilizador escolher a partição inicial para as peças de trabalho, entrando em conta com a sua experiência bem como com alguns requisitos operacionais.

Os horários admissíveis são então gerados, eliminando-se alguns efectuando testes de dominancia de custos. Usando um conjunto de Heurísticas rápidas, obtém-se uma solução admissível para o problema de Cobertura.

Como a solução obtida depende da partição que foi escolhida inicialmente, permite-se gerar sucessivamente novas combinações a partir de outras partições, mas de forma a aproveitar os horários já obtidos, que cobrem peças não alteradas.

### Abstract

This paper reports on the implementation of a set covering based heuristic for bus driver scheduling at RODOVIARIA NACIONAL (RN).

In fact, the bus driver scheduling problem can be formulated as a minimum cost set covering problem but, for many real applications (like the ones dealt with at RN), this becomes too large for solving by exact methods. Actually, the number of pieces of work is the crucial factor for yielding such large size SCPs.

Hence, the system now implemented at RN allows the user to define an initial partition for the pieces of work which can be dictated by experience or some operational requirements. Then, all possible work schedules are generated, some of them being ruled out by cost dominance tests. Fast heuristics are used to obtain feasible solutions for the corresponding set covering problem. Then, a new partitioning for the pieces of work can be produced and the process is resumed in such way that schedules previously obtained covering unchanged pieces of work are maintained. This process is repeated until a satisfactory solution is obtained.

**KEYWORDS:** bus driver duties, set covering, greedy heuristics.

### 1. INTRODUÇÃO

O problema da determinação dos serviços de condutores de viaturas tem uma importância relevante para empresas transportadoras, em particular a RODOVIARIA NACIONAL (RN), onde se efectuou este estudo.

Conhecido os percursos a realizar por um conjunto de viaturas, pretende-se determinar a afectação de custo mínimo de motoristas a essas viaturas, tendo em conta que cada viatura fica afectada a um motorista e que todas as cláusulas da contratação colectiva devem ser respeitadas.

## 2. DEFINIÇÕES e FORMULAÇÃO - Modelo de Cobertura (MC)

Sejam  $B = \{b_1, b_2, \dots, b_{|B|}\}$  o conjunto das viaturas a considerar no período de planeamento e  $T = \{t_1, t_2, \dots, t_{|T|}\}$  o conjunto de horários referentes a essas mesmas viaturas.

Uma **peça de trabalho** consiste num conjunto de uma ou mais viagens simples, e é entendida como um período contínuo de condução. Define-se, então, como sendo o quádruplo  $(t_i, t_f, p_i, p_f)$  em que:

$t_i$  - instante em que se inicia  
 $t_f$  - instante em que termina  
 $p_i$  - ponto ou local onde se inicia  
 $p_f$  - ponto ou local onde termina

O **percurso** a efectuar por uma viatura é identificado pelo conjunto de peças de trabalho que lhe é atribuído; o percurso a efectuar pela viatura  $\ell$  representa-se por:

$$T_\ell = \left\{ \left( t_{\ell}^1, t_{\ell}^1, p_{\ell}^1, p_{\ell}^1 \right), \left( t_{\ell}^2, t_{\ell}^2, p_{\ell}^2, p_{\ell}^2 \right), \dots, \left( t_{\ell}^{n(\ell)}, t_{\ell}^{n(\ell)}, p_{\ell}^{n(\ell)}, p_{\ell}^{n(\ell)} \right) \right\}$$

em que  $(t_{\ell}^k, t_{\ell}^k, p_{\ell}^k, p_{\ell}^k)$  é a  $k$ -ésima peça de trabalho a realizar pela  $\ell$ -ésima viatura ( $\ell = 1, \dots, |B|$ ;  $k = 1, \dots, n(\ell)$ ), sendo  $n(\ell)$  o número de peças de trabalho a realizar pela  $\ell$ -ésima viatura).

Chama-se **serviço** a realizar por um condutor, ou simplesmente **serviço**, a uma combinação de peças de trabalho. Um serviço diz-se admissível se respeitar as normas de contratuais (por exemplo, número de horas extraordinárias, tempo normal de serviço, etc).

Seja  $D_j$ ,  $j \in N$  ( $N$  é o conjunto de serviços admissíveis).  $D_j$  pode-se representar na forma seguinte:

$$\left\{ d_{k_1 \ell_1}^1, d_{k_2 \ell_2}^2, \dots, d_{k_m(j) \ell_m(j)}^{m(j)} \right\}$$

em que  $d_{k_p \ell_p}^{k_p} = (t_{\ell_p}^{k_p}, t_{\ell_p}^{k_p}, p_{\ell_p}^{k_p}, p_{\ell_p}^{k_p})$  com  $1 \leq p \leq m(j)$ ,  $1 \leq \ell_p \leq |B|$  e  $1 \leq k_p \leq n(\ell_p)$

ou seja, a  $p$ -ésima componente de  $D_j$ , corresponde à  $k$ -ésima peça de trabalho a ser realizada pela  $\ell$ -ésima viatura.

As componentes de um mesmo serviço têm, então, de verificar o seguinte :

$$t_{\ell_p}^{k_{p+1}} \geq t_{\ell_p}^{k_p} + \text{dist} \left( p_{\ell_p}^{k_p}, p_{\ell_{p+1}}^{k_{p+1}} \right) \quad (1)$$

ou seja, o instante em que se inicia a  $(p+1)$ -ésima componente de um serviço, tem de ser superior ao instante em que termina a  $p$ -ésima componente considerando adicionalmente o tempo de deslocação do local onde aquela termina para o local onde se inicia a componente seguinte.

A cada serviço  $D_j$  vai estar associado um custo  $c_j$  que é calculado a partir de regras que dependem da situação em estudo ( por exemplo, tempo extraordinário diurno e nocturno, etc..).

Se se considerar

$$a_{k \ell}^j = \begin{cases} 1 & \text{se } \left\{ t_{\ell}^{k \ell}, t_{\ell}^{k \ell}, p_{\ell}^{k \ell}, p_{\ell}^{k \ell} \right\} \in D_j \\ 0 & \text{caso contrário} \end{cases}$$

e se se considerarem as variáveis  $x_j$  ( $j \in N$ ) tais que:

$$x_j = \begin{cases} 1 & \text{se } D_j \text{ é seleccionado} \\ 0 & \text{caso contrário} \end{cases}$$

então o problema pode ser formulado do seguinte modo:

$$\begin{aligned}
 \text{(MC) minimizar} \quad & \sum_{j \in N} c_j x_j \\
 \text{s.a:} \quad & \sum_{j \in N} a_{k\lambda} x_j \geq 1 \quad \begin{array}{l} \lambda = 1, \dots, |B| \\ k = 1, \dots, n(\lambda) \end{array} \\
 & x_j \in \{0,1\} \quad j \in N
 \end{aligned} \quad (2)$$

em que  $N = \{\text{serviços admissíveis}\}$ .

As restrições (2), (cada uma correspondente a uma peça de trabalho), garantem que cada peça de trabalho é coberta por pelo menos um serviço.

O modelo (MC) é um modelo de Cobertura de Custo Mínimo de um Conjunto ([3]), bastante conhecido da literatura e que tem várias aplicações (por exemplo: recuperação de informação de ficheiros, distribuição de produtos, problemas de coloração, etc). O referido modelo matemático revela, também, algumas propriedades que lhe permitem ser tratado por vários tipos de métodos (por exemplo: planos de corte, programação linear, enumeração, etc). Contudo qualquer deste tipo de métodos é bastante sensível ao aumento de dimensão, tornando-se proibitiva a sua utilização para casos de dimensão elevada.

Infelizmente é o que se passa o nosso caso uma vez que o número de peças de trabalho e de serviços admissíveis é, em geral, bastante elevado.

### 3. RESOLUÇÃO do PROBLEMA MC - Algoritmos

Como se referiu atrás, as dimensões atingidas em problemas associados a situações reais são muito elevadas. De facto, o número de linhas do (MC) é dado pelo número de peças de trabalho e o número de colunas corresponde ao número de serviços admissíveis. Surgem, então, dois factores importantes que respeitam ao modo como se definem as peças de trabalho e à forma como se efectua a geração dos serviços admissíveis. Estes últimos não são mais do que combinações de peças de trabalho que verificam as normas contratuais e a restrição (1) mencionada no ponto anterior.

Uma primeira diminuição do número de serviços a considerar na formulação inicial pode ser realizada logo no processo de geração dos serviços admissíveis, eliminando-se aqueles que à partida se sabe, garantidamente, que não vão figurar na solução óptima. Uma referência detalhada a essas eliminações pode ser encontrada em [3] e [4]. Quanto ao processo de geração dos serviços é descrito pelo algoritmo ALG1, que se apresenta de seguida.

Antes, porém, convém estabelecer alguma notação. Assim, sejam :

$$\begin{aligned}
 P &= \{\lambda: \lambda \text{ é peça de trabalho}\} \\
 C_i &= \{j: j \text{ é combinação admissível de peças de trabalho}\} \\
 k \rightarrow j &\text{ se a peça de trabalho } j \text{ é realizada imediatamente a seguir à} \\
 &\text{combinação de peças } k \\
 0 \rightarrow j &\text{ se a peça de trabalho } j \text{ é a primeira a ser realizada} \\
 C &= \{j: j \text{ pertence a algum } C_i\} = \{\text{serviços admissíveis}\}
 \end{aligned}$$

#### ALGORITMO 1 (ALG1)

Passo 0 (Inicialização)

$$C_0 = \{(0 \rightarrow p): p \in P\}$$

$$\text{custo}(0 \rightarrow p) = \text{CFIXO} + f(0 \rightarrow p), \forall p \in P$$

Passo 1 (Iteração k)

para todo o  $j \in C_{k-1}$  e para todo o  $\lambda \in P$  fazer

se  $(j \rightarrow \lambda)$  é admissível então  $(j \rightarrow \lambda) \in C_k$

$$\text{custo}(j \rightarrow \lambda) = \text{CFIXO} + f(j \rightarrow \lambda)$$

Passo 2 (Critério de Paragem)

se  $C_k = \emptyset$  então  $C = \bigcup C_i (i=1, \dots, k-1)$  e STOP

Passo 3 (Eliminação k-1)

para todo o  $j \in C_k$  e para todo o  $j' \in C_{k-1}$  fazer

se custo(j')=custo(j) e  $j' \subset j$   
 então  $C_{k-1} = C_{k-1} - \{k\}$   
 $k = k + 1$

Volta ao passo 1.

Note-se que, no algoritmo ALG1,  $C = \bigcup C_i$  ( $i=1,2,\dots,k$ ) e que se  $j$  pertence a algum  $C_i$  então é da forma

$$0 \rightarrow l_1 \rightarrow l_2 \rightarrow \dots \rightarrow l_{k(j)}$$

em que cada  $l_j \in P$ . Para  $k$  e  $j \in C$  considera-se que  $k \subset j$  (Passo 3), sse para toda a peça de trabalho que constitui  $k$  também está incluída em  $j$ .

Acrescente-se, ainda, que custo(i) é o custo associado à combinação de peças  $i$  e definido como sendo igual a um valor fixo, CFIXO ( correspondente a um custo fixo por equipa ), mais um custo variável  $f(i)$  dependente da situação em estudo, o qual é calculado em função dos seguintes parâmetros: período extraordinário diurno, período extraordinário nocturno, período nocturno em período normal de serviço, etc. Assim, tem-se que  $\text{custo}(i) = \text{CFIXO} + f(i)$ .

Após a geração dos serviços pelo ALG1, determina-se uma solução admissível para MC usando um conjunto de heurísticas descritas por Balas & Ho [1] e implementadas em [2]. Trata-se de heurísticas do tipo greedy, ou seja, métodos que procedem à construção de uma solução admissível para o problema numa sequência de passos. Cada um destes consiste na selecção de um serviço  $D_j$  ( $x_j = 1$ ), que minimize uma função dos coeficientes não nulos associados a  $x_j$ . As várias heurísticas usadas diferem na escolha da chamada função de selecção, como pode ser estudado com detalhe nos trabalhos acima mencionados. Obtém-se, ainda, um limite inferior para o valor óptimo do problema através da utilização de uma outra heurística, tipo dual, também descrita em [1].

A solução obtida por este processo depende fortemente do modo como são definidas as peças de trabalho. Este facto explica o interesse em experimentar novas formas de dividir o trabalho a realizar pelas viaturas de modo a tentar obter uma solução melhor.

Note-se que os vários problemas resultantes das várias possibilidades de definição dos conjuntos de peças de trabalho, podem ser considerados, todos eles, como aproximações do problema obtido quando se faz corresponder uma peça de trabalho a cada uma das viagens simples. Neste, que designamos por problema principal, cada linha corresponde a uma viagem simples. Cada aproximação corresponde, então, à resolução do problema Principal mas considerando apenas um subconjunto de colunas. Este facto é ilustrado no esquema seguinte:

		serviços				
1	1...1...1	0...0...0	1...1...1	0...0...0		
2	0...0...0	1...1...1	1...1...1	0...0...0		
3	...	...	...	...		viagens simples
4	...	...	...	...		
...						
n						

As linhas representam viagens simples e as colunas correspondem aos serviços admissíveis.

1	1	0	1	0
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1

As colunas da matriz podem, para o exemplo em causa, ser divididas nos quatro tipos acima representados. Ora, se se definir uma peça de trabalho como constituída pelas duas primeiras viagens simples, tal corresponderá não só a eliminar todas as colunas dos tipos

1	0
0	1
1	1
1	1

mas também a eliminar a linha 1 ou 2, pois uma coluna ou cobre ambas as linhas ou não cobre nenhuma. Daí a razão de se tentarem várias aproximações (várias definições de peças de trabalho) no sentido de se obter uma melhor solução para o problema Principal.

**ALGORITMO 2 (ALG2)**

Seja  $P=P_1 \cup P_2$  com  $P_1=\{p \in P: p \text{ não foi alterada}\}$  e  $P_2=\{p \in P: p \text{ foi alterada}\}$ , e ainda  $NA=\{j \in C: j \text{ é formada apenas por peças de trabalho não alteradas, ou seja } j \in P_1\}$

Passo 0 (Inicialização)

$k \leftarrow 1$

$C_0^1 \leftarrow NA \cup \{(0 \rightarrow p): p \in P_1\}$

$C_0^2 \leftarrow \{(0 \rightarrow p): p \in P_2\}$

$C_0 \leftarrow C_0^1 \cup C_0^2$

Passo 1 (Iteração 1)

Para todo o  $j \in C_0$  e todo o  $l \in P_2$  fazer

se  $(j \rightarrow l)$  é admissível então  $(j \rightarrow l) \in C_1$

custo( $j \rightarrow l$ ) = CFIXO + f( $j \rightarrow l$ )

Para todo o  $j \in C_0^2$  e todo o  $l \in C_0^1$

se  $(j \rightarrow l)$  é admissível então  $(j \rightarrow l) \in C_1$  (\*)

custo( $j \rightarrow l$ ) = CFIXO + f( $j \rightarrow l$ )

Segue para o passo 3.

Passo 2 (Iteração k)

Para todo o  $j \in C_{k-1}$  e todo o  $l \in P$  fazer

se  $(j \rightarrow l)$  é admissível então  $(j \rightarrow l) \in C_k$

custo( $j \rightarrow l$ )  $\leftarrow$  CFIXO + f( $j \rightarrow l$ )

Passo 3 (Critério de Paragem)

Se  $C_k = \bar{\emptyset}$  então  $C = \bigcup C_i$  ( $i=1, \dots, k-1$ ) e STOP

Passo 4 (Eliminação k-1)

Para todo o  $j \in C_k$  e todo o  $j' \in C_{k-1}$  se custo( $j'$ ) = custo( $j$ ) e  $j' \subset j$   
então  $C_{k-1} \leftarrow C_{k-1} - \{k\}$

$k \leftarrow k+1$  e volta ao passo 2.

(\*) aqui faz-se um abuso de linguagem quando se considera  $l$  uma combinação em vez de uma peça simples.

O processo de alteração da divisão dos serviços das viaturas requer um esforço computacional considerável, uma vez que envolve um novo processo de geração dos serviços. Assim, será de se aproveitar todos os serviços constituídos apenas por combinações de peças de trabalho não alteradas. Os restantes serviços terão, necessariamente, de ser gerados. A geração dos serviços admissíveis, a partir de uma definição de peças de trabalho  $P$  é feita de acordo com o algoritmo ALG2, descrito na página anterior. Em seguida procura-se uma nova solução admissível para o problema recorrendo-se aos métodos heurísticos já mencionados, [2].

Um processo alternativo para limitar o número de serviços a serem gerados, consiste em estabelecer-se 'padrões' para estes. Isto é, só se geram as combinações admissíveis de peças de trabalho que se enquadrem num conjunto de horários, no nosso caso definidos por um modelo agregado, [6]. Este processo está descrito no algoritmo ALG3, correndo-se, no entanto, o risco de os serviços gerados não cobrirem todas as peças de trabalho. Tal depende, por um lado, dos horários-padrão escolhidos e, por outro, das peças seleccionadas. No primeiro caso, há que proceder a algumas tentativas até se atingir a garantia de cobertura. No segundo caso, pode-se adaptar o processo descrito pelo algoritmo ALG2, constituindo-se um novo algoritmo, ALG4, também descrito a seguir.

**ALGORITMO 3 (ALG3)**

Sejam  $H=\{h: h \text{ é serviço}\}$  (supõe-se uma enumeração dos serviços)

$PE=\{p: p \text{ é peça de trabalho}\}$

$P_h=\{p \in PE: p \text{ é coberta pelo serviço } h\}$

$C^h=\{j: j \text{ é combinação de peças de trabalho coberta pelo serviço } h\}$

$CH = \{j: j \text{ é combinação de peças de trabalho coberta por pelo menos um serviço}\}$   
= $\{ \text{serviços admissíveis gerados a partir do conjunto } H \}$

Passo 0  $k \leftarrow 1$

Passo 1 Para todo o  $p \in PE$ , se o serviço  $h$  cobre  $p$  então  $p \in P_h$

Passo 2 Aplicar algoritmo ALG1 com  $P = P_h$

Passo 3  $Ch=C$

Se  $h=|T|$  então  $CH = \bigcup Ch (h=1, \dots, |T|)$  e STOP

$h < h+1$

Segue para o passo 1

#### ALGORITMO 4(ALG4)

Sejam  $PE=PE_1 \cup PE_2$  com  $PE_1 = \{p \in PE: p \text{ não alterada}\}$  e  $PE_2 = \{p \in PE: p \text{ foi alterada}\}$

$NAL = \{j \in C: j \text{ é formada apenas por peças de trabalho não alteradas}\}$

$P_h^1 = \{p \in P_h: p \in PE_1\}$ ,  $P_h^2 = \{p \in P_h: p \in PE_2\}$

$NAL^h = \{j: j \text{ é combinação de peças não alteradas coberta pelo serviço } h\}$

Passo 0  $h < -1$

Passo 1 para todo o  $p \in PE_1$ , se  $p$  é coberta pelo serviço  $h$  então  $p \in P_h^1$

para todo o  $p \in PE_2$ , se  $p$  é coberta pelo serviço  $h$  então  $p \in P_h^2$

para todo o  $l \in NAL$ , se  $l$  é coberta pelo serviço  $h$  então  $h \in NAL^h$

Passo 2 Aplicar ALG2 para  $NA = NAL^h$ ,  $P_1 = P_h^1$  e  $P_2 = P_h^2$

Passo 3  $Ch < C$

Se  $h=|H|$  então  $CH = \bigcup Ch (h=1, \dots, |T|)$  e STOP

$h < h+1$

Segue para o passo 1

Para efeitos de realização de testes computacionais, foram elaborados dois programas, Chapas1 e Chapas2, [4]. No primeiro são determinadas duas soluções para MC, aplicando o método heurístico aos conjuntos de serviços admissíveis gerados pelos algoritmos ALG1 e ALG3, para o mesmo conjunto de peças de trabalho. O segundo programa actua de forma semelhante, mas os conjuntos de serviços admissíveis são gerados usando os algoritmos ALG2 e ALG4, após se ter definido um novo conjunto de peças de trabalho.

Note-se que em ambos os programas o conjunto de peças de trabalho é escolhido pelo próprio utilizador por meio de um menu, o que permite não só entrar em conta com a experiência do utilizador como também com alguns requisitos operacionais (ex: locais ou pontos de transferência de condutores).

#### 4. EXPERIÊNCIA

Nesta secção, apresentam-se alguns resultados referentes a aplicações reais realizadas na Rodoviária Nacional considerando-se o problema formulado como modelo de cobertura. Trata-se de casos reais respeitantes às áreas de operação de Carcavelos, Figueira da Foz, Palmela, Queluz de Baixo e Torres Vedras que podemos considerar como casos típicos da RN.

No primeiro quadro ilustram-se, para o exemplo 3, as variações produzidas escolhendo diferentes conjuntos de peças de trabalho embora mantendo o mesmo número destas. Ou seja, considerando diferentes partições do conjunto das viagens simples.

Os valores constantes da coluna 'LIMITE INF', correspondem aos limites inferiores produzidos, para cada um dos exemplos, por um procedimento heurístico aplicado ao dual da relaxação linear. Assim, estes limites dependem do conjunto de peças de trabalho escolhido.

exemplo 3:

GERADOS	ELIMINADOS	NºPEÇAS	NºEQUIPAS	CUSTO	LIMITE INF
1357	641	43	15	64682	46988
1452	770	43	16	66471	58702
1171	52	43	16	68799	40445

Para cada um dos exemplos, testou-se mais do que um conjunto de peças de trabalho e apresentam-se as soluções correspondentes a um menor custo.

Os tempos apresentados (segundos), foram obtidos num microcomputador Sperry IT, e referem-se à geração dos "duties" e à obtenção da solução a partir destes.

exemplo	Nº SERVIÇOS		NºPEÇAS	NºEQUIP	CUSTO	LIMITE INF	TEMPO CPU	
	gerados	eliminados					ger.	solu.
1	91	37	25	13	38491.00	35556.00	2.80	1.32
* 2	244	146	23	16	46988.00	46988.00	3.46	1.82
3	1357	641	43	15	64682.00	38668.00	74.93	3.57

(\*) Para o conjunto de peças de trabalho considerado a solução obtida para o problema formulado como MC é óptima.

exemplo 4:

PROGRAMA	NºPEÇAS	NºEQUIPAS	CUSTO	LIMITE INF	TEMPO CPU	
					geração	solução
RN	---	6	22150.00	----	---	19.06
CHAPAS	24	6	22132.00	18031.00	7.85	1.48

exemplo 5:

PROGRAMA	NºPEÇAS	NºEQUIPAS	CUSTO	LIMITE INF	TEMPO CPU	
					geração	solução
RN	---	22	75220.00*	----	---	73.22
CHAPAS	51	24	81004.00	59026.00	7.85	3.57

A linha indicada por RN refere-se à solução obtida para o problema MC, afectando manualmente os "duties" (obtidos otimizando uma relaxação linear) ao serviço das viaturas. Este era o método usado anteriormente.

Para este caso não se conhecem os tempos de cpu referentes à geração dos "duties".

Note-se que os tempos de cpu apresentados para os dois métodos não podem ser directamente comparados. Enquanto no método presentemente descrito se obtêm "duties" em que o serviço dos motoristas está completamente definido (serviços + peças de trabalho), no método anteriormente em uso apenas se obtinha um conjunto de serviços, sendo necessária a sua posterior afectação ao serviço das viaturas.

Essa afectação era feita manualmente, podendo demorar mais do que um dia normal de trabalho, (no exemplo 5 foram necessários dois dias).

Também em relação aos custos, tem interesse chamar a atenção para a discrepância observada nos custos obtidos no exemplo 5. Essa discrepância é explicada pelo facto de na afectação manual dos serviços, ao serviço das viaturas, se violar uma ou outra norma contratual (por ex: nº max de horas extraordinárias).

E por vezes, pequenas alterações deste tipo têm grande influência no custo final obtido.

Tinha interesse apresentar mais resultados do tipo dos dois últimos exemplos mas não se dispôs dos ficheiros de dados necessários, pois alguns deles sendo auxiliares, tinham sido destruídos.

## 5. CONCLUSÕES

O número de casos reais testados não foi suficiente para tirar conclusões quanto à qualidade das soluções obtidas. Houve contudo, uma melhoria considerável no tempo necessário à obtenção de uma solução admissível, sendo esta a tarefa mais difícil e morosa.

## 6. BIBLIOGRAFIA

- [1] Balas, E. and Ho, A. -'Set Covering Algorithms using Cutting Planes, Heuristics, and Subgradient Optimization: A Computational Study', Carnegie-Mellon University, Pittsburg, Pennsylvania (July 1979).
- [2] Constantino, M. -'Relatório de Estágio', Faculdade de Ciências, Universidade de Lisboa.
- [3] Garfinkel, R. e Nemhauser, G. -'Integer Programing', Wiley and sons, New York, 1972, capítulo 8.
- [4] Paiais, A.M. -'Manual de utilização dos programas Chapas1 e Chapas2' Faculdade de Ciências, Universidade de Lisboa.
- [5] Paiais, A.M. -'Relatório de Estágio', Faculdade de Ciências, Universidade de Lisboa, 1987.
- [6] Paixão, J.P., Eusébio, J.A.R. e Amado, L. -'Generation of Bus Driver Schedules', Nota nº12, CEAUL (1985).



## IMPLEMENTAÇÃO DE UMA HEURÍSTICA PARA DETERMINAÇÃO DAS ROTAÇÕES DE SERVIÇOS DO PESSOAL TRIPULANTE

Maria da Conceição da Fonseca  
D.E.I.O.C  
Faculdade de Ciências de Lisboa

### Resumo

Nesta comunicação, descreve-se uma heurística desenvolvida e implementada na empresa pública Rodoviária Nacional, (RN), para determinar as rotações de serviços do pessoal tripulante.

A heurística consiste numa adaptação do método apresentado por Carraresi e Gallo [1984] para o problema matemático designado por problema de multi-afecção "bottleneck".

As rotações produzidas garantem a realização de todas as chapas previstas para a semana de trabalho e são obtidas por forma a uniformizar o mais possível as correspondentes retribuições.

As rotações semanais obtidas podem posteriormente associar - se de forma a cobrir um período de tempo mais lato.

Apresentam-se alguns resultados obtidos com problemas reais.

### Abstract

In this paper, we describe an heuristic developed and implemented for obtaining driver rosters at Rodoviária Nacional, a large mass transport operator in Portugal.

The heuristic is an adaptation of the method presented by Carraresi and Gallo [1984] for the multilevel bottleneck assignment problem. It provides the stable rosters satisfying the requirement that every shift due in a given time horizon is assigned to a driver. This is done in such way that an even distribution of the workload is obtained by minimizing the maximum total cost of each roster in the final solution.

Firstly, the weekly rosters are generated by considering only the shifts performed in 5 consecutive regular workdays followed up by 2 consecutive days - off. Then, these are combined in order to obtain the rostering over the time horizon.

Experimental results obtained with real life problems are presented.

**Keywords :** afecção "bottleneck"

### 1. Definição do problema

O problema a resolver consiste na determinação das escalas periódicas de serviço para cada um dos trabalhadores, a partir das chapas diárias relativas aos serviços a realizar numa determinada zona. São estas escalas periódicas que designaremos por rotações.

No caso concreto da Rodoviária Nacional, as rotações pretendidas, além de terem de garantir a realização de todas as chapas previstas, devem ainda respeitar as seguintes condições :

- existência de 2 dias de folga consecutivos ;
- existência de 5 dias de trabalho consecutivos ;
- existência de pelo menos 10 horas de descanso entre chapas realizadas em dias consecutivos e atribuídas ao mesmo trabalhador.

Admite - se, contudo, a violação desta última restrição, sendo no entanto penalizada com um custo adicional de horas retiradas ao descanso.

Pretende-se calcular as rotações que ao longo de um determinado horizonte de tempo estabelecido garantam a realização das chapas previstas, verifiquem as condições acima referidas, isto é, sejam admissíveis e uniformizem as remunerações atribuídas a cada trabalhador.

Este objectivo é atingido, minimizando o máximo dos custos das rotações na solução final.

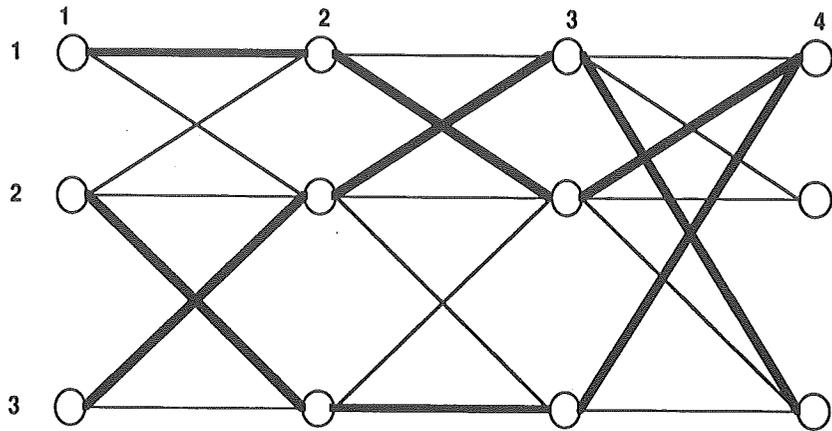
O problema definido pode formular-se como problema de redes. Assim, considere-se um grafo  $m$ -partido  $G=(N_1, \dots, N_m, A_1, \dots, A_{m-1})$  definido do modo seguinte :

cada conjunto de vértices  $N_i$ ,  $i=1, \dots, m$ , contém um nodo por cada chapa prevista para o  $i$ -ésimo dia incluindo as folgas;

cada conjunto de arcos  $A_i$ ,  $i=1, \dots, m-1$ , contém os arcos que constituem sequências admissíveis entre as chapas do  $i$ -ésimo dia e chapas do  $(i+1)$ -ésimo dia.

Uma solução admissível para o problema corresponderá, neste caso, a um conjunto de  $n$  caminhos disjuntos no grafo  $m$ -partido acima definido.

Por exemplo para  $m=4$  e para  $n=3$ , ter-se-ia como possível grafo e como possível solução admissível :



— Solução admissível

## 2. Método de resolução

Tratando-se de um problema de difícil resolução exacta devido ao facto de a função objectivo ser não linear e, ainda, ao tipo de restrições envolvidas, justifica-se plenamente o recurso a métodos heurísticos.

A heurística utilizada consiste em primeiro lugar na transformação do problema dado num problema de multi-afecção "bottleneck" ou seja, numa sequência de problemas de afectação em que a função objectivo é da forma min max. No fundo, vamos introduzir algumas alterações de maneira a garantir que as sequências de chapas sejam admissíveis e se aproximem o mais possível do objectivo desejado.

O método de resolução do problema de multi-afecção "bottleneck" (MBA) foi adaptado e desenvolvido para a Rodoviária Nacional a partir do processo apresentado por Carraresi e Gallo (1984) [1].

Vejamos, em primeiro lugar, como ultrapassar as dificuldades relativas à construção de sequências de chapas admissíveis.

Relativamente à restrição referente às horas de descanso é fácil incluí-la associando custos infinitos a qualquer sequência de chapas não admissível.

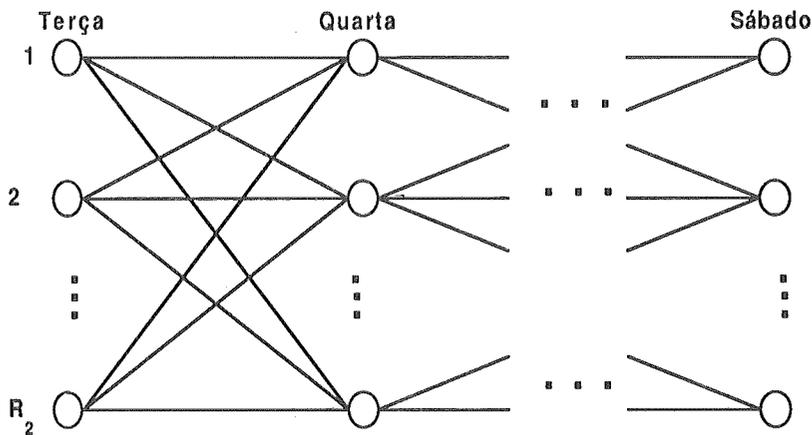
No caso específico da Rodoviária Nacional em que existem 5 dias de trabalho consecutivos aos quais se seguem 2 dias de folga também consecutivos existem 7 tipos

diferentes de rotações semanais ilustradas na figura abaixo.

Tipo/Dia	SEG	TER	QUA	QUI	SEX	SAB	DOM
1	X	X	X	X	X	-	-
2	-	X	X	X	X	X	-
3	-	-	X	X	X	X	X
4	X	-	-	X	X	X	X
5	X	X	-	-	X	X	X
6	X	X	X	-	-	X	X
7	X	X	X	X	-	-	X

X dia de trabalho  
 - dia de folga

Um trabalhador a quem é atribuída por exemplo uma rotação semanal do tipo 3 trabalhará de quarta a domingo, tendo os seus dias de folga na segunda e terça-feira. Sabendo à priori o número  $R_j$  de rotações semanais de cada tipo  $j$  necessárias para realizar todas as chapas previstas em determinada zona, basta seleccionar para cada um dos dias de trabalho que constituem as rotações do tipo  $j$ ,  $R_j$  chapas e resolver o problema MBA resultante para obter as  $R_j$  rotações semanais do tipo  $j$  pretendidas. O número de rotações semanais necessárias para cada um dos tipos é determinada pelo programa SHEFF existente na Rodoviária Nacional [2]. Suponhamos por exemplo que são necessárias  $R_2$  rotações do tipo 2 para satisfazer a realização das chapas previstas. Seleccionávamos então  $R_2$  chapas de terça-feira,  $R_2$  chapas de quarta-feira e assim sucessivamente até sábado. Ficávamos, deste modo com um problema MBA representado no grafo abaixo e de cuja resolução resultaria a obtenção de  $R_2$  escalas semanais do tipo 2.



Considerando para a obtenção das rotações semanais de cada tipo apenas as chapas que dizem respeito aos 5 dias consecutivos de trabalho eliminámos as dificuldades que as restrições relativas aos dias de trabalho consecutivos e aos dias de folga também consecutivos apresentavam. Isto aliado à atribuição de custos infinitos às ligações não

admissíveis entre chapas que se realizam em dias consecutivos permite-nos transformar o problema inicial num problema MBA.

O problema MBA pode formular-se do seguinte modo :

$$\min z$$

$$\text{sujeito a :} \quad (i) \quad \sum_{j \in S_k(i)} x_{ij}^k = 1 \quad \begin{array}{l} k=1, \dots, m-1; \\ i=1, \dots, n \end{array}$$

$$(ii) \quad \sum_{j \in P_k(i)} x_{ij}^k = 1 \quad \begin{array}{l} k=1, \dots, m-1; \\ j=1, \dots, n \end{array}$$

$$(iii) \quad s_j^1 = w_{1j} \quad j=1, \dots, n$$

$$s_j^k = w_{kj} + \sum_{i \in P_{k-1}(j)} s_i^{k-1} x_{ij}^{k-1} \quad \begin{array}{l} j=1, \dots, n \\ k=2, \dots, m \end{array}$$

$$v_i^m = w_{mi} \quad i=1, \dots, n$$

$$v_i^k = w_{ki} + \sum_{j \in S_{k+1}(i)} v_j^{k+1} x_{ij}^k \quad \begin{array}{l} i=1, \dots, n; \\ k=m-1, \dots, 1 \end{array}$$

$$(iv) \quad x_{ij}^k \in \{0, 1\} \quad \begin{array}{l} k=1, \dots, m-1; \\ i, j=1, \dots, n \end{array}$$

onde  $m$  : número de dias

$n$  : número de chapas diárias

$w_{kj}$  : custo da chapa  $j$  do dia  $k$

$S_k(i)$  : conjunto das chapas que no dia  $(k+1)$  sucedem à chapa  $i$  do dia  $k$ .

$P_k(i)$  : conjunto das chapas que no dia  $k$  antecedem a chapa  $i$  do dia  $(k+1)$ .

$$x_{ij}^k = \begin{cases} 1 & \text{se à chapa } i \text{ do dia } k \text{ se segue a chapa } j \text{ do dia } (k+1) \\ 0 & \text{caso contrário} \end{cases}$$

$$z = \max \{ s_i^m : i=1, \dots, n \} = \max \{ v_i^1 : i=1, \dots, n \}$$

As restrições (i) e (ii) dizem respeito à garantia de sucessão diária de chapas. A restrição (iii) define os vectores  $s^k$  e  $v^k$  ( $k=1, \dots, m$ ) enquanto a restrição (iv) define o domínio de variação das variáveis  $x_{ij}^k$ .

Para  $m=2$  este problema resume-se ao caso especial de um problema de afectação "bottleneck".

No caso  $m > 2$  a determinação de uma solução admissível para MBA consiste na

resolução de uma sequência de (m-1) problemas de afectação "bottleneck",  $BA_k$ , ( $k=1, \dots, m-1$ ) :

Para a resolução de cada um dos  $BA_k$ , considera - se que o custo de cada chapa  $j$  ( $j = 1, \dots, n$ ) do dia  $k$  ( $k = 1, \dots, m-1$ ) é dado pelo correspondente  $s^k_j$  (ou seja, o custo acumulado) enquanto o custo de cada chapa  $j$  do dia  $(k+1)$  é  $w_{(k+1)j}$ .

Assim, para cada  $BA_k$ , entra - se em linha de conta com o custo das afectações já efectuadas até ao correspondente dia  $k$ .

Ao resolver a sequência de problemas de afectação "bottleneck" tem-se a garantia de obtenção de uma solução admissível para MBA.

Esta solução é susceptível de ser melhorada através do seguinte processo :

- para cada  $BA_k$  considera - se o custo de cada chapa  $j$  no dia  $k$  igual a

$s^k_j$  e no dia  $(k+1)$  igual a  $v^{k+1}_j$ .

- resolve - se novamente a sequência de problemas obtendo - se nova solução admissível eventualmente melhor que a anterior.

No caso de se obter uma solução com custo inferior ao da obtida anteriormente, esta é actualizada e o processo repete - se agora com a nova solução, até que não seja possível melhorar o valor da correspondente solução admissível. Obtém - se então o que se designa por uma solução estável.

Falta finalmente referir os métodos utilizados para a resolução de cada um dos problemas de afectação "bottleneck". Para resolver este problema foram estudados e implementados os algoritmos de Derigs e Zimmermann [2] e de Carraresi e Gallo, os quais serão seguidamente descritos nos seus pontos essenciais.

## 2.1 Algoritmo de Derigs e Zimmermann

Defina - se em primeiro lugar :

$q(i)$  : sucessor de  $i$

$p(j)$  : antecessor de  $j$

$c_{ij}$  : custo do arco  $(i, j)$ , ( $i, j=1, \dots, n$ )

$z$  : um limite inferior para o valor óptimo.

### Algoritmo

PASSO 1. Calcular o limite inferior  $z = \max_{1 \leq i \leq n} \{ r_i, s_i \}$

$$\text{em que } r_i = \min_{1 \leq j \leq n} c_{ij} \quad s_j = \min_{1 \leq i \leq n} c_{ij} \quad i, j = 1, \dots, n$$

PASSO 2. Calcular para cada  $i$   $a_i = \# \{ (i, j) : j=1, \dots, n \text{ e } c_{ij} \leq z \}$ , isto é, o número de arcos divergentes de  $i$  e que são admissíveis.

PASSO 3. Calcular para cada  $j$   $b_j = \# \{ (j, i) : i=1, \dots, n \text{ e } c_{ji} \leq z \}$ , isto é, o número de arcos divergentes de  $j$  e que são admissíveis.

PASSO 4. Fazer  $L=C=\{ 1, 2, \dots, n \}$ ,  $q(i)=p(i)=0 \quad i = 1, \dots, n$

PASSO 5. { Determinação de uma solução parcial inicial }

Repetir até que  $L = \emptyset$

- Escolher  $i \in L$  tal que  $a_i = \min_{t \in L} a_t$

-  $L = L \setminus \{ i \}$

- Escolher  $j \in C$  tal que  $b_j = \min_{t \in L} b_t$  e existe  $(i, j)$

- Se existe  $j \in C$  nestas condições. Então  $C = C \setminus \{j\}$   
 $q(i) = j$   
 $p(j) = i$

PASSO 6. { "Aumentar" a solução parcial inicial e actualizar z }

Para  $i$  tal que  $q(i) = 0$  fazer

6.1 Para  $j = 1$  até  $n$  fazer  $D(j) = c_{ij}$  e  $V(j) = i$

6.2 Determinar  $j$  tal que  $D(j) = \min_{1 \leq t \leq n} D(t) = D$

Se  $p(j) = 0$  Então vai para 6.3

Senão Determinar  $l$  tal que  $j = q(l)$

Determinar  $t = 1, \dots, n$  tal que  $t \neq j$  e  $c_{lt} < D(t)$

Para  $t$  nestas condições fazer  $D(t) = c_{lt}$   
 $V(t) = l$

Volta a 6.2

6.3 Fazer  $w = V(j)$ ,  $p(j) = w$ ,  $t = q(w)$ ,  $q(w) = j$

Se  $(w = i)$  Então  $z = \max \{ z, D \}$

Senão  $j = t$

Volta a 6.3

No caso particular da presente aplicação deste algoritmo, a matriz dos custos  $C = [c_{ij}]$  ( $i = 1, \dots, n$ ;  $j = 1, \dots, n$ ) é calculada do seguinte modo :

$$c_{ij} = c_i + c_j + p_{ij}$$

onde  $c_i$  : custo da chapa  $i$  do dia  $k$

$c_j$  : custo da chapa  $j$  do dia  $(k+1)$

$p_{ij}$  : penalização por associar a chapa  $i$  do dia  $k$  com a chapa  $j$  do dia  $(k+1)$

O valor de  $p_{ij}$  será zero se for respeitado o número de horas de descanso entre chapas realizadas em dias consecutivos, caso contrário será dado por  $p_{ij} = c * h_{ij}$  onde  $c$  é o custo fixo por hora retirada ao descanso e  $h_{ij}$  é o número de horas retiradas ao descanso quando se associa a chapa  $i$  do dia  $k$  com a chapa  $j$  do dia  $(k+1)$ .

## 2.2 Algoritmo de Carraresi e Gallo

Vamos começar por definir alguns símbolos e conceitos utilizados.

Sejam  $N_1$  e  $N_2$  : conjuntos de vértices num grafo bipartido.

$$(\# N_1 = \# N_2 = n)$$

$a_i$  : peso associado ao vértice  $i \in N_1$

- $b_j$  : peso associado ao vértice  $j \in N_2$
- $y$  : solução admissível para o problema de afectação "bottleneck",  $BA$ .
- $q(i)$  : elemento de  $N_2$  tal que  $y_{iq(i)} = 1$
- $p(i)$  : elemento de  $N_1$  tal que  $y_{p(i)i} = 1$
- $M = \{ (i,j) : y_{ij} = 1 \}$

*Definição* : Diz-se que  $(r,s)$  é um arco "bottleneck" se  $a_r + b_s = \max \{ a_i + b_{q(i)} : i \in N_1 \}$ .

*Definição*: Cadeia alternante decrescente de  $r$  para  $s$  é uma cadeia de  $r$  para  $s$  tal que

- (i) não contém  $(r, s)$
- (ii) os arcos que a constituem pertencem alternadamente a  $M$  e a  $\bar{M}$ .
- (iii) os arcos que não pertencem a  $M$  têm custos associados menores do que  $a_r + b_s$ .

### Algoritmo

*PASSO 1* - Determinação de uma solução inicial que pode ser não admissível.

*PASSO 2* - Determinação do arco "bottleneck"  $(r, s)$ .

*PASSO 3* - Determinação de uma cadeia alternante decrescente de  $r$  para  $s$ .  
 Se não existe cadeia nestas condições  
 Então Se a solução de que dispomos é admissível Então ela é ótima  
 Caso contrário não existe solução para o problema  
 Caso contrário vai para 4.

*PASSO 4* - Actualização da solução.  
 Se a nova solução é diferente da anterior então volta a 2 com a nova solução.  
 Caso contrário a solução obtida é ótima.

Foi utilizado o algoritmo de Derigs e Zimmermann para resolver os casos em que se admite violação da restrição respeitante ao número mínimo de horas de descanso entre chapas realizadas em dias consecutivos e atribuídas ao mesmo trabalhador, visto que como foi referido atrás a estrutura deste algoritmo permite considerar, além do custo de cada chapa diária, o custo adicional que penaliza tais sequências não admissíveis no problema original.

Nos restantes casos, utilizou-se também o algoritmo de Carraresi e Gallo uma vez que em termos de tempo de execução este último algoritmo é francamente melhor. Existe assim nestes casos a possibilidade de resolver o problema pelos dois algoritmos e optar depois pela melhor solução obtida.

O método descrito permite - nos determinar as rotações semanais necessárias de cada um dos tipos. No caso da Rodoviária Nacional em que as folgas são fixas para cada um dos trabalhadores, as escalas de serviço para mais do que uma semana obtém - se associando rotações semanais do mesmo tipo.

Esta associação é feita utilizando o método descrito para determinar as rotações semanais substituindo, neste caso, as chapas diárias pelas rotações semanais do mesmo tipo.

### 3. Programas desenvolvidos e Resultados obtidos

A heurística apresentada foi implementada num microcomputador IBM-compatível, tendo sido utilizada a linguagem Fortran 77 para codificação do programa do cálculo das rotações.

Foram ainda implementados em Turbo - Pascal vários programas que permitem :

- fazer a gestão do ficheiro com informação relevante para o cálculo das rotações e relativa às chapas previstas para determinada zona ;
- fazer a selecção das chapas para cada um dos grupos de rotações ;
- fazer alterações no output constituído pelas rotações obtidas pela heurística, trocando de cada vez, 2 chapas do mesmo dia, sempre que a troca seja admissível ;
- representar gráficamente as rotações obtidas e respectivos custos .

Tudo isto é feito de um modo acessível a qualquer utilizador recorrendo à utilização de menus.

Apresentam-se seguidamente alguns resultados obtidos para vários quintúplos de chapas de cada um dos sete tipos atrás referidos.

Os dados utilizados dizem respeito a chapas diárias reais previstas para as zonas de Palmela e Carcavelos.

Para os dois exemplos os resultados foram obtidos quer utilizando o algoritmo de Derigs e Zimmermann para resolver o problema de afectação "bottleneck", quer utilizando o algoritmo de Carraresi e Gallo para resolver o mesmo problema. Isto permite-nos comparar os 2 algoritmos em termos de tempos de execução e de soluções obtidas.

#### Exemplo 1

Os resultados referentes a este exemplo foram obtidos num microcomputador Sperry PC/IT, sendo os tempos de execução referidos em segundos .

As chapas foram seleccionadas de entre as previstas para a zona de Palmela durante uma semana típica.

Os custos variáveis das chapas estão entre 0.0 e 1760.7 escudos.

Utilizando para resolver cada um dos problemas de afectação "bottleneck" o algoritmo de Derigs e Zimmermann obtiveram-se os seguintes resultados :

<u>Tipo de rotação</u>	<u>Nº de chapas</u>	<u>Mínimo</u>	<u>Máximo</u>	<u>Tempo exec.</u>
<u>Seg - Sexta</u>	14	3969.10	4686.90	8.60
<u>Terça - Sab.</u>	12	3071.90	3888.20	4.40
<u>Quarta-Dom.</u>	9	3260.10	3937.90	2.58
<u>Quinta -Seg.</u>	9	2586.90	3298.60	2.53
<u>Sexta - Terça</u>	9	2584.50	3114.60	2.37
<u>Sab. - Quarta</u>	9	1612.70	3340.10	2.96
<u>Dom. - Quinta</u>	9	1898.40	3589.90	2.01

Utilizando o algoritmo de Carraresi e Gallo os resultados obtidos foram :

<u>Tipo de rotação</u>	<u>Nº de chapas</u>	<u>Mínimo</u>	<u>Máximo</u>	<u>Tempo exec.</u>
<u>Seg - Sexta</u>	14	3432.50	4984.90	1.14
<u>Terça - Sab.</u>	12	3364.10	3820.40	0.95
<u>Quarta-Dom.</u>	9	3095.00	4003.70	0.74
<u>Quinta -Seg.</u>	9	2613.60	3239.70	0.69
<u>Sexta - Terça</u>	9	2457.40	3082.60	0.89
<u>Sab: - Quarta</u>	9	1781.10	3406.10	0.57
<u>Dom. - Quinta</u>	9	1781.50	3489.40	0.56

Uma análise destes dois quadros com os resultados mostra-nos que o algoritmo de Carraresi e Gallo é, em termos de tempo de execução, francamente melhor que o de Derigs e Zimmermann.

Quanto às soluções obtidas verifica-se que em alguns casos o algoritmo de Carraresi e Gallo dá melhores resultados que o algoritmo de Derigs e Zimmermann, enquanto que noutros casos este último é nitidamente melhor.

Sugere-se pois que, sempre que possível, isto é, nos casos em que se respeita a restrição referente ao número de horas de descanso entre chapas realizadas em dias consecutivos se adopte uma das duas soluções propostas :

- Resolver o problema recorrendo aos 2 algoritmos, escolhendo depois a melhor solução obtida.

- Resolver o problema utilizando o algoritmo de Carraresi e Gallo, uma vez que a sua execução é relativamente rápida.

No caso de os resultados obtidos serem pouco satisfatórios quanto à optimalidade, então utilizar o algoritmo de Derigs e Zimmermann. Podem assim comparar-se as soluções obtidas e escolher a melhor.

### Exemplo 2

Os resultados referentes a este exemplo foram obtidos num microcomputador NCR PC8 com os tempos de execução igualmente expressos em segundos.

As chapas foram seleccionadas de entre as previstas para a zona de Carcavelos durante uma semana típica do período escolar e com custos variáveis das chapas entre 0.0 e 1838.80 escudos.

Resultados obtidos utilizando o algoritmo de Derigs e Zimmermann :

<u>Tipo de rotação</u>	<u>Nº de chapas</u>	<u>Mínimo</u>	<u>Máximo</u>	<u>Tempo exec.</u>
<u>Seg - Sexta</u>	<u>25</u>	<u>1579.20</u>	<u>4739.70</u>	<u>27.9</u>
<u>Terça - Sab.</u>	<u>25</u>	<u>4236.50</u>	<u>4544.70</u>	<u>18.81</u>
<u>Quarta-Dom.</u>	<u>25</u>	<u>3452.40</u>	<u>4276.40</u>	<u>32.07</u>
<u>Quinta -Seg.</u>	<u>25</u>	<u>3510.40</u>	<u>3784.90</u>	<u>24.15</u>
<u>Sexta - Terça</u>	<u>25</u>	<u>2727.00</u>	<u>3607.50</u>	<u>29.13</u>
<u>Sab. - Quarta</u>	<u>25</u>	<u>1535.60</u>	<u>3907.90</u>	<u>16.91</u>
<u>Dom. - Quinta</u>	<u>25</u>	<u>1544.90</u>	<u>4000.80</u>	<u>26.01</u>

Resultados obtidos utilizando o algoritmo de Carraresi e Gallo :

<u>Tipo de rotação</u>	<u>Nº de chapas</u>	<u>Mínimo</u>	<u>Máximo</u>	<u>Tempo exec.</u>
<u>Seg - Sexta</u>	<u>25</u>	<u>1633.20</u>	<u>4993.80</u>	<u>6.20</u>
<u>Terça - Sab.</u>	<u>25</u>	<u>3765.60</u>	<u>4660.20</u>	<u>2.63</u>
<u>Quarta-Dom.</u>	<u>25</u>	<u>2988.70</u>	<u>5006.00</u>	<u>1.99</u>
<u>Quinta -Seg.</u>	<u>25</u>	<u>2924.80</u>	<u>4112.80</u>	<u>2.61</u>
<u>Sexta - Terça</u>	<u>25</u>	<u>1871.80</u>	<u>4023.60</u>	<u>3.00</u>
<u>Sab. - Quarta</u>	<u>25</u>	<u>1927.00</u>	<u>4322.20</u>	<u>2.95</u>
<u>Dom. - Quinta</u>	<u>25</u>	<u>1709.60</u>	<u>4262.10</u>	<u>4.10</u>

Este exemplo confirma as conclusões sugeridas pelo caso anterior e, embora a nossa

experiência tenha sido algo limitada, parece-nos que qualquer dos algoritmos estudados e implementados constitui um elemento útil de abordagem do problema de determinação das rotações de serviços do pessoal tripulante.

## BIBLIOGRAFIA

- [1] Carraresi, P. e Gallo, G.,  
"A Multi-Level Bottleneck Assignment Approach To The Bus Drivers Rostering Problem ", European Journal of Operational Research 16 (1984), pag163-173.
- [2] Eusébio, J.R., Amado, L., Paixão, J.  
"Development and Implementation of an Automatic System for Bus and Crew Scheduling at RN - Portugal" in "Computer-aided Scheduling of Public Transport" Lecture Notes in Economics and Mathematical Systems, Spring-Verlag, Berlin Heidelberg New York (1988), pag 147-159
- [3] Bukard R. e Derigs, U.,  
"Assignment and Matching Problems : Solution Methods with Fortran Programs" Lecture Notes in Economics and Mathematical Systems, Spring-Verlag Berlin Heidelberg New York (1980), pag 16-23.
- [4] Fonseca, M.C.  
"Relatório de Estágio", Faculdade de Ciências, Universidade de Lisboa.

## SINGLE MACHINE SCHEDULING: MINIMIZING THE WEIGHTED NUMBER OF LATE JOBS

Jorge Pinho de Sousa

Faculdade de Engenharia da Universidade do Porto

### Resumo

As formulações como Programação Inteira parecem constituir um instrumento com grandes potencialidades para o sequenciamento de tarefas em um ou mais processadores. Neste trabalho ilustra-se esta abordagem no caso da minimização do número pesado de tarefas em atraso num processador único. Este problema é, em geral, NP-completo. São referidas formulações alternativas e descritas brevemente as técnicas utilizadas para resolver o problema. Estas técnicas baseiam-se fundamentalmente em resultados da teoria combinatória poliédrica, em particular na reformulação do problema original e na geração de desigualdades válidas fortes. Finalmente, são apresentados resultados computacionais para problemas envolvendo até 400 tarefas, e sugeridos alguns desenvolvimentos a partir da abordagem apresentada.

### Abstract

Integer Programming formulations appear to be a promising general tool to handle single and multiple machine scheduling problems. In this work we illustrate this general approach by considering the minimization of the weighted number of late jobs on a single machine. This problem is NP-complete. We present alternative formulations and describe briefly the approach used to solve the problem. This approach relies primarily on results in polyhedral combinatorics, and in particular on the reformulation of the original problem and on the generation of strong valid inequalities. Some computational results for the problem instances with up to 400 jobs are presented, and generalizations of the basic approach are suggested.

**Keywords:** Machine scheduling; weighted number of late jobs; integer programming; branch-and-bound.

## 1. INTRODUCTION

Consider the problem of scheduling  $n$  jobs on a single machine which can process only one job at a time. For each job  $i$  (where  $i = 1, \dots, n$ ), let  $p_i$  be its processing time,  $d_i$  its due date and  $w_i$  the penalty (weight) incurred if job  $i$  is late. For a given sequence, we say that a job is late if its completion time is greater than its due date. We assume that the processing times and the due dates are non-negative integers and that the weights are non-negative as well. We also assume that job  $i$  has  $p_i \leq d_i$ .

In this work, our aim consists in defining a sequence such that the sum of the weights of the late jobs is minimized. For convenience we will also assume that the jobs have been previously ordered according to their earliest due dates (EDD). This means that two consecutive jobs  $i$  and  $i+1$  have  $d_i \leq d_{i+1}$ .

In order to present a solution for the problem we do not need to know the exact optimal sequence. It is enough to have the jobs partitioned in two disjoint sets: one with those which are completed in time (we will call these the **early** jobs) and another with the **late** jobs. Then the solution to be implemented consists in sequencing the early jobs by the EDD rule and the late jobs arbitrarily (see Lawler and Moore [1969] for a proof).

To identify the different types of problems, we use the standard 3-field notation  $\alpha|\beta|\gamma$  (Lawler et al. [1982]), where  $\alpha$  will always be 1 as we are only dealing with single machine problems. The second field  $\beta$  describes the characteristics of the jobs. By default, we assume that no release dates are present (i.e. all jobs become available for processing at the same time), no preemption (job splitting) is allowed and no precedence constraints exist. The third field  $\gamma$  identifies the objective function. The unit penalty incurred by job  $j$  when it is late, is usually denoted by  $U_j$ . Therefore, in the present case we let  $\gamma$  be equal to  $\sum w_j U_j$ , or just  $\sum U_j$  if the weights are equal (unit).

### Minimizing the weighted number of late jobs

The general weighted version of the problem ( $1||\sum w_j U_j$ ) is NP-complete. This can easily be shown by reduction from the *knapsack* problem (Karp [1972]). In spite of its intrinsic *hardness*, by using a dynamic programming approach (Lawler and Moore [1969]) the problem can be efficiently solved in pseudopolynomial time  $O(n \max\{d_j\})$ .

There are however some special cases for which efficient approaches exist. On the other hand, some extensions of the above problem present additional structure which can be successfully used to design solution procedures. We will now briefly describe these particular cases.

### Special cases

If all the weights are equal ( $1||\sum U_j$ ), the problem can be solved in  $O(n \log n)$  time, by the following simple procedure:

**Algorithm (Moore [1968]).** Jobs are added to the schedule in order of nondecreasing due dates (EDD). If addition of job  $j$  results in this job being completed after  $d_j$ , the scheduled job with the largest processing time is marked to be late and removed.

When the processing times are equal ( $1|p_j = p|\sum w_j U_j$ ), the problem is *easy*

to solve. It can be formulated as an *assignment* problem (Lawler [1964]) or handled as a *matroid* (Lawler [1976]).

When the weights are *agreeable* (i.e. when we have  $w_i \geq w_j$  for all jobs  $i$  and  $j$  such that  $p_i < p_j$ ), the problem can be solved in  $O(n \log n)$  time (Lawler [1976]). Note that this case encompasses the two previous ones.

If all the jobs have the same due date ( $1|d_j = d|\sum w_j U_j$ ), the problem can be viewed as a *knapsack* problem (this will become clear with the formulation presented later), thus being reasonably easy to solve.

### Some extensions

If arbitrary *release dates*  $r_j$  are allowed, even in the case of unit weights ( $1|r_j|\sum U_j$ ) we have a strongly NP-complete problem. However, if we have an ordering of the jobs such that  $r_i \leq r_{i+1}$  with  $d_i \leq d_{i+1}$  for all  $i = 1, \dots, n$  (case of *compatible* release and due dates), the unit weights problem can be solved in polynomial time (Kise et al. [1978], Lawler [1983]).

When *preemption* (job splitting) is allowed, ( $1|pmtn, r_j|\sum w_j U_j$ ), the problem can be solved by a dynamic programming procedure (Lawler [1983]), running in time  $O(n^3 W^3)$  where  $W = \sum w_j$  (integer weights assumed). This implies that if the weights are unit, the running time will be polynomial.

In this work we will describe an approach based on integer programming formulations, for the general problem of minimizing the weighted sum of late jobs, without release dates ( $1|\sum w_j U_j$ ). These formulations seem to be reasonably *strong*, providing fast optimal solutions when inserted in a general purpose enumeration procedure.

## 2. GENERAL FRAMEWORK

Scheduling problems can in general be easily formulated as linear pure (or mixed) integer programs. This means that a problem is described by specifying a decision space (i.e. a set of decision variables)  $x$ , a linear objective function  $cx$ , and a feasible region for  $x$ , represented by a set of linear constraints  $Ax \leq b$ . Note that some (or all) components of  $x$  must satisfy the additional constraint of being integer (or binary, in particular). This general approach turns out to be quite suitable, as scheduling problems deal with sequences of tasks. These sequences are basically different permutations of the whole set of tasks, being therefore *naturally* modelled by some kind of discrete decision variables.

Note that for a given decision space, we may have different formulations, i.e. we may have different *polyhedra* which, along with the *integrality* constraints,

represent just the same set of feasible solutions. Remember that a polyhedron  $P \subseteq \mathbb{R}^n$  is the set of points in  $\mathbb{R}^n$  satisfying a finite number of linear inequalities i.e.,  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ .

As we deal with integer variables when trying to solve a scheduling problem, we will be forced to use some kind of *enumeration*, typically a *branch-and-bound* procedure. We would like this enumeration to be as effective and efficient as possible, and this may be achieved by using *strong* formulations, i.e. polyhedral descriptions as close as possible to the *convex hull* of  $P$ ,  $\text{conv}(P)$ . Ideally such a description would consist of the convex hull itself. In this ideal situation, relaxing the integrality constraints and solving the resulting linear program (l.p. relaxation) would lead us straight to the optimal (integer/mixed) solution for the original problem. This results from the fact that all the extreme points of  $\text{conv}(P)$  are feasible solutions for that original problem.

For this purpose, a first natural approach consists in reformulating some basic version of the problem, designing a new description of the feasible region through alternative, *stronger* constraints. In other words, the original  $P$  is replaced by another polyhedron.

In this context, we define a *valid inequality* for  $P$  as being an inequality  $\pi x \leq \pi_0$  which is satisfied by all points in  $P$ . If we are able to define families of valid inequalities for a given problem, we will be able to solve l.p. relaxations progressively closer to  $\text{conv}(P)$ , by iteratively adding some of those inequalities to the original formulation. Then either we find, in a reasonable amount of time, a solution in integers which must be optimal for the original problem, or we will have to turn at some stage, to a *branch-and-bound* procedure. In general, this will be the case as, due to the intrinsic *hardness* of most scheduling problems, there is little hope to avoid an enumeration phase.

Nevertheless this enumeration procedure will hopefully be faster, as good *lower bounds* on the optimal integer solution (for minimization problems) are provided by the l.p. relaxations.

On the other hand, we would like to dispose of some good feasible solution, thus having an *upper bound* on the optimal solution. The tighter this bound is, the shorter will be the *search* in the enumeration tree. Good initial feasible solutions can be obtained by *heuristic* algorithms whose effectiveness may unfortunately be quite problem dependent.

Note that any sequence of the jobs leads at least to one possible *schedule*, and defines therefore a feasible solution. In general, applying an adequate simple rule to create an initial sequence, followed by some exchange procedure performed on adjacent jobs, will lead to quite effective feasible solutions.

### 3. PROBLEM FORMULATION

The particular features of the problem under consideration (see section 1) allow us to use the following binary decision variables:  $u_i = 1$  if job  $i$  is early and  $u_i = 0$  otherwise ( $i = 1, \dots, n$ ). We assume here that  $d_1 \leq d_2 \leq \dots \leq d_n$  and that  $p_i \leq d_i$  for every  $i$ . This yields the following basic formulation:

$$\begin{aligned}
 &\text{Maximize} && \sum_{i=1}^n w_i u_i \\
 &\text{subj. to} && \sum_{i=1}^j p_i u_i \leq d_j && \text{for } j = 1, \dots, n \\
 &&& u_i \in \{0, 1\} && \text{for } i = 1, \dots, n.
 \end{aligned} \tag{3.1}$$

#### Redundant constraints

If  $d_{j+1} = d_j$ , constraint  $j$  is obviously redundant and can therefore be discarded. In the limit if for all  $i$ ,  $d_i = d$  (i.e. the due date  $d$  is the same for all jobs), all constraints except the last one are redundant and the problem reduces then to a 0/1 knapsack.

#### Reformulation

The constraints of the problem can be tightened in a very simple way (see Peters [1986]). Consider job  $j$  with due date  $d_j$ . Within the set of jobs which are not late, by time  $d_j$  the machine will have necessarily processed:

- 1] the jobs with due dates not greater than  $d_j$ , i.e. those for which  $i \leq j$  (this includes job  $j$  itself);
- 2] and partially, from the jobs  $k$  with  $d_k > d_j$ , those having  $d_k - p_k < d_j$ .

In fact, for a job  $k$  having  $d_k > d_j$  to be in time, it must start being processed at latest by time  $d_j - [p_k - (d_k - d_j)]$ . This yields the following set of valid inequalities which are obviously stronger than the original ones and can therefore replace them:

$$\sum_{i=1}^j p_i u_i + \sum_{i=j+1}^n \max\{0, d_j - (d_i - p_i)\} u_i \leq d_j \quad \text{for } j = 1, \dots, n-1 \tag{3.3}$$

In more general terms, this procedure corresponds to lifting the originally absent variables into each of the constraints. We will from now on, consider only this preprocessed formulation.

#### Bounds on the number of early jobs [MOO]

Let  $n_j^L$  be the minimum number of late jobs for the problem containing jobs  $1, \dots, j$  (where  $j = 1, \dots, n$ ). The numbers  $n_j^L$  are easily computed by considering

the simpler problem with *no weights*, i.e.  $1 \parallel \sum U_j$ . This is a valid relaxation of the original weighted problem, which can be tackled by a procedure based on Moore's algorithm, running in  $O(n \log n)$  time.

Let  $n_j^E = j - n_j^L$  be the maximum number of early jobs for the problem containing jobs  $1, \dots, j$  (the first  $j$  variables). Then we have the following *valid inequalities*:

$$\sum_{i=1}^j u_i \leq n_j^E \quad \text{for } j = 1, \dots, n \quad (3.4)$$

These inequalities do not result trivially from considering *covers* over each single original constraint (which might be viewed as a 0/1 knapsack). In fact, by construction, they result from the set of original constraints taken as a whole.

Note that we may have  $j$  such that  $\sum_{i=1}^j u_i \leq j$ , or such that  $n_j^E = n_{j+1}^E$ . In these cases, constraint  $j$  is obviously *redundant* and may therefore be discarded.

#### 4. SOME ADDITIONAL FEATURES

##### Equal processing times

If the processing times are all equal to  $p$ , they can be reduced to unity by simply dividing each constraint by  $p$ , thus yielding  $\sum_{i=1}^k u_i \leq d'_k$ ,  $d'_k$  being the largest integer less than  $d_k/p$ .

In this case, if we relax the integrality constraints (replacing these constraints by  $0 \leq u_i \leq 1$  for  $i = 1, \dots, n$ ), the resulting linear program will have an optimal solution in integers, i.e. an optimal solution for the original problem. This is easy to check as the due date constraints are as follows:

$$\sum_{i=1}^j u_i \leq d'_j \quad \text{for } j = 1, \dots, n \quad (4.1)$$

The coefficients matrix is an *interval matrix* (as in each column all non zero elements lay in adjacent rows), thus being totally unimodular.

##### Other bounds on the number of early jobs [LAW]

Let  $(i, k)$  denote the job with the  $i^{\text{th}}$  smallest processing time among the first  $k$  jobs and  $f_k$  be such that:

$$\sum_{j=1}^{f_k} p_{j,k} \leq d_k \leq \sum_{j=1}^{f_k+1} p_{j,k} \quad (4.2)$$

with  $f_k = k$  if the sum of the first  $k$  processing times is less than  $d_k$ . The problem of maximizing  $\sum_{i=1}^n w_i u_i$  subject to

$$\sum_{i=1}^k u_i \leq f_k \quad \text{for } k = 1, \dots, n \tag{4.3}$$

is a valid relaxation of the original problem (see Villarreal and Bulfin [1983] for a proof).

The numbers  $f_k$  and implicitly inequalities (4.2), can be easily computed after ordering the jobs by non-decreasing processing times. The relaxed problem obtained in this way has the same structure of the original problem, but all processing times are unit. This is a special case of *agreeable weights* (in the sense that  $p_i < p_j \Rightarrow w_i \geq w_j$ ) and an efficient algorithm exists to solve it (Lawler [1976]). In fact, as we are just trying to tighten our formulation, we are interested in the inequalities rather than in solving the relaxed problem itself.

However, in general we will only use the inequalities (3.4) and forget about these bounds, as the following result holds (we will skip the proof because it is lengthy but not difficult):

**Proposition 4.1.** *The valid inequalities (3.4) obtained through approach [MOO] are at least as strong as the valid inequalities (4.3) obtained using approach [LAW].*

**A dominance property [V+B]**

Finally we present a property (established by Villarreal and Bulfin [1983]) defining a relation between pairs of jobs:

**Theorem 4.2.** *If for two given jobs  $i$  and  $j$ ,  $p_i \leq p_j$ ,  $w_i \geq w_j$ , and  $d_i - p_i \geq d_j - p_j$ , then either  $i$  is early or  $j$  is late, i.e.*

$$u_j \leq u_i \tag{4.4}$$

Proof: Assume that we have an optimal schedule where job  $i$  is late, even with job  $j$  being early. Let  $c_k$  be the completion time of job  $k$ . If we interchange jobs  $i$  and  $j$ , job  $i$  will necessarily become early, as its new completion time  $c'_i$  will be:

$$c'_i = c_j - p_j + p_i \leq d_j - p_j + p_i \leq d_i \tag{4.5}$$

If job  $j$  remains early, the objective function is reduced by  $w_i$ , but even if it becomes late, the objective function can never be worse, since  $w_i \geq w_j$ .

□

## 5. AN ALTERNATIVE FORMULATION

The problem of minimizing the weighted number of late jobs may, in some sense, be viewed as a sequential decision process. Consider that in each stage  $i$  ( $i = 1, \dots, n$ ) we decide whether job  $i$  is put into the set of early jobs or not. The available time at each stage will be seen as a kind of resource, for which a consumption of  $p_i u_i$  units and an increase of  $d_i - d_{i-1}$  units take place.

The decision variable  $u_i$  is set to 1 if job  $i$  is early and to 0 otherwise. Let  $s_i$  be the amount of available time passed to stage  $i+1$ . Note that if a job is late, there is no *real* time consumption.

This yields the following valid formulation:

$$\begin{aligned}
 &\text{Maximize} && \sum_{i=1}^n w_i u_i \\
 &\text{subj. to} && p_1 u_1 + s_1 &= d_1 \\
 &&& p_j u_j - s_{j-1} + s_j &= d_j - d_{j-1} && \text{for } j=2, \dots, n \\
 &&& u_i \in \{0, 1\} && 0 \leq s_i && \text{for } i=1, \dots, n.
 \end{aligned} \tag{5.1}$$

Under this formulation, the problem can be seen as a special case of the *single item capacitated fixed charge lotsizing problem*. For this problem, a family of *valid inequalities* has been developed (see Pochet [1986]) which turn out to be, in the current case, the preprocessed due data constraints (3.3) of the previous formulation.

## 6. FEASIBLE SOLUTIONS

In order to obtain a good *initial feasible solution* and implicitly a good *bound* on the optimal integer solution, some heuristic algorithms have been tested.

A first approach was to consider *exchange procedures*: from a given sequence of the jobs, pairwise interchanges are performed until no improvement of the objective function takes place. The results so obtained are quite bad, due to the fact that the objective function does not increase continuously with the completion times of the jobs. This approach was therefore abandoned.

Now we describe a simple new heuristic which is a generalization of Moore's algorithm (see section 1).

### Heuristic

Jobs are added to the schedule in order of nondecreasing due dates (in case

of a tie, choose the one with smallest  $p_j/w_j$  ratio). If addition of job  $j$  results in this job being completed after  $d_j$ , look for job  $l$  already scheduled such that its removal creates a feasible on time set of jobs and the ratio  $p_l/w_l$  is the largest. Mark job  $l$  to be late and remove it from the schedule.

□

### Performance of the heuristic

It is easy to verify that the above heuristic runs in  $O(n \log n)$  time, being quite fast in practice.

When the weights are *agreeable* (i.e. when we have  $w_i \leq w_j$  for all jobs  $i$  and  $j$  such that  $p_i < p_j$ ), the heuristic yields an optimal solution. In fact, in this case it turns out to be the algorithm proposed by Lawler [1976]. The cases of *equal processing times* and *equal weights* are special cases of the previous one. Note that if the weights are equal the heuristic is simply Moore's algorithm (see section 1).

To verify the effectiveness of this heuristic on more general instances of the problem, computational tests have been performed on several randomly generated data sets, with up to 400 jobs. In general, the solution obtained  $z^H$  was very close to the optimal solution  $z^{OPT}$ . In the worst cases, the relative error  $(z^{OPT} - z^H)/z^{OPT}$  induced by the heuristic was never higher than 0.3 %.

A quite sharp bound on the optimal solution can therefore be obtained by this simple and efficient procedure.

## 7. SOME COMPUTATIONAL EXPERIENCE

### Implementation

According to the general framework described, the following basic approach has been implemented:

- i] Take the reformulated problem plus the whole set of constraints [MOO] (as described in section 3);
- ii] Solve the l.p. relaxation;
- iii] If an integer solution is obtained, STOP; else get a feasible solution through the *heuristic* algorithm of section 6, and let CUTOFF be the resulting lower bound on the optimal integer solution;
- iv] Plug the value CUTOFF into a branch-and-bound, and proceed until an optimal integer solution is proved to have been reached.

In order to improve this procedure, we tried to use inequalities (4.4) resulting from the dominance property [V+B], to cut fractional solutions off. In practice,

they appeared not to be useful, and their use was therefore abandoned.

### Generation of test problems

The random generation of data sets for the  $1||\sum w_j U_j$  problem was performed using the approach by Potts and Van Wassenhove [1986]. Recall that an instance of the problem is completely defined by the number  $n$  of jobs (variables), and by three  $n$ -vectors of processing times  $p$ , due dates  $d$  and weights  $w$ . The procedure is the following:

- i] Integer *processing times* are generated from the uniform distribution  $U[1, 100]$ ;
- ii] Given a *relative range of due dates* as a pair  $d_L$  and  $d_U$  (with  $d_L \leq d_U$ ), where  $d_L \in \{0.2, 0.4, 0.6, 0.8\}$  and  $d_U \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ , generate the *due dates* as integers from the uniform distribution  $U[Pd_L, Pd_U]$ ,  $P$  being the sum of all processing times  $\sum_{i=1}^n p_i$ ;
- iii] Different possible correlations between processing times and weights are considered, the *weights* being generated according to one of the following problem types:
  - 1] uncorrelated —  $w_j \in U[1, 100]$
  - 2] weakly correlated —  $w_j \in U[p_j + 1, p_j + 20]$
  - 3] strongly correlated —  $w_j = p_j + 20$ .

Therefore, for a fixed number of jobs  $n$ , 42 different problems may be defined, one for each combination of the 14 pairs  $d_L, d_U$  with the 3 correlation types.

### Computational results

Computations were carried out on a DATA GENERAL MV8000, using the mathematical programming system SCICONIC (version VM/D1.20). To create the input matrices in MPS standard format, a matrix generator (MGG) was used, allowing therefore a straightforward data specification.

Tests have been performed on randomly generated problems with 50, 100 and 200 jobs. The range of solution values was in the order of  $[20n, 60n]$ ,  $n$  being the number of jobs in the problem, the solution values depending obviously on the input parameters. The main results can be summarized as follows:

- i] The difference between the solution of the l.p. relaxation  $z^{LP}$  and the feasible integer solution  $z^H$  given by the heuristic (duality gap) was often less than 5 units, and was almost never higher than 10 units. In the above mentioned range of solution values (in the order of some thousand units), this means that  $(z^{LP} - z^H)/z^H$  was in general below 0.2 %. For the strongest correlated problems, this gap was even smaller.

- ii] However, the time needed by the branch-and-bound procedure was highly dependent on the data pattern and on the particular problem instance. On average, it took around half minute and 200 nodes of the search tree to solve problems with 50 variables. These figures grew up to two minutes and 500 nodes for 100 variable problems, and up to 8 minutes and 1000 nodes for 200 variable problems. Note however that for some particular data sets, these times could be quite small, and for others they reached much higher values.
- iii] The weakly correlated problems were the hardest to solve.

Due to the limitations of space in memory, for larger problem instances (say, with more than 300 jobs), the *alternative formulation* (see section 5) was used. Although having a larger number of variables, the space needed in core is kept within acceptable limits. Comparison tests showed that this formulation gives slightly weaker l.p. solutions, and take more time to solve the linear programs and to reach the optimal integer solutions. However these drawbacks are in some sense compensated by the time gained in the data preparation phase (i.e. in generating the input matrix for the mathematical programming system) which was in general, drastically reduced (by at least a factor of 10). With this alternative formulation, we were able to solve problems with up to 400 variables.

## 8. CONCLUSIONS AND FURTHER RESEARCH

The formulations described in this work present l.p. relaxations leading to solution values very close to the integer optimum. On the other hand, the heuristic algorithm we have used produces quite good feasible schedules, and therefore very sharp lower bounds on the *true* optimal solution are available.

In fact, as we manage to produce such good bounds, we would expect the branch-and-bound procedure to perform much better. The poor performance observed may in part be due to the fact that we were using a standard mathematical programming system, over which we had very limited control. Therefore, the definition and implementation of search policies for the enumeration procedure merits further research.

Note that the compactness of these formulations results mainly from the fact that we can model the  $1||\sum w_j U_j$  problem, using a small set of binary variables. For broader classes of scheduling problems, this is no longer possible. Therefore, we would like to design more general approaches. Using decision spaces based on the idea of *time discretization* seems to provide a promising framework in that direction.

Finally, we would like to emphasize the need for an interface allowing the decision-maker to interact with the computer system, and partially conduct the optimizing procedure. Such an interface could possibly be based on a standard spreadsheet software, taken as a tool to design Gantt charts, and therefore easily visualize schedules.

### Acknowledgement

The author is most grateful to Laurence Wolsey and to Yves Pochet for stimulating discussions and suggestions. Financial support from JNICT/INVOTAN is also gratefully acknowledged.

### REFERENCES

- Karp R.M. [1972], "Reducibility among combinatorial problems" in Miller R.E. (eds.) *Complexity of Computer Computations*, 85-103, Plenum Press, New York.
- Kise H., Ibaraki T., Mine H. [1978], "A solvable case of the one-machine scheduling problem with ready and due times", *Operations Research*, **26**, 121-126.
- Lawler E.L. [1964], "On Scheduling Problems with Deferral Costs", *Management Science*, **9**, 280-288.
- Lawler E.L. [1976], "Sequencing to Minimize the Weighted Number of Tardy Jobs", *RAIRO Recherche Operationelle*, **10 suppl.**, 27-33.
- Lawler E.L. [1983], "Recent Results in the Theory of Machine Scheduling", in Bachem A. et al. (eds.), *Mathematical Programming: The State of the Art - Bonn 1982*, 202-234, Springer, Berlin.
- Lawler E.L., Moore J.M. [1969], "A Functional Equation and Its Application to Resource Allocation and Sequencing Problems", *Management Science*, **16**, 77-84.
- Lawler E.L., Lenstra J.K., Rinnoy Kan A.H.G. [1982], "Recent Developments in Deterministic Sequencing and Scheduling", in Dempster M.A.H. et al. (eds.), *Deterministic and Stochastic Scheduling*, 35-73, D.Reidel Publ.Company, Dordrecht, Holland.
- Moore J.M. [1968], "An  $n$  Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs", *Management Science*, **15**, 102-109.
- Peters R. [1986], "L'ordonnancement sur une Machine avec des Contraintes de Délai", *Mémoire d'Ingénieur Civil*, Faculté des Sciences Appliquées, UCL, Louvain-la-Neuve, Belgium.
- Pochet Y. [1986], "Valid inequalities and Separation for capacitated economic lot sizing", *CORE Discussion Paper 8649*, UCL, Louvain-la-Neuve, Belgium.
- Potts C.N. and Van Wassenhove L.N. [1986], "Algorithms for scheduling a single machine to minimize the weighted number of late jobs", *to appear*.
- Villareal F.J. and Bulfin R.L. [1983], "Scheduling a Single Machine to Minimize the Weighted Number of Tardy Jobs", *IIE Transactions*, **15**, 337-343.

## THE SOLUTION OF THE LINEAR BILEVEL PROGRAMMING PROBLEM BY USING THE LINEAR COMPLEMENTARITY PROBLEM

J. J. Júdice\*  
A. M. Faustino\*\*

\*Dept. de Matemática, Universidade de Coimbra

\*\*Dept. de Matemática, Universidade do Porto

### Resumo

O Problema Linear de 2 níveis (LBLP) é um problema de optimização não convexa que aparece muito frequentemente em aplicações. Bialas e Karwan desenvolveram um algoritmo SLCP para a solução do LBLP que consiste em resolver uma sucessão de Problemas Lineares Complementares. Contudo, o algoritmo não é sempre convergente. Neste artigo é proposta uma versão modificada e convergente desse algoritmo SCLP. Experiência computacional com o algoritmo SCLP modificado na solução de LBLPs de média dimensão e estrutura esparsa é também incluída e mostra que o processo é eficiente.

### Abstract

The Linear Bilevel Programming Problem (LBLP) is a nonconvex optimization problem that occurs quite frequently in applications. Bialas and Karwan have designed a SLCP algorithm for the solution of the LBLP, which consists of solving a sequence of linear complementarity problems. However, the method is not convergent in general. In this paper we describe a modified convergent version of the SLCP algorithm. Computational experience with the modified SLCP algorithm on the solution of medium-scale LBLPs is also included and shows the efficiency of the approach.

**Keywords:** Bilevel Programming, Nonconvex Optimization, Linear Complementarity Problem, SLCP algorithms, sparse matrices.

### 1 - INTRODUCTION

The Bilevel Programming Problem (BLP) can be defined as

$$\left. \begin{array}{l} \text{Minimize } f_1(x,y) \\ y \in R^m \\ \text{subject to } g_i(x,y) \leq 0, \quad i = 1, \dots, l \end{array} \right\} \quad (1)$$

where  $x \in R^n$  is implicitly defined as the optimal solution of the program

$$\left. \begin{array}{l} \text{Minimize } f_2(x,y) \\ x \in R^n \\ \text{subject to } h_i(x,y) \leq 0, \quad i = 1, \dots, r \end{array} \right\} \quad (2)$$

and  $f_1, f_2, g_i$  and  $h_j$  are real functions of the variables  $x_j$  and  $y_j$ .

The BLP is nowadays a well-known nonconvex optimization problem. A large number of applications of the BLP have been proposed, primarily in economic planning [6, 11, 12]. A number of algorithms [2,3,4,5,6,7,9,12] have been developed for finding a global or a local minimum for the BLP.

In [7], Biales and Karwan study the linear BLP (LBLP), that is, the BLP in which all the functions are linear. They discuss a number of algorithms and recommend a Sequential Linear Complementarity Problem (SLCP) algorithm for the solution of LBLPs of small dimension ( $n \leq 50, m \leq 50$ ). This algorithm consists of solving a sequence of linear complementarity problems (LCP) by using a Restricted Basis Entry Simplex method similar to the Wolfe-Markowitz method [13,17] for quadratic programming. However, such scheme is not convergent in general [5].

In this paper we describe a modification of the SLCP algorithm capable of finding a global minimum for the LBLP. In this procedure, the LCPs required by the sequential algorithm are solved by an extension of Al-Khayyal's enumerative method [1]. We show that the computational effort of this hybrid enumerative method is highly reduced because of the special structure of the LCPs. We also present some computational experience with medium-scale LBLPs ( $n \leq 150$  and  $m \leq 250$ ) which shows the efficiency of the modified SLCP algorithm.

The organization of the paper is as follows. In Section 2 Biales and Karwan SLCP algorithm for the LBLP is described. In Section 3 the incorporation of the hybrid enumerative method in the SLCP algorithm is discussed. Finally computational experience on the solution of medium-scale LBLPs by the modified SLCP method is presented in Section 4.

## 2 - SLCP ALGORITHM FOR THE LBLP

Consider the LBLP as stated in [7]

$$\left. \begin{array}{l} \text{Minimize } c^T x + d^T y \\ y \in R_+^m \end{array} \right\} \quad (3)$$

where  $R_+^m = \{y \in R^m : y \geq 0\}$  and  $x \in R^n$  solves the linear program

$$\left. \begin{array}{l} \text{Minimize } a^T x \\ x \in R^n \\ \text{subject to } A_1 x + A_2 y \geq b, x \geq 0 \end{array} \right\} \quad (4)$$

where  $A_1 \in R^{n \times n}$  and  $A_2 \in R^{n \times m}$ . As in [7], we assume that the set

$$H = \{(x,y) \in R^{n \times m} : A_1 x + A_2 y \geq b, x \geq 0, y \geq 0\} \tag{5}$$

is bounded, so that the LBLP has an optimal solution [7].

Consider the dual of the linear program (4)

$$\left. \begin{array}{l} \text{Maximize} \quad (b - A_2 y)^T u \\ \text{subject to} \quad A_1^T u \leq a \\ \quad \quad \quad u \geq 0 \end{array} \right\} \tag{6}$$

If  $\alpha$  and  $\beta$  are the vectors of the slack variables corresponding to the primal and dual respectively, then by the complementarity slackness property [14, pages 165-167], the LBLP is equivalent to the following nonconvex optimization problem

$$\left. \begin{array}{l} \text{Minimize} \quad c^T x + d^T y \\ \text{subject to} \quad \alpha = -b + A_1 x + A_2 y \\ \quad \quad \quad \beta = a - A_1^T u \\ \quad \quad \quad x, y, u, \alpha, \beta \geq 0, \quad x^T \beta = u^T \alpha = 0 \end{array} \right\} \tag{7}$$

Hence a global minimum  $(\bar{x}, \bar{y})$  of the LBLP can be found by solving the nonconvex program (7). Bialas and Karwan [7] propose a SLCP algorithm for such purpose. In this method a parameter  $\lambda$  is introduced and the objective function is replaced by the constraint  $c^T x + d^T y \leq \lambda$  to obtain the following parametric LCP

LCP( $\lambda$ ):

$$\left. \begin{array}{l} \left[ \begin{array}{c} \alpha \\ \beta \\ v_0 \end{array} \right] = \left[ \begin{array}{c} -b \\ a \\ 0 \end{array} \right] + \left[ \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right] \lambda + \left[ \begin{array}{ccc} 0 & A_1 & A_2 \\ -A_1^T & 0 & 0 \\ 0 & -c^T & -d^T \end{array} \right] \left[ \begin{array}{c} u \\ x \\ y \end{array} \right] \\ x, y, u, \alpha, \beta, v_0 \geq 0, \quad x^T \beta = u^T \alpha = 0 \end{array} \right\} \tag{8}$$

The SLCP algorithm consists of finding the solution of the LCP( $\bar{\lambda}$ ), where  $\bar{\lambda}$  is the smallest value of  $\lambda$  such that LCP( $\lambda$ ) has a solution. To achieve this, the method solves a sequence of LCP( $\lambda_k$ ), where  $\{\lambda_k\}$  is a decreasing sequence defined by

$$\left. \begin{aligned} \lambda_0 &= \text{upper-bound of } c^T x + d^T y \text{ on } H \\ \lambda_k &= c^T x^{k-1} + d^T y^{k-1} - |\gamma_k(c^T x^{k-1} + d^T y^{k-1})| \end{aligned} \right\} \quad (9)$$

with  $(x^{k-1}, y^{k-1})$  the solution of  $LCP(\lambda_{k-1})$  and  $\gamma_k$  a small positive number. The method terminates in a iteration  $k$  such that  $LCP(\lambda_k)$  has no solution. When this occurs, the solution  $(x^{k-1}, y^{k-1})$  of the  $LCP(\lambda_{k-1})$  satisfies

$$0 \leq c^T x^{k-1} + d^T y^{k-1} - \text{VAL} \leq |\gamma_k(c^T x^{k-1} + d^T y^{k-1})| \quad (10)$$

where  $\text{VAL}$  is the value of the objective function at the optimal solution. Hence, if the set  $H$  defined by (5) is nonempty and bounded, the algorithm finds an  $\epsilon$ -optimal solution of the LBLP, where

$$\epsilon = |\gamma_k(c^T x^{k-1} + d^T y^{k-1})| \quad (11)$$

In practice, if  $\gamma_k$  is quite small, the solution  $(x^{k-1}, y^{k-1})$  of the last  $LCP(\lambda_{k-1})$  is usually the optimal solution of the LBLP. Bialas and Karwan [7] recommend the choice of  $\gamma_k = 0.01$  in each iteration  $k$  and our experience has confirmed that this choice is usually the most suitable.

The steps of the SLCP algorithm are presented below.

### SLCP Algorithm

STEP 0 - Let  $k = 0$  and  $\lambda_0$  be an upper-bound of  $c^T x + d^T y$  on  $H$ .

General Step - Solve the  $LCP(\lambda_k)$ . If  $LCP(\lambda_k)$  has no solution, go to Exit.

Otherwise let  $(x^k, y^k)$  be the solution of this LCP. Set

$$\lambda_{k+1} = c^T x^k + d^T y^k - |\gamma_k(c^T x^k + d^T y^k)| \quad (12)$$

where  $\gamma_{k+1}$  is a fixed value.

Set  $k = k + 1$  and repeat.

EXIT - If  $k = 0$  the LBLP is infeasible. Otherwise  $(x^{k-1}, y^{k-1})$  is an  $\epsilon$ -optimal solution of the LBLP, where  $\epsilon$  is given by (11).

The efficiency of the SLCP algorithm depends essentially on the procedure to solve the  $LCP(\lambda_k)$ . Biales and Karwan [7] recommend the use of a Basis Restricted Entry Simplex (BRES) method for this purpose. They do not provide sufficient details for the complete understanding of the algorithm, but this kind of method can not always be used to process the  $LCP(\lambda_k)$  [5]. Next we describe a BRES algorithm with exactly one artificial variable. Consider the  $LCP(\lambda_k)$  in the following form

$$\left. \begin{aligned} w &= q + Mz + Ny, \quad w \geq 0, z \geq 0, y \geq 0 \\ z_i w_i &= 0, \quad i = 1, \dots, r+n \end{aligned} \right\} \quad (13)$$

where

$$w = \begin{bmatrix} \alpha \\ \beta \\ v_0 \end{bmatrix} \in \mathbb{R}^{r+n+1}, \quad z = \begin{bmatrix} u \\ x \end{bmatrix} \in \mathbb{R}^{r+n}, \quad q = \begin{bmatrix} -b \\ a \\ \lambda_k \end{bmatrix} \in \mathbb{R}^{r+n+1}$$

$$M = \begin{bmatrix} 0 & A_1 \\ -A_1^T & 0 \\ 0 & -c^T \end{bmatrix} \in \mathbb{R}^{(r+n+1) \times (r+n)}, \quad N = \begin{bmatrix} A_2 \\ 0 \\ -d^T \end{bmatrix} \in \mathbb{R}^{(r+n+1) \times m}$$

Let  $z_0$  be an artificial variable and  $p$  be a nonnegative vector satisfying  $p_i > 0$  for all  $i$  such that  $q_i < 0$ . Consider the following program

$$\left. \begin{aligned} \text{Minimize} \quad & z_0 \\ \text{subject to} \quad & w = q + z_0 p + Mz + Ny \\ & z, w, y, z_0 \geq 0 \\ & z_i w_i = 0, \quad i = 1, \dots, r+n \end{aligned} \right\} \quad (14)$$

The BRES algorithm is an extension of the Phase 1 method with a single artificial variable [14, pages 136-138], and seeks the optimal solution of the program (14) by only using complementary feasible solutions (that is, solutions satisfying the constraints of (14)). To assure this condition, a nonbasic variable  $z_i$  (or  $w_i$ ) with a positive reduced cost coefficient can only be candidate to be an entering variable if its complementary  $w_i$  (or  $z_i$ ) is also nonbasic or becomes nonbasic in this iteration. Because of this additional criterion, the BRES algorithm may have three possible terminations:

TERM = 1 - an optimal solution  $(\bar{w}, \bar{z}, \bar{y})$  of (14) with  $\bar{z}_o = 0$ .

TERM = 2 - an optimal solution  $(\bar{w}, \bar{z}, \bar{y})$  of (14) with  $\bar{z}_o > 0$ .

TERM = 3 - a basic solution which is not optimal for (14) but there are no candidates for an entering variable.

In the first case (TERM = 1)  $(\bar{w}, \bar{z}, \bar{y})$  is a solution of the LCP (13). If TERM = 2 this LCP has no solution. In the latter case (TERM = 3) no conclusion can be made about the existence of a solution to the LCP(13). In appendix we present an example of a LCP of the form (13) for which there is a complementary solution but the BRES algorithm terminates with TERM = 3. This shows that the BRES method can not be used in general to solve the LCPs( $\lambda_k$ ) that the SLCP algorithm requires. However, the BRES method can be incorporated in a hybrid enumerative algorithm and this usually improves the efficiency of such procedure. This is discussed in the next section.

**3 - A HYBRID ENUMERATIVE METHOD FOR THE LCP( $\lambda_k$ )**

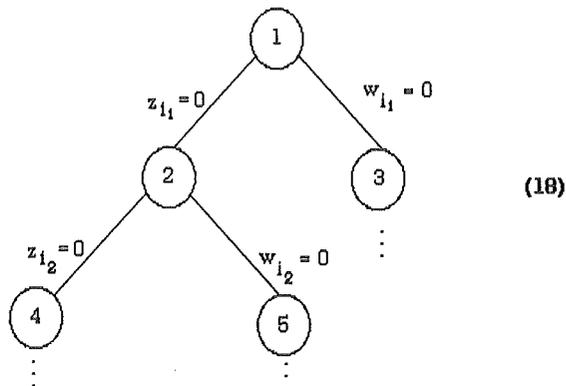
In this section we describe a hybrid enumerative method capable of solving any LCP( $\lambda_k$ ) required by the SLCP algorithm. Consider a LCP( $\lambda_k$ ) in the form

$$w = q + Mz + Ny \tag{15}$$

$$w \geq 0, z \geq 0, y \geq 0 \tag{16}$$

$$z_i w_i = 0, i = 1, \dots, r+n \tag{17}$$

where  $w \in R^{r+n+1}$ ,  $z \in R^{r+n}$  and  $y \in R^m$ . As in linear programming, a solution  $(z,w,y)$  satisfying the linear constraints (15) and (16) is called Feasible. A solution is Complementary if the variables  $z_i$  and  $w_i$  satisfy (17). An enumerative method attempts to find a complementary solution by only using basic feasible solutions of the system (15). To achieve this, the following tree is explored



where  $i_1, i_2, \dots$  are integer numbers of  $\{1, \dots, r+n\}$ . In node 1 an initial feasible solution is obtained by a variant of the Phase 1 algorithm with a single artificial variable, as is discussed later. Each one of the other nodes is generated by solving a subproblem, which consists of minimizing a variable  $z_i$  or  $w_i$  subject to the linear constraints of the LCP and some constraints  $z_j = 0$  or  $w_j = 0$ . For instance, to generate node 4 of the tree (18) it is necessary to solve the linear program

$$\left. \begin{array}{l} \text{Minimize } z_{i_2} \\ \text{subject to } w = q + Mz + Ny, z \geq 0, w \geq 0, y \geq 0, \\ z_{i_1} = 0 \end{array} \right\} \quad (19)$$

Such linear program is solved by a modification of the Phase 2 of the simplex method and two cases may occur:

- (i) if the variable minimized has value equal to zero, then it is fixed at zero in all descendent paths of the tree (the variable is said to be starred),
- (ii) if the minimum value of the variable is positive, then the branch is pruned and the node is fathomed.

The enumerative method attempts to solve the LCP by generating successive nodes of the tree according to the process explained above. The algorithm either finds a solution of the LCP (it is the first complementary feasible solution) or establishes that the LCP has no solutions (all the nodes of the tree are fathomed).

The enumerative method is efficient if there are only few nodes to be visited before a complementary solution is found or it is verified that none exists. There exist some heuristic rules and some procedures that usually improve the efficiency of the algorithm. The heuristic rules are related with the choice of the node of the tree and of the pair  $(z_{i_k}, w_{i_k})$  and are presented later. Next we describe the extensions for the LCP (13) of two efficient algorithms [10] that are usually incorporated in the enumerative method [10].

**(1) A MODIFIED BRES ALGORITHM**

Consider the problem of generating the node 1 of the tree (18), that is, finding a first feasible solution for the LCP (13). As stated before, this can be achieved by solving the linear program

$$\begin{array}{ll}
 \text{Minimize} & z_0 \\
 \text{subject to} & w = q + z_0 p + Mz + Ny \\
 & z, w, y, z_0 \geq 0
 \end{array} \quad \left. \vphantom{\begin{array}{l} \text{Minimize} \\ \text{subject to} \end{array}} \right\} \quad (20)$$

where  $p$  is a nonnegative vector satisfying  $p_i > 0$  for all  $i$  such that  $q_i < 0$  and  $z_0$  is an artificial variable. Since our ultimate goal is to find a complementary feasible solution, then the BRES algorithm can be used to solve such linear program. If this algorithm terminates with  $TERM = 1$  or  $TERM = 2$ , then a solution of the LCP (13) has been found or LCP has no solutions respectively, and the enumerative method stops. If  $TERM = 3$ , the value of  $z_0$  can still be reduced if a simplex pivot step is performed, where the pivot belongs to a column with a positive reduced cost coefficient. The modified BRES algorithm consists of performing such pivot step when  $TERM = 3$  and reapplying the BRES method by starting with the basic solution obtained by this pivot step. The process is then repeated. This algorithm can also be used to generate any other node  $k$ , since, as discussed before, the generation of a node consists of minimizing a variable  $z_i$  or  $w_i$ .

To describe the modified BRES algorithm, we associate to each basic solution a quantity NCP defined as the number of pairs of complementary variables  $(z_i, w_i)$  such that  $z_i$  and  $w_i$  are both basic in this solution. If  $NCP = 0$  the basic solution is complementary. The converse is true for nondegenerate basic solutions, but a degenerate basic solution may be complementary with  $NCP > 0$ . Hence the value of NCP is increased by one whenever  $TERM = 3$  occurs in the BRES algorithm. In addition, NCP never increases and may even be reduced during the BRES procedure. Hence a complementary feasible solution may occur during this algorithm. The steps of the modified BRES algorithm are presented below.

---

#### **A Modified BRES Algorithm for Generating a node $k$**

---

STEP 0 - Let NCP be the number of pairs of complementary variables  $(z_i, w_i)$  such that  $z_i$  and  $w_i$  are both basic in the basic solution ( $NCP = 0$  if  $k = 1$ ).

STEP 1 - Apply the BRES algorithm. If a complementary feasible solution is found when the BRES algorithm is in use, set  $TERM = 1$  and  $NCP = 0$  and go to Exit. If the BRES algorithm terminates with  $TERM = 1$  or  $TERM = 2$ , go to Exit. Otherwise ( $TERM = 3$ ) go to step 2.

STEP 2 - Set  $NCP = NCP + 1$  and perform a simplex pivot step, where the pivot belongs to a column with a positive reduced cost coefficient. Go to step 1.

EXIT - If  $TERM = 1$  and  $NCP = 0$ , a solution of the LCP has been found and the enumerative method stops. If  $TERM = 1$  and  $NCP > 0$ , the node  $k$  has been generated. If  $TERM = 2$ , the node  $k$  can not be generated and is fathomed.

(II) A MODIFIED REDUCED-GRADIENT (MRG) ALGORITHM

This algorithm is described in [1] and consists of finding a local star minimum of the function

$$f(z,w,y) = \sum_{i=1}^{r+n} z_i w_i \tag{21}$$

that is, a basic feasible solution  $(\bar{z}, \bar{w}, \bar{y})$  such that

$$f(\bar{z}, \bar{w}, \bar{y}) \leq f(z,w,y) \tag{22}$$

for all its adjacent basic feasible solutions  $(z,w,y)$ . To describe the algorithm for the case of the LCP(13), consider a basic feasible solution  $(\bar{z}, \bar{w}, \bar{y})$  with Basis  $B$ . If  $(\check{z}, \check{w}, \check{y})$  is an adjacent basic solution of  $(\bar{z}, \bar{w}, \bar{y})$ , then

$$(\check{z}, \check{w}, \check{y}) = (\bar{z}, \bar{w}, \bar{y}) + \mu_0 (d^z, d^w, d^y) \tag{23}$$

where  $\mu_0$  is the minimum ratio of the simplex method and  $d = (d^z, d^w, d^y)$  is a feasible direction such that  $d^z, d^w$  and  $d^y$  are the vectors of the components of  $d$  corresponding to the variables  $z,w$  and  $y$  respectively. To define this direction, let  $F$  and  $T$  be the sets of the indices of the basic and nonbasic variables respectively and  $s$  be the index of the nonbasic entering variable that is increased to generate the basic feasible solution  $(\check{z}, \check{w}, \check{y})$ .

Then

$$d_s = 1$$

$$d_j = 0 \text{ for all } j \in T - (s)$$

$$d_F = \begin{cases} -B^{-1}M_{.s} & \text{if } s \text{ is a column of a } z \text{ variable} \\ B^{-1}I_{.s} & \text{if } s \text{ is a column of a } w \text{ variable} \\ -B^{-1}N_{.s} & \text{if } s \text{ is a column of a } y \text{ variable} \end{cases}$$

where  $M_{.s}$ ,  $N_{.s}$  and  $I_{.s}$  represent the column  $s$  of the matrices  $M, N$  and identity matrix  $I$  respectively. By the structures of the matrices  $M$  and  $N$  of the LCP(13), we can write

$$P [I \ -M \ -N] Q = \left[ \begin{array}{c|c} E & 0 \\ \hline 0 & G \end{array} \right] \quad (24)$$

where  $P$  and  $Q$  are permutation matrices and  $E \in R^{(r+1) \times (r+n \cdot m+1)}$ ,  $G \in R^{n \times (r+n)}$  are matrices such that  $E$  contains a column of the variable  $z_i(w_i)$  if and only if  $G$  contains a column of the variable  $w_i(z_i)$ . Hence

$$PBQ = \left[ \begin{array}{c|c} B_1 & 0 \\ \hline 0 & B_2 \end{array} \right] \quad (25)$$

where  $B_1 \in R^{(r+1) \times (r+1)}$  and  $B_2 \in R^{n \times n}$  are nonsingular matrices, and this implies that

$$d = \left[ \begin{array}{c} 0 \\ \hline 1 \\ B_2^{-1}G_{.s} \end{array} \right] \quad \text{or} \quad d = \left[ \begin{array}{c} 1 \\ B_1^{-1}E_{.s} \\ \hline 0 \end{array} \right] \quad (26)$$

Therefore

$$d_i^w = 0 \quad \text{or} \quad d_i^z = 0 \quad \text{for each } i = 1, \dots, r+n \quad (27)$$

Computing the value of the function (21) at the adjacent basic feasible solution given by (23), we have

$$f(\bar{z}, \bar{w}, \bar{y}) = \sum_{i=1}^{r+n} \bar{z}_i \bar{w}_i = \sum_{i=1}^{r+n} (\bar{z}_i + \mu_0 d_i^z) (\bar{w}_i + \mu_0 d_i^w)$$

$$= \sum_{i=1}^{r+n} \bar{z}_i \bar{w}_i + \mu_0 \sum_{i=1}^{r+n} (\bar{z}_i d_i^w + \bar{w}_i d_i^z) + \mu_0^2 \sum_{i=1}^{r+n} d_i^z d_i^w$$

Hence by (27) we have

$$f(\bar{z}, \bar{w}, \bar{y}) - f(\bar{z}, \bar{w}, \bar{y}) = \mu_0 \sum_{i=1}^{r+n} (\bar{z}_i d_i^w + \bar{w}_i d_i^z) \tag{28}$$

But

$$\sum_{i=1}^{r+n} (\bar{z}_i d_i^w + \bar{w}_i d_i^z)$$

is exactly the symmetric of the reduced-cost coefficient  $\bar{e}_s$  of the nonbasic variable of index  $s$  associated with the linear function

$$\sum_{i=1}^{r+n} (\bar{w}_i z_i + \bar{z}_i w_i) \tag{29}$$

Therefore we have shown that for the  $LCP(\lambda_k)$  Al-Khayyal's MRG Algorithm reduces to a simplex-type method in which the reduced-cost coefficients  $\bar{e}_j$  of the nonbasic variables are associated with the linear function (29). The algorithm also incorporates a modification of Bland's rule [8] to avoid the occurrence of cycling and is presented below.

---

**MRG Algorithm**

---

STEP 0 - Let  $(\bar{z}, \bar{w}, \bar{y})$  be a basic feasible solution with Basis B.

Set  $\text{deg} = 0$ .

STEP 1 - Compute the reduced-cost coefficients  $\bar{e}_j$  associated with the linear function (29). If  $\bar{e}_j \leq 0$  for all  $j$ , then  $(\bar{z}, \bar{w}, \bar{y})$  is a local star minimum and stop. Otherwise go to step 2.

STEP 2 - Find the entering variable as the nonbasic variable in the column  $s$  given by

$$s = \begin{cases} \min \{i : \bar{e}_i = \max \bar{e}_j\} & \text{if } \text{deg} = 0 \\ \min \{i : \bar{e}_i > 0\} & \text{if } \text{deg} = 1 \end{cases} \tag{30}$$

STEP 3 - Compute the minimum ratio  $\mu_0$  for this entering variable and let the leaving variable be the first basic variable at which  $\mu_0$  is attained. If  $\mu_0 = 0$  ( $\mu_0 > 0$ ) set  $\text{deg} = 1$  ( $\text{deg} = 0$ ). Perform a simplex pivot step and obtain a basic feasible solution  $(\bar{z}, \bar{w}, \bar{y})$  with Basis B. Go to step 1.

As described above, the MRG algorithm always terminates in a local star minimum  $(\bar{z}, \bar{w}, \bar{y})$  of the function  $f(z,w,y)$  given by (21). The method is quite suitable to be incorporated in a hybrid enumerative method, since the value of the complementarity function  $\sum_{i=1}^{r+n} z_i w_i$  is usually reduced and may take the value zero, in which case a solution of the LCP(13) is found.

The hybrid enumerative method, incorporating the two procedures referred in this section and the most efficient heuristic rules, is stated below.

#### Hybrid Enumerative Method for Solving LCP(13)

STEP 0 - Set  $\text{NODE} = \text{NNODE} = 1$ , where  $\text{NNODE}$  is the total number of nodes to be investigated and  $\text{NODE}$  is the current node. Set  $\text{STZ}(\text{NODE}) = \text{STW}(\text{NODE}) = \emptyset$ , where  $\text{STZ}(\text{NODE})$  and  $\text{STW}(\text{NODE})$  are the sets of the indices of the  $z$  and  $w$  starred variables at node  $\text{NODE}$  respectively.

STEP 1 - Generation of node 1 (First feasible solution).

(i) Apply the modified BRES algorithm for solving the linear program (20). If  $\text{TERM} = 1$  and  $\text{NCP} = 0$ , go to Exit. If  $\text{TERM} = 2$  go to Exit. Otherwise ( $\text{TERM} = 1$  and  $\text{NCP} > 0$ ) go to (ii).

(ii) Apply the MRG algorithm to find a local star minimum  $(\bar{z}, \bar{w}, \bar{y})$  of the function  $f(z,w,y)$  given by (21). If  $f(\bar{z}, \bar{w}, \bar{y}) = 0$ , set  $\text{TERM} = 1$  and go to Exit. Otherwise set  $\text{NCP}(\text{NODE}) = \text{NCP}$  and go to step 2.

STEP 2 - Branching - Set  $\text{NCP} = \text{NCP}(\text{NODE})$ ,  $\text{STZ} = \text{STZ}(\text{NODE})$  and  $\text{STW} = \text{STW}(\text{NODE})$ . Choose the pair of complementary variables  $(z_s, w_s)$  in node  $\text{NODE}$  such that

$$\bar{z}_s \bar{w}_s = \max \{ \bar{z}_i \bar{w}_i : i = 1, \dots, \text{NCP} \} \quad (31)$$

Fathom node NODE and go to step 3.

STEP 3 - Generate NNODE+1 - Set NODE = NNODE = NNODE + 1.

(i) Apply the modified BRES algorithm for solving the linear program

$$\left. \begin{aligned} & \text{Minimize } z_s \\ & \text{subject to } w = q + Mz + Ny, z, w, y \geq 0 \\ & z_j = 0, w_\lambda = 0, j \in \text{STZ}, \lambda \in \text{STW} \end{aligned} \right\} \quad (32)$$

If TERM = 1 and NCP = 0, go to Exit. If TERM = 2, fathom node NODE and go to step 4. Otherwise set STZ(NODE) = STZ U {s} and go to (ii).

(ii) Apply the MRG algorithm to find a local star minimum  $(\bar{z}, \bar{w}, \bar{y})$  of the function  $f(z, w, y)$  given by (21). If  $f(\bar{z}, \bar{w}, \bar{y}) = 0$ , set TERM = 1 and go to Exit. Otherwise set NCP (NODE) = NCP,  $f(\text{NODE}) = \sum_{i=1}^{\text{NCP}} \bar{w}_i \bar{z}_i$  and go to step 4.

STEP 4 - Generate node NNODE + 2 - This is step 3 with  $z_s$ , STZ and step 4 replaced by  $w_s$ , STW and step 5, respectively.

STEP 5 - Choice of the node - If all the nodes 1 ..., NNODE are fathomed, set TERM = 2 and go to Exit. Otherwise choose the node NODE by

$$\text{NCP}(\text{NODE}) + \frac{f(\text{NODE})}{10} = \min \left\{ \text{NCP}(j) + \frac{f(j)}{10}; j = 1, \dots, \text{NNODE} \text{ and } j \text{ is unfathomed} \right\} \quad (33)$$

and go to step 2.

EXIT - If TERM = 1, the complementary basic feasible solution  $(\bar{z}, \bar{w}, \bar{y})$  is a solution of the LCP(13). If TERM = 2, the LCP(13) has no solution.

We note that (31) and (33) are heuristic rules that we have found efficient to reduce the computational effort of the algorithm [10]. The hybrid enumerative method can be implemented for solving large and sparse LCPs. The implementation uses reinversion and updating techniques for the LU decomposition of the Basis matrices [16] and special data structures for the representation of the sets discussed above. We suggest [10] for a description of this procedure.

As discussed in section 2, the hybrid enumerative method is used to solve the

LCPs ( $\lambda_k$ ) required by the SLCP algorithm. For any two values  $\lambda_k < \lambda_{k-1}$ , the LCP( $\lambda_{k-1}$ ) and LCP( $\lambda_k$ ) only differ in the  $r$ -th component of the vector  $q$ . Furthermore, if  $B$  is the Basis associated to the solution  $(\bar{z}, \bar{w}, \bar{y})$  of the LCP( $\lambda_{k-1}$ ) obtained by the hybrid enumerative method and  $q$  is the right-hand side vector of the LCP( $\lambda_k$ ), then the vector  $\bar{q} = B^{-1}q$  satisfies

$$\bar{q}_j \geq 0 \text{ for all } j = 1, \dots, r+n \text{ and } \bar{q}_{r+n+1} < 0 \quad (34)$$

Hence the solution  $(\bar{z}, \bar{w}, \bar{y})$  of the LCP( $\lambda_{k-1}$ ) can be used as the initial basic solution for the LCP( $\lambda_k$ ). Since the vector  $B^{-1}q$  satisfies (34), then only the last component of the vector  $B^{-1}p$  has to be positive. Hence we can choose in (20)

$$p = B \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (35)$$

where  $0 \in \mathbb{R}^{r+n}$  is the null vector. The choice of this initial basic solution usually provides great reductions in the hybrid enumerative method. In fact, the algorithm requires quite often a unique node and a quite small number of pivot steps to find the solution of the LCP( $\lambda_k$ ). This is shown by the computational results presented in the next section.

In the original SLCP algorithm the LCP( $\lambda_0$ ) is solved in step 0, where  $\lambda_0$  is an upper-bound of  $c^T x + d^T y$  on  $H$ . The computation of this value of  $\lambda_0$  may be expensive in certain cases. Next, we present a modified step 0 of the SLCP algorithm, which does not require the computation of this upper-bound.

Modified Step 0 - Set  $k = 0$  and solve the LCP(0) with  $v_0$  unrestricted in sign. If this LCP has no solution go to Exit. Otherwise let  $(x^0, y^0)$  be the solution of this LCP and  $\lambda_0 = c^T x^0 + d^T y^0$  be the value of the variable  $v_0$ . Compute  $\lambda_1$  by (12).

Since in this step the variable  $v_0$  is unrestricted in sign and is originally basic, then it remains basic during the solution of the LCP(0). Hence this LCP is exactly the LCP( $\lambda_0$ ) without the constraint  $c^T x + d^T y \leq \lambda_0$ . Since the value of  $\lambda_0$  obtained by the Modified Step 0 is usually smaller than the upper-bound chosen by the original SLCP algorithm, then this modification reduces the number of LCPs to be solved and improves the efficiency of the SLCP algorithm. This is another important reason for the use of the Modified Step 0.

#### 4. COMPUTATIONAL EXPERIENCE

In this section we present some computational experience with the SLCP

algorithm described in this paper on the solution of some medium-scale sparse LBLPs. The matrix  $A = [A_1, A_2]$  of the test problems has been constructed by an extension for sparse matrices of the technique described in [15]. This modification consists of introducing a quantity  $NZ$ , representing the number of nonzero elements of the matrix  $A$ , and randomly distributing these nonzero elements in a scheme similar to [15]. We have also considered in the matrix  $A$  a dense row of ones to make bounded the set  $H$  defined by (5). We associate to each matrix  $A$  two test problems, which differ in the cost coefficients. In the problems denoted by TPN, all the cost coefficients  $c, d$ , and  $a$  of the two linear functions are nonnegative. In the problems denoted by TPG, the cost coefficients  $c$  and  $d$  of the function of the first level are nonnegative but the linear function of the second level has some negative cost coefficients. By doing this we simulate LBLPs with conflicting (problems TPG) and nonconflicting (problems TPN) objectives.

The characteristics of the test problems and the results of the experiences of solving these problems by the SCLP algorithm are presented in Table 1. All the tests have been performed on a CDC CYBER 180-830 of the University of Porto. In these tables there are some parameters with the following meanings:

$n$  = number of second level variables  $x$

$m$  = number of first level variables  $y$

$r$  = number of constraints = number of rows of  $A = [A_1, A_2]$

$rsm$  = relative sparsity of  $A = [A_1, A_2] = \frac{\text{number of nonzeros of } A}{n + m}$

$nrow$  = number of rows of the  $LCP(\lambda_k) = r + n + 1$

$ncol$  = number of columns of the  $LCP(\lambda_k) = r + n + m$

NLCP = number of LCPs( $\lambda_k$ ) to be solved by the SLCP algorithm.

NI = Total number of simplex pivot steps required by the SLCP algorithm.

NIMIN = minimum number of pivot steps that a  $LCP(\lambda_k)$  has required.

NIMAX = maximum number of pivot steps that a  $LCP(\lambda_k)$  with a solution has required.

T = CPU time in seconds for the SLCP algorithm.

The results presented in Table 1 show that it is worthwhile to solve medium-scale LBLPs by the SLCP algorithm discussed in this paper. As expected the efficiency of the

TP	n	m	r	rsm	nrow	ncol	NLCP	NI	NIMIN	NIMAX	T
TPN1	30	50	30	3.5	61	110	14	54	3	5	.96
TPG1							55	249	3	8	5.5
TPN2				5.0			27	315	3	40	10.3
TPG2							80	475	3	42	15.
TPN3	50	120	50	5.2	101	220	44	230	3	51	9.8
TPG3							124	529	3	7	32.5
TPN4				7.5			120	964	3	13	59.4
TPG4							163	1547	3	97	124.
TPN5	100	300	100	5.3	201	500	139	1009	3	70	107.
TPG5							247	2207	3	203	308.
TPN6				7.1			126	682	3	17	79.9
TPG6							268	1899	3	20	380.
TPN7	150	250	150	5.3	301	550	77	1934	3	115	128.
TPG7							214	3775	3	450	889.
TPN8				7.1			127	4740	3	1365	726.
TPG8							189	9170	3	194	2198.

TABLE 1

SLCP algorithm decreases with an increase of the dimension and density of the LBLP. The method performs much better for the LBLPs with nonconflicting objectives (problems TPN).

In all the experiences presented in Table 1, we have fixed  $\gamma_k = 0.01$  in the SLCP algorithm. We have also tested larger values of  $\gamma_k$  for the first LCPs( $\lambda_k$ ). This usually reduces the number of LCPs to be solved by the SLCP algorithm. However, the solutions of these LCPs usually require more computational work and this reduces the overall efficiency of the SLCP algorithm.

The results of the experiences also show that the number of LCPs to be solved by the SLCP algorithm is not too large. Furthermore the values of NIMIN and NIMAX are small and show that the hybrid enumerative method is efficient for solving the LCPs required by the SLCP algorithm. As expected the hybrid enumerative method performs

much worse when the last  $LCP(\lambda_k)$  is solved. Since this LCP has no solution, then the algorithm stops only when all the nodes are fathomed and this requires a great amount of tree search. Our experience has showed that the computational effort for solving this last LCP is similar to the computational work performed by the SLCP algorithm on the solution of all the remaining LCPs for the smaller Bilevel programs and is even bigger for the larger problems.

As a final conclusion, we can claim that the SLCP method described in this paper is an efficient and convergent algorithm for solving the LBLP. The efficiency of the SLCP method can be substantially improved if it is possible to design an efficient procedure to establish that the last  $LCP(\lambda_k)$  has no solution. This last  $LCP(\lambda_k)$  may even be unnecessary if the SLCP algorithm is seen as a process to provide a good incumbent for the branch-and-bound method described in [5]. These are two topics of our current research.

### APPENDIX

In this appendix we present an example in which the BRES algorithm fails to find a solution of a  $LCP(\lambda_k)$ . Consider the LBLP stated in [7]. Since  $x_1 \geq 0$ , then 0 is an upper-bound of the objective function  $-x_1$  and the  $LCP(0)$  is firstly solved. To solve this problem by the BRES algorithm, we consider  $p = (1,0,0,0,0,0,0)^T$  and obtain the following initial tableau

BASIC VARIABLES	NONBASIC VARIABLES									BASIC VALUES
	$z_0$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$y_1$		
$w_1$	-1	0	0	0	0	0	-2	-1	-10	
$w_2$	0	0	0	0	0	0	-2	1	6	
$w_3$	0	0	0	0	0	0	-1	2	21	
$w_4$	0	0	0	0	0	0	2	1	38	
$w_5$	0	0	0	0	0	0	2	-1	18	
$w_6$	0	2	2	1	-2	-2	0	0	1	
$w_7$	0	0	0	0	0	0	-1	0	0	

A complementary basic feasible solution for the program (14) is obtained by performing the simplex pivot step whose pivot is circled in the tableau. The tableau associated with this basic solution is given below.

	$w_1$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$y_1$	
$z_0$	-1	0	0	0	0	0	2	1	10
$w_2$	0	0	0	0	0	0	-2	①	6
$w_3$	0	0	0	0	0	0	-1	2	21
$w_4$	0	0	0	0	0	0	2	1	38
$w_5$	0	0	0	0	0	0	2	-1	18
$w_6$	0	2	2	1	-2	-2	0	0	1
$w_7$	0	0	0	0	0	0	-1	0	0

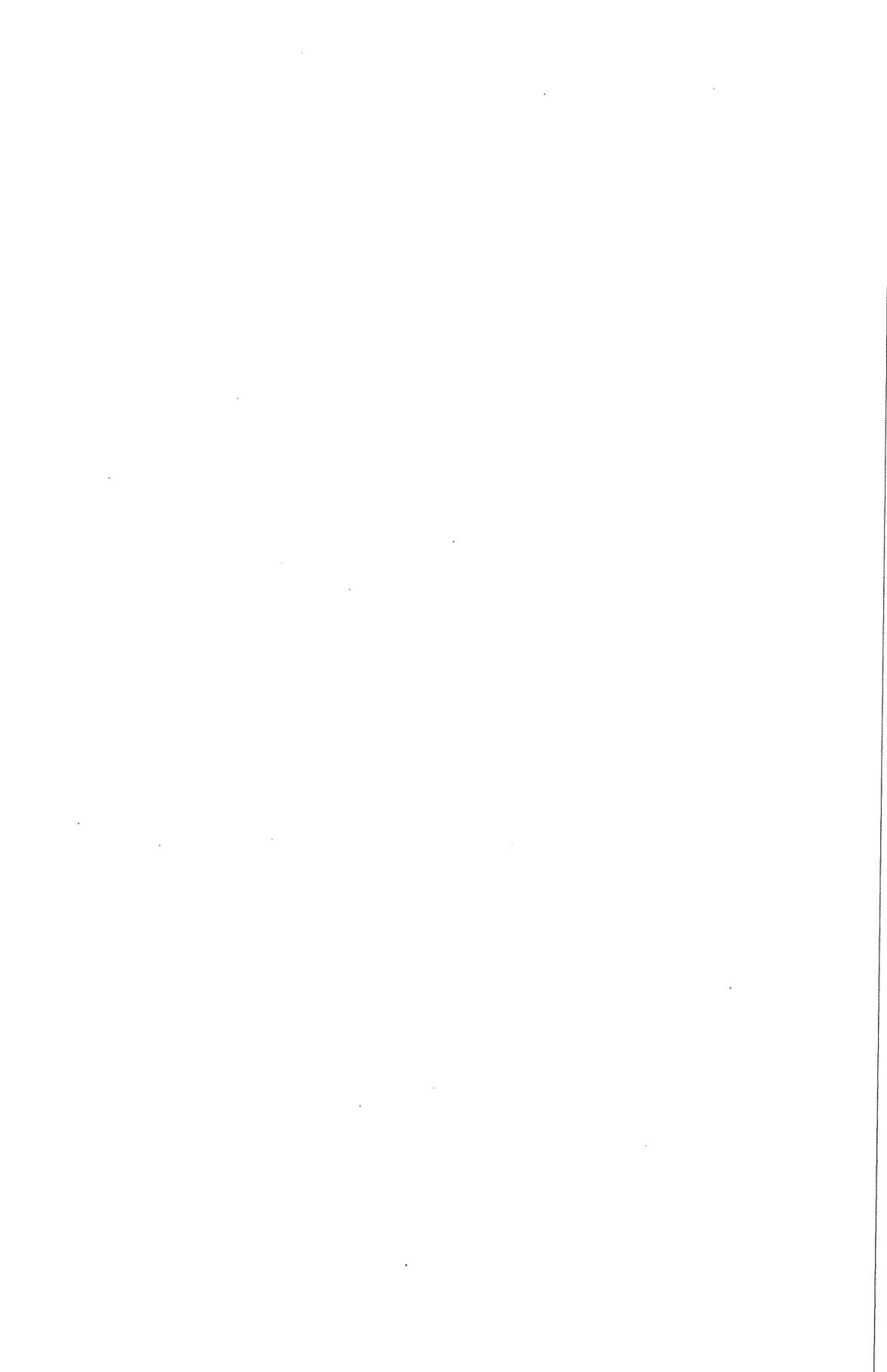
Since we wish to minimize the value of the variable  $z_0$ , then the variables  $z_6$  and  $y_1$  are eligible to be the entering variable. However  $z_6$  can not be a candidate, since  $w_6$  is basic and does not become nonbasic if  $z_6$  is chosen as entering variable. Hence  $y_1$  must be the entering variable. After finding the pivot (entry circled in the tableau) by the usual minimum ratio criterion and performing the corresponding pivot step, we obtain another complementary basic feasible solution given by the following tableau

	$w_1$	$z_1$	$z_2$	$z_3$	$z_4$	$z_5$	$z_6$	$w_2$	
$z_0$	-1	0	0	0	0	0	4	-1	4
$y_1$	0	0	0	0	0	0	-2	1	6
$w_3$	0	0	0	0	0	0	3	-2	9
$w_4$	0	0	0	0	0	0	4	-1	32
$w_5$	0	0	0	0	0	0	0	1	24
$w_6$	0	2	2	1	-2	-2	0	0	1
$w_7$	0	0	0	0	0	0	-1	0	0

In this iteration only  $z_6$  is eligible to be the entering variable. As before this variable can not be a candidate, whence the termination TERM = 3 occurs. However, the LCP(0) has at least the solution  $z = \left(\frac{1}{2}, 0, 0, 0, 0, 1\right)$ ,  $y_1 = 8$ .

**REFERENCES**

- [1] F.A. Al-Khayyal, An implicit enumeration procedure for the general linear complementarity problem, *Mathematical Programming Studies* 31 (1987) 1-20.
- [2] J.F. Bard, An efficient point algorithm for a linear two-stage optimization problem, *Operations Research* 31 (1983) 670-684.
- [3] J.F. Bard, Convex two-level optimization, *Mathematical Programming* 40(1988) 15-28.
- [4] J.F. Bard and J.E. Falk, An explicit solution to the multi-level programming problems, *Computers and Operations Research* 9 (1982) 77-100.
- [5] J.F. Bard and J.J. Moore, A branch-and-bound algorithm for the Bilevel linear programming problem, Technical Report, Department of Mechanical Engineering, University of Texas at Austin, 1988.
- [6] O. Ben-Ayed, C.E. Blair and D. E. Boyce, Solving a real world highway network design problem using bilevel linear programming, BEBR Faculty Working Paper 1463, University of Illinois at Urbana-Champaign, 1988.
- [7] W.F. Bialas and M.H. Karwan, Two-level linear programming, *Management Science* 30(1984) 1004-1020.
- [8] R.G. Bland, New finite pivoting rules for the simplex method, *Mathematics of Operations Research* 2 (1977) 103-107.
- [9] W. Candler and R. Townsley, A linear two-level programming problem, *Computers and Operations Research* 9(1982) 59-76.
- [10] J.J. Júdice and A.M. Faustino, An experimental investigation of enumerative methods for the linear complementarity problem, *Computers and Operations Research* 15(1988) 417-426.
- [11] C.D. Kolstad, A review of the literature on Bilevel Mathematical Programming, Technical Report LA-10284-MS, Los Alamos National Laboratory, 1985.
- [12] C.D. Kolstad and L.S. Lasdon, Derivative evaluation and computational experience with large Bilevel mathematical programs, Faculty Working Paper 1266, College of Commerce and Business Administration, University of Illinois at Urbana-Champaign, 1986.
- [13] H. M. Markowitz, The optimization of a quadratic function subject to linear constraints, *Naval Research Logistics Quarterly* 3 (1956) 111-133.
- [14] K.G. Murty, *Linear and Combinatorial Programming*, Wiley, 1976.
- [15] B. Ramarao and C. M. Shetty, Application of disjunctive programming to the linear complementarity problem, *Naval Research Logistics Quarterly* 31 (1984) 589-600.
- [16] J.K. Reid, A sparsity-exploitation variant of the Bartels-Golub decomposition for linear programming bases, *Mathematical Programming* 24 (1982) 55-63.
- [17] P. Wolfe, The simplex method for quadratic programming, *Econometrica* 27 (1959) 332-338.



## APLICAÇÃO DE UM MODELO DE SIMULAÇÃO À TRANSFERÊNCIA DE UMA BIBLIOTECA

João César das Neves

Faculdade de Ciências Humanas  
Universidade Católica Portuguesa

### Resumo

Neste texto é apresentado um modelo de simulação construído com o objectivo de apoiar a mudança de instalações de uma biblioteca. Esta mudança reveste-se de características particulares devido à necessidade de manter, durante o transporte, a ordem dos livros catalogados.

O modelo de simulação de um sistema de filas de espera foi utilizado em duas fases do processo. Durante a fase de planeamento o modelo serviu para estudar várias alternativas estratégicas para o transporte, determinando o tipo e número de veículos a utilizar. Durante a execução do transporte, o modelo, além de fornecer estimativas do tempo total de mudanças, foi usado para organizar a distribuição das equipas de trabalho.

### Abstract

In this paper is presented a simulation model built to plan the transportation of a library to a new site. The need to keep the sequence of the books, determined by the catalog, from the old to the new place presents some problems for the transport.

The simulation model is a very queuing system, and was used in two stages of the process. In the planning stage it compared several transportation strategies, analysing the number and type of vehicles to be used. During the actual processing, the model helped organising the personnel and provided forecasts for the duration of the process.

**Keywords:** Simulação, Filas de Espera, Planeamento.

### 1- INTRODUÇÃO

A mudança de instalações de uma biblioteca apresenta problemas especiais de alguma complexidade. Estando os livros catalogados e classificados na instalações originais, exige-se do transporte que não altere essa ordem, embora a colocação das estantes seja diferente no novo local. Assim é necessário um controlo apertado de todo o processo, através de uma planificação cuidada.

No caso da Biblioteca Universitária João Paulo II, cuja mudança para as instalações definitivas se realizou em Abril de 1987, foi construído um modelo de simulação de filas de espera para apoiar a planificação dos trabalhos.

No presente texto é apresentado o uso do referido modelo ao longo de todo o processo. Na secção seguinte são esboçadas as características particulares deste problema, realçando a sua dimensão e as especificidades do processo de transporte. No ponto 3 estuda-se a formulação utilizada, apresentando-se o modelo de filas de espera e o programa computacional.

As secções 4 e 5 descrevem a utilização deste instrumento durante as duas fases do trabalho, a planificação e a execução. Os resultados do programa, construído em BASIC num

micro-computador, são usados na escolha dos valores das variáveis do sistema. O trabalho termina com uma breve secção de conclusões.

Este texto desprezencioso não se destina a ensinar qualquer técnica, e a sua simplicidade e pragmatismo tornam trivial a sua ilustração dos modelos de simulação. O objectivo do trabalho é apenas demonstrar como estes modelos podem ser úteis para a obtenção de uma solução rápida, mesmo para problemas complexos.

Pretende o texto ilustrar como um "investigador operacional", com falta de tempo e uma questão complicada para resolver pode, apenas usando meios simples, apresentar resultados em poucas horas de trabalho. Não sendo esta a virtualidade principal dos métodos de simulação é sem dúvida uma das suas características mais preciosas!

## 2- O PROBLEMA

A Biblioteca Geral da Universidade Católica Portuguesa, na sua sede em Lisboa, realizou recentemente a transferência da sua colecção de obras para o edifício da "Biblioteca e Centro Universitário João Paulo II", situado a poucas dezenas de metros das antigas instalações.

Do total de obras disponíveis rondando as duzentas mil espécies, as cerca de setenta mil já catalogadas, classificadas e disponíveis ao público, apresentavam um problema delicado de transporte. Os livros e revistas, que deviam manter a ordem da classificação do catálogo, destinavam-se a estantes de dimensões diferentes das originais, situadas em salas de características distintas.

Assim os livros deveriam ser empacotados no local original, contendo o caixote uma identificação para orientar o seu transporte e, no destino, seriam colocados na estante definitiva, evitando assim qualquer trabalho de rearrumação das espécies. Devido à complexidade do processo foi decidido entregar ao pessoal especializado da biblioteca o encargo de empacotar e desempacotar os livros, na origem e no destino, ficando o transporte a cargo de outras entidades.

Como método geral de identificação dos pacotes de livros foi usado um código que indicativo da prateleira de destino do caixote. Salvo casos especiais, devidamente identificados, a cada prateleira destinava-se um e um só caixote, e assim, através de uma numeração prévia dos conjuntos de estantes e mediante o uso de uma planta das instalações de destino, era possível escrever no caixote qual o andar, bloco, estante e prateleira a que ele se destinava.

Este método, apresentado pelo Professor Doutor P. Jacques Denis, consultor da biblioteca (ver Denis (1987)), apresentava várias vantagens, a principal das quais era que, sendo um sistema indexado e não sequencial, permitia um fácil tratamento de eventuais erros os quais, a ocorrerem, poderiam noutras condições causar grande confusão devido à elevada dimensão do problema.

Outra questão prendia-se com a quantidade de caixotes necessários. Não sendo razoável obter um número suficiente para empacotar toda a biblioteca, o que exigiria muitos milhares, tornou-se imperioso usar um método de transferência circular, onde as equipas de empacotamento na origem e desempacotamento no destino, trabalhavam simultaneamente, enquanto as equipas de transporte levavam os caixotes cheios para as novas instalações e voltavam trazendo-os vazios.

A geografia do local recomendava a divisão do trajecto de cada caixote num máximo de três troços: o trajecto da estante original até à porta exterior do edifício das antigas instalações, o percurso externo entre os dois edifícios, e a distância entre a porta do novo edifício e a estante de destino.

Para efectuar estes trajectos apresentavam-se quatro tipos diferentes de transportes: o transporte manual dos caixotes, o uso de pequenos carros, não motorizados, com capacidade máxima de dez caixotes e, para o percurso exterior, mais duas alternativas pelo uso de uma camioneta (capacidade estimada de 18 caixotes) ou

um camião (50 caixotes).

A curta distância do trajecto exterior, que no total rondava os trezentos metros, permitia encarar a hipótese de usar qualquer dos tipos de transporte sobre rodas, em carros simples ou motorizados. Nos dois troços internos, de dimensão variável devido à diferente colocação das várias estantes, era possível levar os caixotes à mão ou nos carros simples (sem motor). A necessidade de usar o elevador, para os vários andares do edifício novo não levantava problemas, pois os carros podiam entrar nele carregados. O uso do mesmo carrinho para todo o percurso, desde a estante inicial até à final, estava vedado pela existência de uma escada à saída do edifício de origem, o que forçava assim a, pelo menos, um transbordo.

As questões que se levantavam, e para as quais era preciso resposta ligavam-se, portanto, à determinação das seguintes variáveis:

- o tipo de transporte mais adequado a cada trajecto
- o número de caixotes necessários
- a distribuição óptima das equipas de empacotamento, desempacotamento e transporte.
- o tempo necessário para a mudança

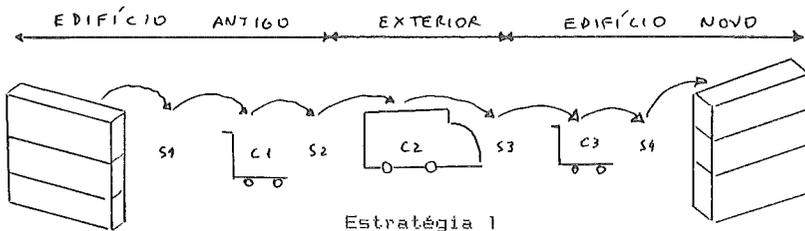
Para resposta a estas perguntas, no período de planeamento, contava-se apenas com estimativas da velocidade de trabalho das equipas (empacotamento/desempacotamento) e da velocidade e capacidade de cada meio de transporte, bem como do tempo de transbordo entre meios de transporte. Estas estimativas, por muito erróneas que fossem, tiveram de ser usadas na determinação da estratégia a adoptar. Durante a execução desta foram obtidos valores mais realistas para os parâmetros, o que permitiu calibrar melhor aquelas decisões que podiam ainda ser revistas.

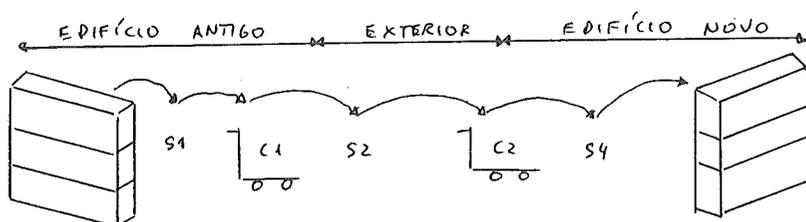
A necessidade de encerrar a biblioteca durante o período de transferência, que causava óbvios transtornos embora a universidade se encontrasse em período de férias, recomendava a minimização do tempo do processo, que assim se transformou em função objectivo do problema. Para a resolver construiu-se o modelo de simulação que se apresenta na secção seguinte.

### 3- O MODELO

O sistema usado para formular o problema foi um modelo de filas de espera múltiplas e encadeadas, com vários servidores e tempos de serviço aleatórios. A figura 1 esboça as duas estratégias possíveis para o transporte, com ou sem o uso de veículos motorizados no transporte exterior.

A primeira estratégia utiliza meios de transporte não motorizados para os percursos internos e veículos motorizados para os trajectos externos. Neste caso existem dois transbordos, um à saída das antigas instalações e outro à entrada das novas. A segunda estratégia consiste em usar apenas meios não motorizados: homens ou carros no transporte interno nas antigas





Estratègia 2

Figura 1

instalações (no fim do qual era forçoso fazer um transbordo, como se disse, devido à existência de uma escada) e carros no transporte externo e interno do novo edifício. Neste caso poupa-se um transbordo (o da entrada do novo edifício). Seria importante escolher entre as duas estratégias, bem como definir em cada uma o valor das variáveis do problema, que são:

- o tempo de empacotamento e desempacotamento dos livros.
- o número de equipas a empacotar e a desempacotar
- o tempo de viagem dos transportes (internos e externo), carregados e descarregados e dos transbordos
- a capacidade dos meios de transporte
- o número de veículos de cada tipo de transporte

O programa desenvolvido usa um método de incremento variável de tempo, saltando o tempo morto entre fenómenos, e simula um período de quatro horas de trabalho, correspondente a meio-dia. Os tempos de empacotamento/desempacotamento e de viagem dos meios de transporte não motorizados são tomados como variáveis aleatórias exponenciais. O tempo de viagem dos veículos motorizados, que inclui os transbordos, é considerado fixo.

Para estudar a solução mais adequada procedeu-se usando um método heurístico, devido à inexistência de método de optimização imediatamente aplicável e ao elevado número de alternativas. É neste ponto que se torna evidente a utilidade do modelo e, ao mesmo tempo o seu aspecto reductionista. A obtenção da solução óptima não é garantida pelo processo, mas a formulação em sistema de optimização, possivelmente impraticável no estado actual das técnicas, era neste caso impossível devido ao reduzido tempo disponível para a obtenção de uma resposta. Assim este método

## CENÁRIOS BASE

## EMPACOTAMENTO/DESEMPACOTAMENTO

Média do tempo de empacotamento/desempacotamento de um caixote.....	3'30''
Número de equipas de empacotamento.....	3
desempacotamento.....	2

## TRANSPORTE INTERNO

	manual	em carros
Média do tempo de viagem carregado.....	5'	3'30''
descarregado.....	2'	2'30''
Capacidade individual.....	1	10
Número na origem.....	3	3
no destino.....	2	2

## TRANSPORTE EXTERNO

	camião	camioneta	em carros(*)
Tempo de viagem.....	30'	15'	10'
Capacidade individual.....	50	18	10
Número.....	1	1	1

(\*) Este é o caso da estratégia 2, Neste caso o número de meios no transporte interno reduz-se para 2 homens ou 1 carro.

expedito é, na verdade o melhor dos possíveis, o "óptimo" para esta situação.

O processo escolhido foi o de definir cenários base (vide quadro 1), com algumas alternativas para os vários parâmetros e, analisando as soluções obtidas, alterar esses cenários, fazendo análises de sensibilidade, até conseguir uma situação que se mostrasse satisfatória perante as restrições concretas a enfrentar. Este método, fortemente interactivo, adaptou-se muito bem à situação em estudo, fornecendo indicações preciosas para os trabalhos de transferência.

#### 4- A FASE DE PLANEAMENTO

Na fase de planeamento pretendia-se escolher a estrutura do processo a adoptar definindo aquelas variáveis que implicavam decisões a tomar com antecedência (tipos de meios de transporte, número de caixotes, etc). Para a fase de execução deixava-se o tratamento daqueles pontos imediatamente ajustáveis (e.g. distribuição das equipas) pois, nessa altura, o efectivo desenrolar dos trabalhos tornaria disponíveis estimativas mais fiáveis de alguns parâmetros fundamentais do sistema.

É de notar a falta de cenários alternativos para os dados relativos ao trabalho das equipas de empacotamento e desempacotamento. Esta falta deve-se fundamentalmente a dois factores. Em primeiro lugar à reduzida informação sobre estes valores, o que fazia com que, dentro do razoável, qualquer estimativa fosse, a priori, tão boa como qualquer outra. Por outro lado, até a consideração de poucas alternativas multiplicaria de tal modo o número de ensaios que tornaria mais difícil análise das outras variáveis. No entanto, durante a fase de planeamento foram feitas algumas análises de sensibilidade que não são apresentadas para não estender o artigo, visto não alterarem as decisões finais.

No quadro 1 estão apresentados os valores tomados como hipótese para as constantes do sistema e das variáveis de decisão que constituem os cenários base. Para estes cenários, os resultados encontram-se no quadro 2. Os parâmetros recolhidos da solução, entre os vários que o programa gera, foram:

- [Cx.dest] e [Cx.tran] - o número de caixotes colocados no destino, e de caixotes em trânsito (nos meios de transporte  $C_i$  ou nas filas de espera  $S_i$ ) ao fim das quatro horas de trabalho.

- [T.trans] - o número de dias necessários para transferir a totalidade da Biblioteca (cerca de 2800 caixotes); os limites do intervalo apresentado, e calculado a partir dos dados anteriores, provêm da suposição de que os caixotes em trânsito estarão colocados na estante de destino, quando se retoma o trabalho (para novo serviço de quatro horas) ou voltaram para a origem.

- [P.T.fim] - a percentagem do tempo total de trabalho na qual não houve caixotes para desempacotar no local de destino, ou seja em que as equipas de desempacotamento estiveram desocupadas.

- [Max.cai] - o número máximo de caixotes que se encontraram simultaneamente cheios de livros no período das quatro horas.

- [ $S_i$  (max), (med), (d.p)] - O montante de livros nos stocks das filas de espera intermédias ( $S_i$ , com  $i=1,2,3,4$  na estratégia 1 e  $i=1,2,4$  na estratégia 2, ver figura 1). São indicados os stocks máximos e médios, bem como o seu desvio padrão, com o fim de situar os pontos fracos do sistema.

Dos valores obtidos ressalta a inferioridade no transporte interno da solução manual face ao uso de carros. No transporte externo o resultado é menos claro, embora a estratégia 2, pelo não uso de carros motorizados que evita um transbordo, apareça

## RESULTADOS DOS CENARIOS BASE

Tr.Int.	Carros			Homens		
Tr.Ext.	Camião	Camioneta	Carros	Camião	Camioneta	Carros
Cx.dest	118	114	162	41	80	60
Cx.tran	155	172	120	222	201	214
T.trans	5.1-11.9	4.9-12.3	5.0-8.7	5.3-34.4	5.0-17.6	5.1-23.5
P.T.fim	35%	38%	14%	81%	55%	70%
Max.cai	155	172	121	222	202	218
max	9	9	25	149	178	212
S1 med	4.38	4.48	8.30	67.82	80.05	110.93
d.p	2.96	2.92	5.95	39.66	51.09	61.38
max	100	134	30	49	17	9
S2 med	38.04	63.07	3.33	20.62	8.01	4.10
d.p	24.66	40.67	6.71	14.57	5.19	2.80
max	30	8	-	60	19	-
S3 med	2.38	4.03	-	13.18	6.06	-
d.p	7.31	3.16	-	16.51	5.64	-
max	51	23	105	3	7	10
S4 med	17.48	5.26	47.85	.22	.80	1.49
d.p	15.76	5.60	32.30	.47	1.20	2.75

Quadro 2

como a mais favorável, passando o ponto crítico do sistema de S2 para S4.

Dados estes resultados, eliminou-se do espectro das soluções analisáveis as que usavam homens nos transportes internos, decisão abusiva mas simplificadora. Para clarificar os resultados dos veículos motorizados foi feita uma análise de sensibilidade aos seus parâmetros, apresentada nos quadros 3 e 4.

## ANÁLISE DE SENSIBILIDADE AO TRANSPORTE MOTORIZADO

Tr.Ext.	Camião rápido (20')	Camioneta rápida (6')	Camioneta pequena (12 cai.)	Camioneta grande (12c, 12')	2 Camionetas
Cx.dest	129	140	80	153	140
Cx.tran	146	143	194	121	139
T.trans	5.1-10.9	4.9-10.0	5.1-17.6	5.1-9.2	5.0-10.0
P.T.fim	30%	16%	57%	15%	21%
Max.cai	146	144	194	122	139
max	9	10	10	9	9
S1 med	4.50	4.52	4.36	4.34	4.90
d.p	3.00	3.09	3.08	2.93	2.84
max	50	38	176	90	28
S2 med	19.91	10.83	86.46	40.85	11.30
d.p	14.17	7.87	53.85	23.50	6.56
max	30	8	8	0	10
S3 med	3.93	3.94	3.12	0.00	4.02
d.p	9.04	3.05	2.72	0.00	3.13
max	87	132	18	36	85
S4 med	34.11	51.55	3.02	17.11	35.06
d.p	28.91	38.77	4.30	10.40	29.03

Quadro 3

A alteração das características dos transportes (quadro 3) não melhoram significativamente os resultados destes transportes, mantendo-se inferiores à solução da estratégia 2 (terceira coluna do quadro 2).

A manipulação das equipas de empacotamento e desempacotamento (quadro 4) também não traz resultados muito superiores. Procurou-se aqui, além de analisar o efeito de mais uma equipa de desempacotamento, testar os resultados de, com as cinco equipas originais (3 empac./2 desemp.), transferir ao fim de duas horas uma equipa da origem para o destino, trabalhando-se na segunda parte das quatro horas com duas equipas a empacotar e três a

CENARIOS ALTERNATIVOS PARA AS EQUIPAS

Equipas	5 equipas c/tranf.....			6 equipas.....		
Tr.Ext.	Camião	Camionet.	2 Camion	Camião	Camionet.	2 Camion
Cx.dest	149	120	190	143	120	208
Cx.tran	75	108	45	136	156	67
T.trans	6.3-9.4	6.1-11.7	6.0-7.4	5.0-9.8	5.1-11.7	5.1-6.7
P.T.fim	44%	48%	25%	51%	59%	29%
Max.cai	103	117	87	146	156	79
max	9	10	9	10	9	9
S1 med	4.33	4.28	4.61	4.19	4.43	4.54
d.p	2.88	2.98	2.88	2.99	2.96	2.86
max	70	94	28	110	134	34
S2 med	30.45	51.81	9.97	43.20	61.06	12.18
d.p	18.21	27.94	6.45	27.18	38.65	8.10
max	30	8	10	30	8	10
S3 med	1.93	4.04	3.94	3.38	4.01	3.80
d.p	5.82	3.16	3.26	8.73	3.15	3.14
max	41	23	47	42	20	31
S4 med	11.33	4.74	17.44	10.04	3.62	10.19
d.p	12.61	5.97	13.94	12.34	5.58	8.53

Quadro 4

desempacotar. Neste caso, com a equipa passou também um carro de transporte interno do edificio origem para o de destino.

Os resultados desta manipulação são também inferiores aos da solução da estratégia 2, excepto no caso de duas camionetas (embora este apresente maiores percentagens de tempo perdido no desempacotamento e maiores limites inferiores para o tempo de mudança). Assim a escolha recaiu sobre a política indicada na terceira coluna do quadro 2, pois os resultados dos veículos motorizados não pareciam compensar o seu custo mais elevado.

O número exacto de carros a adquirir deveria ser determinado pelo programa. Para isso consideraram-se três variáveis de política: o número de equipas de empacotamento/desempacotamento (3/2 ou 3/3); a existência ou não de transferência, a meio do trabalho, de uma equipa da origem para o destino e o número e distribuição dos carros pelos dois trajectos [interno no edificio origem]/[externo e interno no edificio destino] ([1]/[2], [1]/[3], [2]/[3] ou [1]/[4]). Os resultados apresentam-se no quadro 5.

Várias conclusões interessantes se podem tirar desta análise. Em primeiro lugar a transferência da equipa, quem se mostrara sempre vantajosa no quadro 4, só melhora os resultados quando o

CENARIOS ALTERNATIVOS PARA AS EQUIPAS

Equipas	3/2	3/2	3/2	3/3	3/3	3/3	3/3	3/3	3/3
Tranf.?	n	s	s	n	s	n	s	n	s
Carros	1/2	1/2	1/3	1/3	1/3	2/3	2/3	1/4	1/4
Cx.dest	167	200	195	218	210	242	210	232	230
Cx.tran	55	27	53	57	8	31	10	73	13
T.trans	6-8	7-7	6-7	5-6	6-7	5-6	6-7	5-6	6-6
P.T.fim	14%	22%	14%	20%	38%	21%	38%	17%	30%
Max.cai	123	70	93	66	48	46	46	75	61
max	26	26	25	17	16	9	9	22	17
S1 med	8.68	7.30	7.80	5.27	4.57	4.43	4.19	6.01	4.70
d.p	5.42	5.14	5.67	3.69	3.38	2.98	2.95	4.35	3.55
max	80	20	30	30	30	20	20	30	30
S2 med	29.27	5.51	3.49	1.76	1.75	1.37	.86	3.91	3.96
d.p	27.03	6.47	6.75	5.35	5.34	3.66	3.07	7.05	7.15
max	43	42	69	59	31	32	31	63	30
S4 med	16.67	15.68	36.05	19.85	6.72	10.13	5.78	18.11	7.21
d.p	9.78	11.77	21.27	16.05	7.97	8.81	6.81	16.57	7.01

Quadro 5

número de carros é pequeno (total de 3); nos outros casos, o aumento da ociosidade das equipas de desempacotamento é suficientemente grande para impedir a melhoria. O aumento do número de carros faz passar o ponto crítico de S2 para S4, além de reduzir significativamente o número de caixotes necessários (de mais de 120 para cerca de 50).

O diminuto ganho adicional da aquisição do quinto carro, reduzindo-se o tempo previsto para a mudança de menos de um dia, torna discutível a compensação do seu custo. Por outro lado a colocação ideal deste quinto carro parece ser no edifício original, o que reduzia praticamente todas as filas de espera. No entanto, como se verá, as diferenças que suportam estas conclusões são já suficientemente pequenas para que aquelas se tornem muito sensível ao parâmetro que não se alterou até agora, o tempo de serviço das equipas.

O número de equipas especializadas para o processamento dos livros (que se tomava variável devido a estas serem convertíveis em equipas de controle geral do processo e vice-versa) deveria ser de seis, pois os ganhos adicionais eram significativos (mais de um dia).

A decisão que resultou de todo este processo de planeamento foi a de usar carros em todo o processo de transporte, colocando pelo menos um no primeiro trajecto e três no segundo trajecto. Foram reservados quatro carros para a tarefa, mantendo-se a possibilidade de usar mais um carro, no caso de ser vantajoso. Assim a tomada de decisão final quanto a algumas variáveis (número e disposição dos carros) pode ser adiada até à obtenção de dados mais realistas para os parâmetros. O número de equipas destinadas ao empacotamento/desempacotamento dos caixotes foi de seis e adquiriu-se uma centena de caixotes.

##### 5- A FASE DE EXECUÇÃO

Nesta fase as únicas hipóteses que foi preciso alterar referiram-se exactamente ao tempo de serviço das equipas

##### RESULTADOS DA FASE DE EXECUÇÃO

Trans.?	.....c/transferência.....			.....s/transferência.....		
Carros	1/3	2/3	1/4	1/3	2/3	1/4
Cx.dest	260	340	350	350	330	410
Cx.tran	86	16	35	121	24	44
T.trans	4.0-5.4	3.9-4.1	3.6-4.0	3.0-4.0	3.9-4.2	3.1-3.4
P.T.fim	66%	55%	54%	50%	52%	41%
Max.cai	171	16	51	129	158	54
max	100	12	31	111	15	32
S1 med	50.29	4.49	10.58	54.06	4.78	9.74
d.p	33.39	2.89	6.65	37.04	2.95	5.70
max	80	100	20	50	140	20
S2 med	29.00	36.79	.93	13.21	64.56	1.87
d.p	16.97	28.57	2.92	13.52	46.49	4.13
max	17	24	27	26	19	32
S4 med	2.20	3.34	3.59	3.68	3.57	5.56
d.p	3.68	4.75	5.14	4.87	4.59	6.75

##### Quadro 6

especializadas. Observou-se, no período inicial da mudança, que estas eram muito mais rápidas do que se previra, rondando o tempo de empacotamento (que incluía a etiquetagem com o endereçamento) o minuto e meio, e o de desempacotamento aproximadamente um minuto.

Perante este dados procurou-se de novo utilizar o programa par definir o perfil final do processo. Os resultados são apresentados no quadro 6. As alternativas que se apresentavam agora eram, além da definição da distribuição dos carros, a resolução da questão da transferência da equipa a meio do período

que, devido aos diferentes resultados que apresentara nas análises anteriores, ainda motivava algumas dúvidas.

O resultado foi claro. A solução que se apresentava mais vantajosa (última coluna do quadro) não incluía transferência da equipa, e o quinto carro desta vez colocado no segundo percurso tinha um contributo significativo. O período total a dedicar à mudança seria de cerca de quatro dias, o que era mais do que satisfatório, embora fossem de prever alguns atrasos, sempre inevitáveis.

## 6- CONCLUSÕES

Deste exercício pode retirar-se fundamentalmente a extrema utilidade dos métodos expeditos e interactivos na resolução deste tipo de problemas. A preciosa ajuda conseguida na determinação das variáveis de decisão, apesar das imprecisões metodológicas conscientemente cometidas, mais do que justificaram o reduzido custo (cerca de dois dias de trabalho) da construção do modelo.

As recomendações do programa foram seguidas à risca, e se o tempo final de execução foi cerca do dobro previsto na melhor solução, a folga programada foi mais do que suficiente para não causar perturbações.

Mesmo este facto \_ que se ficou a dever sobretudo a dificuldades de acesso a algumas estantes do edifício antigo, na parte final do processo, e ao tratamento especial a dar aos periódicos \_ exemplifica correctamente a proposta metodológica do método. A consideração das suas causas, que implicaria alteração das taxas de serviço ao longo do processo, era suficientemente complexa para que a forma expedita utilizada fosse mais recomendável, desde que tomadas as devidas precauções, em termos de folgas.

Assim a grande conclusão a tirar deste exemplo é que, na resolução de problemas concretos, se deve sempre usar de uma avaliação "económica" do esforço analítico em relação ao proveito que dele se tira e que, por outro lado, a técnica da simulação é dos métodos com maior "valor acrescentado marginal", sobretudo em casos em que o tempo disponível para a análise é curto.

## BIBLIOGRAFIA

- Denis, J (1986) "Rapport de mission", mimeo, Universidade Católica Portuguesa, Outubro  
Direcção da Biblioteca da UCP (1987) "Relatório de preparação da B.U.J.P.II", mimeo, Universidade Católica Portuguesa, Março  
Hillier & Lieberman (1980) "Introduction to Operation Research", Holden Day Inc.



# VALIDADE DE UMA SOLUÇÃO APROXIMADA PARA O PERÍODO DE INSPECÇÃO DE SISTEMAS COM DIFERENTES TIPOS DE TAXAS DE RISCO

J. Rodrigues Dias

Dept. de Matemática, Universidade de Évora

## Resumo

Considere-se um sistema que, após uma falha, não evidencia quaisquer sintomas de mau funcionamento pelo que, para se conhecer o seu estado, precisa de ser inspeccionado. Assim, admita-se que há um custo fixo por cada inspecção e um custo fixo por cada unidade de tempo de mau funcionamento não detectado.

Num artigo anterior apresentou-se uma solução aproximada para o período de inspecção, que generalizava uma aproximação anterior, mostrando-se que no caso em que a taxa de risco do sistema é constante (distribuição exponencial) tal solução se pode considerar como (quase) óptima.

Neste artigo, analisa-se a validade da aproximação então obtida no caso em que a taxa de risco assume uma das formas seguintes: a) decrescente; b) crescente; c) inicialmente crescente e depois decrescente; d) e, finalmente, em forma de banheira.

## Abstract

Consider a system in which the bad working state is only known if it is inspected. Suppose that each inspection and each unit time of system bad working have fixed costs. In a previous paper, for the constant failure rate (exponential distribution), we have presented a nearly optimal approximate solution for the inspection period, which is a generalization of a previous one.

In this paper, we analyse the validity of that new approximation when the failure rate is: decreasing, increasing, increasing and after decreasing and, finally, when the failure rate has a "bathtub" shape.

## 1. INTRODUÇÃO

Considere-se um sistema cujo tempo de vida (intervalo de tempo entre o instante  $t=0$  em que o sistema em estado de novo começa a funcionar e o instante em que falha) é uma variável aleatória contínua  $T$ , com função de fiabilidade  $R(t)$  e taxa de risco  $h(t)$  dadas por:

$$R(t) = 1 - F(t), \quad t > 0 \quad (1)$$

$$h(t) = f(t) / R(t), \quad t > 0 \quad (2)$$

sendo  $f(t)$  e  $F(t)$  a função densidade de probabilidade e a função

de distribuição, respectivamente.

Considere-se ainda que o sistema, tendo já falhado, não evidencia quaisquer sintomas de mau funcionamento, pelo que a falha é apenas detectável por inspecções.

Sendo  $C_1$  o custo de cada inspecção e  $C_2$  o custo por unidade de tempo de mau funcionamento não detectado, então uma questão importante que se coloca é determinar a sequência de instantes  $t_k$ ,  $k=1, 2, 3, \dots$ , com  $t_0=0$ , em que o sistema deve ser inspeccionado de tal modo que o custo total médio por ciclo de funcionamento seja mínimo.

Considera-se que um ciclo começa com o sistema em estado de novo e acaba no momento em que a falha é detectada.

Este problema pode encontrar aplicações em áreas tão importantes e diversas como o controlo de qualidade, a medicina, questões de defesa, etc.

Barlow et al (1965) mostraram que no caso em que a distribuição de  $T$  é exponencial e no caso em que a probabilidade de detectar a falha na inspecção que se segue ao momento da sua ocorrência é 1, então a política que minimiza o custo total médio por ciclo consiste em inspeccionar o sistema periodicamente nos instantes  $t_k = kP$ , com  $k = 1, 2, 3, \dots$ , em que  $P$  é a solução da equação

$$\exp(LP) - (1 + Lr + LP) = 0 \quad (3)$$

sendo  $L$  o parâmetro da distribuição exponencial ( com  $R(t) = \exp(-Lt)$  ) e  $r = C_1/C_2$ .

Com o objectivo de na prática se obter facilmente uma solução aproximada do período de inspecção  $P$ , em especial quando a distribuição de  $T$  não é exponencial (distribuição de Weibull, por exemplo, caso em que a equação correspondente a (3) se complica), Nakagawa e Yasui (1979) obtiveram o valor aproximado

$$P^* = [2 r E(T)]^{1/2} \quad (4)$$

Este resultado assenta numa aproximação considerada por

Schneeweiss (1976). Rodrigues Dias (1983) refere que o resultado anterior se pode obter facilmente a partir de (3) usando a aproximação

$$\exp(LP) \approx 1 + LP + (LP)^2/2 \quad (5)$$

## 2. A SOLUÇÃO

Tendo em conta que no caso da distribuição exponencial se tem  $E(T) = 1/L$ , a equação (3) pode-se escrever (Rodrigues Dias (1987a)):

$$\exp(P') - (1 + r' + P') = 0 \quad (6)$$

em que  $P' = P/E(T)$  e  $r' = r/E(T)$ .

No Quadro 1 estão indicados, para diferentes valores de  $r$ , os valores dos cocientes seguintes:

$$Q1 = (P^* - P) / P \cdot 100 \% \quad (7)$$

$$Q2 = [E(C)^* - E(C)] / E(C) \cdot 100 \% \quad (8)$$

em que, recorde-se,  $P^*$  e  $P$  são, respectivamente, os valores dos períodos de inspecção obtidos a partir de (4) e (3), e em que  $E(C)^*$  e  $E(C)$  são os correspondentes custos totais médios por ciclo, com  $E(C)$ , no caso de distribuição exponencial, dado por

$$E(C) = C_1 / [1 - \exp(-LP)] + C_2\{P/[1 - \exp(-LP)] - 1/L\} \quad (9)$$

O aspecto interessante, e curioso, que importa analisar no Quadro 1, e que tinha passado despercebido num artigo anterior (Rodrigues Dias (1983)), é o seguinte: os valores de  $Q1$  ajustam-se muito bem por uma parábola.

Na realidade, usando para efeitos de predição uma curva de regressão do tipo

$$Q1^* = a + b r'^{1/2} \quad (10)$$

ou seja, a recta de regressão

$$Q1^* = a + bX \quad (11)$$

com  $X = r'^{1/2}$ , obtiveram-se a partir dos valores do Quadro 1 os seguintes resultados, recorrendo ao conhecido método dos mínimos quadrados:

r	Q1	Q2	Q1**	Q2**
.001	.75	<.01	<.01	<10 <sup>-6</sup>
.002	1.05	<.01	<.01	<10 <sup>-6</sup>
.003	1.29	<.01	<.01	<10 <sup>-6</sup>
.004	1.49	.01	.01	<10 <sup>-6</sup>
.005	1.67	.01	.01	<10 <sup>-6</sup>
.006	1.83	.02	.01	<10 <sup>-6</sup>
.007	1.97	.02	.01	<10 <sup>-6</sup>
.008	2.11	.02	.01	<10 <sup>-5</sup>
.009	2.24	.02	.02	<10 <sup>-5</sup>
.01	2.36	.03	.02	<10 <sup>-5</sup>
.02	3.33	.05	.02	<10 <sup>-5</sup>
.03	4.08	.08	.03	<10 <sup>-5</sup>
.04	4.71	.11	.03	<10 <sup>-5</sup>
.05	5.27	.13	.03	<10 <sup>-5</sup>
.06	5.77	.16	.04	<10 <sup>-5</sup>
.07	6.23	.18	.04	<10 <sup>-5</sup>
.08	6.66	.21	.04	<10 <sup>-5</sup>
.09	7.06	.23	.04	<10 <sup>-5</sup>
.1	7.45	.26	.04	<10 <sup>-5</sup>
.2	10.52	.49	.05	<10 <sup>-4</sup>
.3	12.87	.71	.05	<10 <sup>-4</sup>
.4	14.85	.92	.05	<10 <sup>-4</sup>
.5	16.59	1.13	.04	<10 <sup>-5</sup>
.6	18.16	1.32	.03	<10 <sup>-5</sup>
.7	19.60	1.51	.02	<10 <sup>-5</sup>
.8	20.94	1.69	.01	<10 <sup>-6</sup>
.9	22.20	1.86	>-.001	<10 <sup>-8</sup>
1.0	23.38	2.03	-.01	<10 <sup>-6</sup>

QUADRO 1

a = .000255 , b = .2340

coeficiente de determinação = .999994

Como é obvio, pode considerar-se excelente o ajustamento referido. Tendo em conta o valor reduzido tomado por a, podemos desprezá-lo na prática e assim escrever

$$Q1 = .234 r'^{1/2}$$

(12)

Tendo agora presentes as relações (4), (7) e (12), pode obter-se finalmente a solução aproximada para o período de inspecção

$$P^{**} = [2 r E(T)]^{1/2} / [1 + .234 r']^{1/2} \quad (13)$$

### 3. VALIDADE DA SOLUÇÃO APROXIMADA NO CASO EM QUE A TAXA DE RISCO É CONSTANTE

Obtida a aproximação anterior, é agora importante analisar o seu grau de precisão no que diz respeito, neste caso, à distribuição exponencial (Rodrigues Dias (1987b)). Assim, ainda no Quadro 1, representam-se os valores dos cocientes

$$Q1^{**} = (P^{**} - P) / P \cdot 100 \% \quad (14)$$

$$Q2^{**} = [E(C)^{**} - E(C)] / E(C) \cdot 100 \% \quad (15)$$

para os mesmos valores de  $r'$ , em que  $E(C)^{**}$  é agora o custo total médio por ciclo correspondente ao período de inspecção  $P^{**}$  estimado a partir de (13).

Da análise dos valores do Quadro 1 podemos tirar as seguintes conclusões, apesar de serem evidentes:

a) Os erros relativos dos períodos de inspecção obtidos a partir da aproximação (13) são extremamente reduzidos, nunca superiores a .05% (e quase sempre bastante inferiores) para o amplo conjunto de valores de  $r'$  apresentado. Para efeitos práticos, pode considerar-se que os referidos erros relativos não têm qualquer significado.

b) Comparando os erros relativos dos períodos de inspecção associados à aproximação (4) de Nakagawa e Yasui (1979) com os correspondentes erros associados à aproximação (13) por nós obtida, é evidente que estes são incomparavelmente menores, em especial quando  $r'$  é elevado. Este facto percebe-se facilmente já

que é para valores elevados de  $r'$  que ganha significado o factor de correcção  $1/(1+.234 r'^{\frac{1}{2}})$  da expressão (13).

c) Os erros relativos dos custos associados à aproximação (13) são de tal modo reduzidos que na prática se podem considerar como inexistentes, assim se reforçando a validade da dita aproximação no caso da distribuição exponencial.

#### 4. VALIDADE DA SOLUÇÃO APROXIMADA NO CASO EM QUE A TAXA DE RISCO APRESENTA DIFERENTES TIPOS

Depois de termos concluído que a aproximação (13), que generaliza a aproximação (4), se pode considerar em termos práticos como óptima no caso da distribuição exponencial, vejamos agora até que ponto é que ela funciona bem no caso de outras distribuições com diferentes tipos de taxas de risco.

Para efeitos de análise e comparação, vamos aqui considerar as distribuições de Weibull, de Hjorth (1980) e a log-normal. A escolha destas distribuições encontra justificação nos seguintes factos:

a) A distribuição de Weibull, com função densidade de probabilidade  $f(t)$  dada por

$$f(t) = B(t/A)^{B-1} \exp[-(t/A)^B] / A, \quad t > 0 \quad (18)$$

em que  $A > 0$  é um parâmetro de escala e  $B > 0$  é um parâmetro de forma, é uma distribuição tal que para  $B < 1$  a taxa de risco é decrescente, para  $B = 1$  é constante (caso da exponencial) e para  $B > 1$  a taxa de risco é crescente;

b) A distribuição de Hjorth (1980), com função densidade de probabilidade  $f(t)$  dada por

$$f(t) = \exp(-\delta t / 2) [(1 + \beta t)\delta t + \theta] / (1 + \beta t)^{\theta/\beta + 1}, \quad t > 0 \quad (17)$$

em que  $\beta$ ,  $\theta$  e  $\delta$  são parâmetros positivos, é uma distribuição cuja taxa de risco pode assumir diferentes formas consoante os valores dos respectivos parâmetros. O caso que aqui mais nos interessa é aquele em que a taxa de risco é do tipo "bathtub", considerando para isso  $\beta = \theta = 1$  e  $\delta = .01$ .

c) A distribuição log-normal, com função densidade  $f(t)$  dada por

$$f(t) = \exp[-(\log t - \mu)^2 / (2\sigma^2)] / [(2\pi)^{1/2} \sigma t], \quad t > 0 \quad (18)$$

em que  $\mu$  e  $\sigma$  são respectivamente a média e o desvio padrão da variável normal  $Y = \log T$ , é uma distribuição em que a taxa de risco começa por crescer para depois decrescer;

Para obter mais pormenores pode ver-se Rodrigues Dias (1987a).

Depois de justificada a escolha das distribuições anteriores (que vão servir de suporte ao nosso estudo), vamos começar por apresentar no Quadro 2, relativo aos valores  $r' = .0125$  e  $.05$ , e no Quadro 3, relativo aos valores  $r' = .2$  e  $.8$ , os valores das grandezas  $Q1$  e  $Q2$  definidas em (7) e (8), referentes agora às distribuições assinaladas.

Para uma análise destes valores e, em particular, para uma sua análise comparativa, pode ver-se Rodrigues Dias (1983) e (1987a).

De uma forma análoga, apresentamos ainda nos Quadros 2 e 3 os valores das grandezas  $Q1$  e  $Q2$ , relativos à aproximação (13) de  $P$  por nós obtida. Mantêm-se as mesmas distribuições e os mesmos valores de  $r'$ .

Comparando nos Quadros 2 e 3 os valores correspondentes de  $Q1$  e de  $Q1$ , por um lado, e, por outro, os valores de  $Q2$  e de  $Q2$ , podemos pôr em evidência as conclusões seguintes:

a) Para a distribuição de Weibull com  $B = .7$  e  $B = 1$  e para as distribuições de Hjorth (1980) e log-normal (esta de um modo geral) verifica-se que a aproximação (13) obtida para o período

		$r' = r/E(T)$							
		.0125				.0500			
Distribuição		-Q1	** Q1	Q2	** Q2	Q1	** Q1	Q2	** Q2
	0.7	7.3	4.5	0.3	0.1	11.3	5.8	0.6	0.2
	1.0	2.6	0.0	0.0	0.0	5.3	0.0	0.1	0.0
	1.5	0.4	- 2.2	0.0	0.0	1.1	- 3.9	0.0	0.1
Weibull	2.0	0.0	- 2.5	0.0	0.0	0.0	- 5.0	0.0	0.1
B=	2.5	0.0	- 2.6	0.0	0.0	- 0.1	- 5.1	0.0	0.1
	3.0	0.0	- 2.6	0.0	0.0	- 0.1	- 5.1	0.0	0.1
	4.0	0.0	- 2.5	0.0	0.0	0.0	- 5.0	0.0	0.1
	5.0	0.0	- 2.5	0.0	0.0	- 0.4	- 5.4	0.0	0.1
	7.0	0.0	- 2.5	0.0	0.0	3.8	- 1.3	0.3	0.0
Hjorth		6.0	3.3	0.2	0.1	11.1	5.6	0.6	0.2
Log-normal		-0.4	- 2.9	0.0	0.0	1.3	- 3.8	0.0	0.1

QUADRO 2

de inspecção é melhor que a aproximação obtida por Nakagawa e Yasui (1979), pois os erros relativos  $Q1^{**}$  e  $Q2^{**}$  são menores que os seus correspondentes  $Q1$  e  $Q2$  ;

b) Para os outros valores considerados de  $B$  da distribuição de Weibull, em particular para  $B = 2$ , verifica-se que a aproximação  $P^*$  funciona melhor que  $P^{**}$  ;

c) Em termos dos diferentes tipos de taxas de risco que considerámos, e tendo em conta as conclusões a) e b), poder-se-á dizer que a aproximação que obtivemos funciona melhor que aquela de Nakagawa e Yasui (1979).

Repare-se que, em certa medida, os resultados obtidos eram previsíveis na medida em que o valor de  $P^{**}$ , aproximando-se ou afastando-se, conforme os casos, do valor correcto do período de inspecção  $P$ , conduziria inevitavelmente a soluções melhores ou

		$r' = r/E(T)$							
		.2000				.8000			
Distribuição		Q1	Q1 **	Q2	Q2 **	Q1	Q1 **	Q2	Q2 **
	0.7	17.5	6.4	1.3	0.2	27.0	5.1	2.5	0.1
	1.0	10.5	0.1	0.5	0.0	20.9	0.0	1.7	0.0
	1.5	3.4	- 6.4	0.1	0.2	11.8	- 7.5	0.6	0.3
Weibull	2.0	0.0	- 9.5	0.0	0.4	1.3	-16.2	0.0	1.3
B=	2.5	- 1.0	-10.4	0.0	0.5	-11.2	-26.5	1.0	4.7
	3.0	- 1.0	-10.4	0.0	0.5	-13.7	-28.6	3.1	10.6
	4.0	-53.0	-54.5	1.8	2.9	-11.4	-26.7	5.1	22.5
	5.0	-52.3	-56.8	12.9	15.4	- 7.9	-23.8	4.1	31.5
	7.0	-50.0	-54.7	29.6	31.9	- 1.9	-18.9	0.4	42.8
Hjorth		19.0	7.7	1.5	0.3	30.0	7.5	3.0	0.2
Log-normal		9.0	- 1.3	0.4	0.0	24.8	3.2	2.5	0.1

QUADRO 3

piores em termos de custos.

5. CONSIDERAÇÕES FINAIS

Em jeito de síntese final, poder-se-á acentuar que:

a) A aproximação (13) por nós obtida para o período de inspeção generaliza a aproximação (4) obtida por Nakagawa e Yasui (1979), ganhando especial significado para maiores valores de  $r'$ ;

b) A referida aproximação (13) pode considerar-se (quase) ótima no caso em que a taxa de risco do sistema é constante (distribuição exponencial), sendo incomparavelmente melhor que a aproximação (4). Repare-se que esta situação merece um especial

interesse prático, pela simplicidade e aplicabilidade da referida distribuição exponencial.

c) Para outras distribuições com taxas de risco de outro tipo, verifica-se que a aproximação (13) é na maioria dos casos estudados melhor que a aproximação (4). Em particular, revela-se melhor nos casos de sistemas com taxas de risco decrescente, de tipo "bathtub", e, de um modo geral, quando a taxa de risco é inicialmente crescente e depois decrescente.

#### REFERÊNCIAS:

- Barlow, R. E. ; Proschan, F.; Hunter, L. C. (1965) - Mathematical Theory of Reliability, Wiley, New-York
- Hjorth, U. (1980) - A Reliability Distribution With Increasing, Decreasing, Constant and Bathtub-Shaped Failures Rates, Technometrics, Vol 22, Nº 1, pp 99-107
- Nakagawa, T; Yasui, K. (1979) - Approximate Calculation of Inspection Policy with Weibull Failure Times, IEEE Trans. on Reliability, Vol R-28, pp 403-404
- Rodrigues Dias, J. (1983) - Influence de la Période d'Inspection sur les Coûts dans l' Inspection Périodique de Systèmes, Revue de Statistique Appliquée, Vol XXXI, Nº4, pp 5-15
- Rodrigues Dias, J. (1987a) - Políticas de Inspeção de Sistemas , Tese de Doutoramento, Universidade de Évora
- Rodrigues Dias, J. (1987b) - Uma Solução para o Período de Inspeção de Sistemas, XII Jornadas Luso-Espanholas de Matemática, Universidade do Minho
- Schneeweiss, W. G. (1976) - On the Mean Duration of Hidden Faults in Periodically Checked Systems, IEEE Trans. on Reliability, Vol R-25, pp 346-348

## INTRODUÇÃO GERAL ÀS ABORDAGENS MULTICRITÉRIO DE APOIO À TOMADA DE DECISÃO

Carlos Bana e Costa

I.S.T./CESUR  
Av. Rovisco Pais 1000 Lisboa

### Resumo

O presente artigo é uma versão revista da primeira parte de uma comunicação do autor ao 3º Congresso da Associação Portuguesa de Investigação Operacional. O seu objectivo é tão simplesmente proporcionar uma primeira introdução geral às abordagens multicritério de apoio à decisão, tendo-se procurado uma orientação bibliográfica no sentido de referenciar as obras fundamentais a um estudo complementar da temática em causa.

### Abstract

This paper presents a general introduction to the "Multiple Criteria Decision Aid Methodology" and it is a revised version of the communication presented by the author to the Third Congress of the Portuguese Operational Research Society.

**Keywords:** Multiple Criteria Analysis.

### 0. ASSUNTOS TRATADOS

- O QUE ENTENDER POR TOMADA DE DECISÃO  
MULTICRITÉRIO OU MULTIOBJECTIVO
- O QUE ENTENDER POR APOIO À TOMADA DE DECISÃO?  
EXERCIDO COMO, POR QUEM E PARA QUEM?
- AS FASES FUNDAMENTAIS DO PROCESSO DE APOIO À DECISÃO
- AS ABORDAGENS MULTICRITÉRIO
  - 1) Abordagem do julgamento local interactivo
  - 2) Abordagem do critério único de síntese
  - 3) Abordagem da subordinação ("surclassement") de síntese
- O PROBLEMA DA PONDERAÇÃO DOS CRITÉRIOS

## 1. O QUE ENTENDER POR TOMADA DE DECISÃO MULTICRITÉRIO OU MULTIOBJECTIVO?

Decision making can be loosely defined as a struggle to resolve the dilemma of conflicting objectives

Milan Zeleny

Multiple Criteria Decision Making, 1982

Decidir, optar entre várias escolhas possíveis, é uma actividade vulgar do nosso dia a dia. Permanentemente, somos confrontados com variadíssimas situações decisórias: individuais ou colectivas, no campo profissional ou na nossa vida familiar, particulares ou no seio dos mais variados grupos, organizações ou sociedades, públicos ou privados.

A tomada de decisão é de facto parte integrante da vida quotidiana. Mas é também uma actividade intrinsecamente complexa e potencialmente das mais controversas, em que temos naturalmente de escolher não apenas entre possíveis alternativas de acção, mas também entre pontos de vista e formas de avaliar essas acções, enfim, de considerar toda uma multiplicidade de factores directa e indirectamente relacionados com a decisão a tomar.

Por exemplo, o processo de decidir que automóvel comprar envolve não só a pré-definição, face aos objectivos da compra, de um conjunto de modelos de entre as marcas presentes no mercado, para subsequente avaliação e escolha final, mas também a selecção de critérios para tal apropriados: preço, custos esperados de manutenção, consumo, potência, desvalorização previsível, aparência, conforto e tantos outros.

Para fazer ressaltar de forma inequívoca este carácter intrinsecamente multidimensional da tomada de decisão na vida real, talvez baste que cada um de nós pergunte a si próprio qual foi a última vez em que tomou uma decisão com base num único critério.

A relevância da chamada "Tomada de Decisão Multiobjectivo" (Multiobjective Decision Making - MODM) resulta do facto de

na maioria das situações decisoriais, no campo da engenharia, da gestão de empresas, no mundo dos negócios, aos vários níveis do sector público administrativo e empresarial do Estado, etc, estarem presentes, e terem de ser ponderados, vários objectivos geralmente conflituosos entre si. Conflituosos no sentido em que um aumento do nível de performance segundo um deles pode vir acompanhado por um decréscimo segundo algum dos outros - como, por exemplo, entre a "minimização do custo" e a "maximização da qualidade de serviço".

Repare-se que a própria relação custos-benefícios, ainda que utilizada como critério único de avaliação de projectos, estabelece implicitamente uma ponderação, uma forma de compensação ("trade-off") entre a maximização dos benefícios e a minimização dos custos.

Como diz Milan Zeleny, a tomada de decisão pode ser de forma simples definida como um esforço para resolver o dilema dos objectivos conflituosos, cuja presença impede a existência de "a solução óptima" e nos conduz para a procura da "solução de melhor compromisso". Daí a grande importância dos métodos multicritério (ou multiobjectivo) como instrumentos de apoio à tomada de decisões.

Não se quer afirmar, porém, que por vezes não seja conveniente privilegiar um só objectivo, baseando então a decisão na sua optimização sujeita a um conjunto de restrições. Mas tal não passa de um caso particular entre as situações comuns em que múltiplos objectivos ou critérios de decisão estão presentes - a realidade é em si própria multidimensional e a sua percepção pluridisciplinar. Até por que a fase de optimização (de um só objectivo), mesmo que pertinente, será sempre sido antecedida de uma outra de escolha múltipla, cronologicamente anterior àquela no todo que é o processo de decisão global, em que "o" objectivo a privilegiar fora seleccionado de entre outros possíveis.

Citando Peter Drucker (1974): "To manage a business is to balance a variety of needs and goals. And this requires multiple objectives".

Em resumo e de uma forma geral, um problema de decisão é um problema em que, face a um conjunto de objectivos, há a considerar um conjunto de soluções possíveis, alternativas,

programas - a que chamaremos **acções potenciais** - explícita ou implicitamente definidas, de entre as quais se pretende escolher a melhor acção, ou delimitar o subconjunto das boas, ou ordená-las por ordem decrescente de preferência global. Ou, tão somente, descrever as acções e caracterizar as suas múltiplas consequências, por forma a facilitar a avaliação e comparação dos seus méritos e desvantagens relativos.

As considerações anteriores mostram também que **o conceito de (tomada de) decisão não pode ser completamente dissociado do conceito de processo de (tomada de) decisão** - composto por várias fases que se sucedem ao passar-se por momentos-chave (tempos fortes) em que são tomadas decisões intermédias, na sua maioria envolvendo múltiplos factores, as quais vão formando a decisão global.

Para se ser capaz de tomar decisões face a circunstâncias progressivamente mais complexas, ter-se-à de envolver, manter e continuamente rever e actualizar todo um reportório de pontos de vista, valores, opiniões e convicções à cerca da realidade. Na prática, como diz Bernard Roy (1985) a decisão global elabora-se de uma forma mais ou menos caótica, com base na confrontação permanente de preferências de diferentes agentes (actores), ao longo de interacções concomitantes e/ou sucessivas que têm lugar entre os intervenientes no seio dos campos de interesse e poder em que se movimentam e agem. **O desenrolar destas confrontações e destas interacções constitui o processo de decisão.**

## 2. O QUE ENTENDER POR APOIO À TOMADA DE DECISÃO? EXERCIDO COMO, POR QUEM E PARA QUEM?

L' aide à la decision contribue à construire, à  
asseoir et à faire partager des convictions.

Ce sur quoi et ce par quoi s' élabore la décision  
doivent pouvoir faire l' object d' une discussion  
critique (...)

Celui qui prend en charge l' aide à la decision  
est généralement distinct du décideur.

Bernard Roy

Méthodologie Multicritère d' Aide à la Decision, 1985

É a necessidade crescentemente sentida de fazer face, de forma sistematizada e mais ou menos formalizada, à complexidade característica da tomada de decisão e de disciplinar o seu processo procurando uma orientação clara que na prática permita combater o seu desenrolar caótico, no sentido antes expresso, que justificam o desenvolvimento, também dos métodos multicritério, mas principalmente e de forma mais abrangente, das metodologias multicritério de apoio à tomada de decisão.

Escolher e preferir são tarefas que o decisor tem de exercer por si próprio - ninguém pode realizá-las por ele, ninguém pode tomar o seu lugar. Mesmo quando, em desespero, ele se abandona ao destino e decide não decidir (Zeleny, 1982).

Claro que o decisor pode socorrer-se da ajuda, do apoio de um analista, de um consultor que, por sua vez, servindo-se de um conjunto de instrumentos, entre eles os **métodos multicritério**, procura apoiar o decisor ao longo do desenrolar do processo de decisão. Mas mesmo quando as várias opções de decisão e os objectivos tiverem sido claramente identificados e os graus de atingimento destes por cada uma daquelas satisfatoriamente medidos, nenhuma técnica computacional pode substituir-se ao decisor, nenhum consultor pode simplesmente assumir por ele a responsabilidade final da decisão.

É essa faceta acima de tudo pessoal do acto de preferir, pela subjectividade característica de cada um ao perceber a realidade, é o carácter essencialmente criativo da tomada de decisão, com a presença participativa da inspiração, da imaginação, da vivência própria de cada decisor, mais ou menos formalizáveis mas não tecnocraticamente substituíveis, que fazem considerar o papel do consultor (analista), não como o de um simples manipulador diplomado, um operador habilitado de "técnicas todo poderosas e infalíveis de tomada de decisão", mas sim o de um conselheiro, um orientador cujo recurso a modelos claramente explicitados pode ajudar poderosamente a compreender e dominar as causas e consequências de todo um conjunto de comportamentos e atitudes, de eventos e de fenómenos particulares, inesperados ou não, que ocorrem no desenrolar do processo; ou ainda, a construir respostas às questões que se colocam no seio de uma entidade decisora, colectiva ou individual. Modelos que permitem a esse analista

o exercício de um papel activo, fornecendo-lhe o "know-how" para aumentar no sentido de fazer admitir como válidas as suas prescrições (Roy, 1985).

Tomando como referência a atitude tradicional da investigação operacional que visa a prescrição de "a" solução óptima, a concepção descrita de apoio à decisão aparece como muito ampla. Mas, por outro lado, revelar-se-à voluntariamente restrita ao constatar-se que ela se restringe ao uso de modelos claramente explicitados (Jacquet-Lagrèze, 1985).

### 3. AS FASES FUNDAMENTAIS DO PROCESSO DE APOIO À DECISÃO

Duas fases fundamentais do processo de apoio à decisão, diferenciadas mas intrinsecamente ligadas, podem ser identificadas nas metodologias multicritério de avaliação:

Uma fase de análise do sistema em estudo, conducente à identificação, caracterização e hierarquização dos principais actores intervenientes e à explicitação das alternativas de decisão potenciais, que se pretendem comparar entre si, em termos dos seus méritos e desvantagens relativos face a um conjunto de critérios de avaliação, nesta fase definidos de acordo com os pontos de vista dos actores. Em termos gerais esta fase trata da formulação do problema e da identificação do objectivo de topo do processo de avaliação ("problemática", na terminologia de Roy (1985); ver Bana e Costa (1986b)).

Uma outra fase, de avaliação propriamente dita no sistema, que se poderá apelidar de fase de síntese, cuja finalidade é esclarecer "a escolha", recorrendo à aplicação de métodos multicritério para apoiar a modelização das preferências dos actores e a sua agregação.

Em presença dos muitas vezes chamados "problemas multi-objectivo", isto é, de um conjunto infinito (ou muito grande) de alternativas, definidas implicitamente através de um conjunto de restrições, tornam-se particularmente adequados os métodos de programação matemática multiobjectivo, cuja descrição se remete para a extensa bibliografia (Cochrane e Zeleny, 1973; Bell et al, 1977; Cohon, 1978; Hwang e Masud, 1979; Goecochea et al, 1982; Zeleny, 1982; Chankong e Haimes, 1983; Zionts, 1985; Szidarovszky et al, 1986; etc).

Por outro lado, quando estamos perante um problema multicritério (discreto), isto é, de um conjunto pequeno de alternativas explicitamente definidas através das suas valorações segundo os vários critérios, o resultado da fase de análise concretiza-se numa "matriz multicritério de avaliação" dessas alternativas.

Seja:

-  $A = \{a_1, \dots, a_i, \dots, a_n\}$ , o conjunto das alternativas;

-  $F = \{g_1, \dots, g_j, \dots, g_m\}$ , o conjunto dos critérios de avaliação;

-  $g_j(a_i)$  o valor da alternativa  $a_i$  segundo o critério  $g_j$ .

Considere-se ainda que, se  $g_j(a_k) > g_j(a_h)$ , então  $a_k$  é melhor que  $a_h$  segundo o critério  $g_j$ , sendo  $a_k$  e  $a_h$  duas quaisquer alternativas de  $A$ .

A matriz de avaliação será da forma:

	$g_1$	...	$g_j$	...	$g_m$
$a_1$	$g_1(a_1)$	...	$g_j(a_1)$	...	$g_m(a_1)$
.	.		.		.
.	.		.		.
.	.		.		.
$a_i$	$g_1(a_i)$	...	$g_j(a_i)$	...	$g_m(a_i)$
.	.		.		.
.	.		.		.
.	.		.		.
$a_n$	$g_1(a_n)$	...	$g_j(a_n)$	...	$g_m(a_n)$

Em muitas situações, a explicitação das alternativas a comparar revela-se uma fase crucial e difícil do apoio à decisão (ver neste contexto, sobre o qual não nos deteremos, Alexander, 1978; Nijkamp e Spronk, 1981; Arbel e Tong, 1982; Ozerney, 1985).

Tal é também extensível ao problema da definição dos critérios de avaliação. A este propósito importa dizer que a montante da construção dos critérios o analista confronta-se

com a tarefa de identificação dos pontos de vista a considerar, entendendo-se por "ponto de vista" todo o aspecto da realidade que um actor (pessoa ou organização) considera como importante para escolher entre as várias alternativas. Em presença de mais do que um actor o número de pontos de vista pode, facilmente, revelar-se enorme, incomportável para qualquer forma de agregação, impondo-se assim a sua selecção.

Como frisa de Montgolfier (1975), os pontos de vista a reter não deverão ser muitos para que a comparação se mantenha praticável, mas também não deverão ser tão poucos que levem a negligenciar factores importantes e interesses fundamentais dos actores. Estes aspectos são cruciais para que o analista possa definir um conjunto de critérios aceite por todos (entendendo por critério uma variável real que permite expressar matematicamente um ponto de vista).

Muitas das técnicas de análise multicritério permitem apenas a agregação de critérios cardinais, isto é, aqueles a cuja avaliação esteja necessariamente subjacente uma medida (Roy, 1974a). A sua aplicabilidade prática limita-se assim a problemas envolvendo pontos de vista unicamente quantitativos e em que sejam suficientemente precisos os indicadores escolhidos para avaliar a prestação de cada alternativa segundo cada critério. Uma descrição geral de vários destes "métodos multicritério quantitativos" pode ser encontrada em (Hwang e Yoon, 1981) e, particularmente sobre os mais aplicados em planeamento Territorial, em (Nijkamp e Voogd, 1985).

Mas, muitas situações decisórias são caracterizadas pela presença de pontos de vista de natureza qualitativa, que regra geral só podem ser expressos através de variáveis ordinais, ou mesmo binárias ou nominais. Para fazer face a este tipo de situações têm vindo a ser desenvolvidos "métodos multicritério qualitativos", particularmente pela que poderemos denominar como "escola holandesa" (ver Paelink, 1976; Nijkamp e Van Delft, 1977; Nijkamp, 1979 e 1981; Voogt, 1982; Hinloopen et al, 1983a).

#### 4. AS ABORDAGENS MULTICRITÉRIO

O reconhecimento de que cada actor é portador de um siste-

ma de preferências próprio é comum aos diferentes tipos de abordagens operacionais desenvolvidas para conduzir o processo de agregação das avaliações parciais das várias alternativas segundo os vários critérios. É assim fundamental fazer explicitar pelos actores as suas preferências.

É nos princípios subjacentes aos processos de modelização das preferências em que se baseiam que se diferenciam entre si os três grandes tipos de abordagens de síntese (Roy, 1985), cada qual abrangendo diversos métodos multicritério:

- 1) Abordagem do julgamento local interactivo;
- 2) Abordagem do critério único de síntese
- 3) Abordagem da subordinação ("surclassement") de síntese

1) Na abordagem do julgamento local interactivo não se procura, como nas outras duas abordagens, a explicitação à priori de uma regra capaz de dar uma resposta de síntese, exaustiva e definitiva ao problema da agregação dos critérios (Vincke, 1982). O processo de modelização das preferências antes consiste numa sequência de interações entre o analista e o decisor (Roy, 1985) - centradas de cada vez sobre um pequeno número de alternativas e suas consequências -, alternadas com fases de cálculo em que o analista selecciona, de acordo com as respostas anteriores do decisor, uma alternativa que submete no diálogo seguinte à sua apreciação. As reacções do decisor permitirão ao primeiro recolher informações suplementares, o que o conduzirá ao cálculo de uma nova solução de compromisso. As fases de diálogo e cálculo suceder-se-ão até que o decisor se dê por satisfeito com a alternativa proposta como melhor. No entanto, muitos métodos interactivos apresentam o inconveniente de convergirem para uma solução independentemente ao decisor.

Vanderpooten (1987) propõe uma classificação dos principais métodos interactivos. De entre estes refiram-se o método STEM (Benayoun et al, 1971), os métodos de Geoffrion et al (1972), Zionts e Wallenius (1976), Steuer (1986), Vincke (1976) e o modelo PREFCALC (Jacquet-Lagrèze e Shakun, 1984), os dois últimos sem preocupações de convergência.

A descrição dos métodos interactivos, fundamentalmente desenvolvidos no âmbito da Programação Matemática Multiobjectivo, pode ser encontrada em Goicoechea et al (1982), Vincke (1982) ou Steuer (1986), entre outras obras.

2) Na abordagem do critério único de síntese, em que se insere fundamentalmente a chamada "Escola Americana", a modelização das preferências é feita através da construção de uma função de agregação  $V(g_1, \dots, g_j, \dots, g_m)$ , que estabelece em A uma estrutura de pré-ordem completa. A função de valor  $V(\underline{g})$  é como que um critério único, agregando os m critérios  $g_j$  ( $j=1, \dots, m$ ).

O princípio subjacente a esta abordagem é o da completa comparabilidade transitiva (Roy, 1974a; Roy e Vincke, 1984), por se admitirem apenas duas situações de preferência global, mutuamente exclusivas e ambas sistematicamente supostas transitivas: preferência estrita (P) e indiferença (I). Exclui-se assim qualquer situação de incomparabilidade entre alternativas.

Este é o modelo de agregação da Teoria das Escolhas Sociais (Arrow, 1963) e dos métodos desenvolvidos no seio da Teoria da Utilidade Multiatributo (Fishburn, 1970; Keeney e Raiffa, 1976), fundada nos princípios axiomáticos de Von Neumann e Morgenstern (1947).

O conceito mais importante é o de valor ou taxa (marginal) de substituição, que "formaliza a noção de compensação (ver Bouyssou, 1986) e generaliza a noção de preço" (Jacquet-Lagrèze, 1985).

Numa primeira definição aproximativa podemos dizer que a taxa de substituição marginal  $s_{jr}$  de um critério  $g_j$  em relação a um critério de referência  $g_r$ , é o acréscimo que é preciso dar segundo  $g_r$  para compensar a perda de uma unidade segundo  $g_j$ .

Compensar significa que a alternativa  $a_k$  de A definida por  $(g_1(a_k), \dots, g_j(a_k), \dots, g_r(a_k), \dots, g_m(a_k))$  é, para o decisor, indiferente a alternativa, real ou fictícia, definida por  $(g_1(a_k), \dots, g_j(a_k) - 1, \dots, g_r(a_k) + s_{jr}^k, \dots, g_m(a_k))$ . Assim, o valor da taxa de substituição não é necessariamente constante

em cada ponto  $K$ , no espaço  $\mathbb{R}^m$  dos critérios, de coordenadas  $(g_1(a_k), \dots, g_j(a_k), \dots, g_r(a_k), \dots, g_m(a_k))$ .

Se a função de valor global  $V(\underline{g})$  fôr diferenciável, a definição matemática precisa de  $s_{jr}^K$  é dada pela expressão:

$$s_{jr}^K = \left( \frac{\partial V(\underline{g})}{\partial g_j} \right)_K / \left( \frac{\partial V(\underline{g})}{\partial g_r} \right)_K$$

No caso de funções de agregação aditivas, do tipo

$$V(\underline{g}) = \sum_{j=1}^m v_j(g_j) \text{ - em que } v_j(g_j) \text{ é a função de valor marginal}$$

segundo o critério  $g_j$  -, a taxa de substituição entre dois critérios é independente dos valores segundo todos os outros  $m-2$  critérios:

$$s_{jr}^K = \left( \frac{\partial v_j(g_j)}{\partial g_j} \right)_K / \left( \frac{\partial v_r(g_r)}{\partial g_r} \right)_K, \text{ uma vez que:}$$

$$\left( \frac{\partial V(\underline{g})}{\partial g_j} \right)_K = \left( \frac{\partial v_j(g_j)}{\partial g_j} \right)_K, \quad \forall g_j \in F$$

Isto é, uma função  $V$  do tipo aditivo só é válida, como demonstra por exemplo Fishburn (1970), se e só se os  $m$  critérios forem independentes entre si. A taxa de substituição entre dois critérios, no ponto  $K$ , é neste caso a razão entre as inclinações (das tangentes às curvas) das duas funções de valor marginal, nesse ponto.

O modelo da soma ponderada é um caso especial do modelo aditivo, muito conhecido e aplicado na prática:

$$V(\underline{g}) = \sum_{j=1}^m w_j g_j.$$

Neste modelo as funções de valor marginal  $u_j(g_j) = w_j g_j$  são

lineares (rectas de inclinação  $w_j$ ) e as curvas de indiferença são rectas (planos, ou hiper-planos) no espaço dos critérios. Assim, o valor da taxa de substituição entre dois critérios é constante em qualquer ponto e dado por:

$s_{jr} = w_j/w_r$ , uma vez que:

$$\left( \frac{v_j(g_j)}{g_j} \right)_K = w_j, \quad \forall g_j \in F \quad \forall a_k \in A$$

De forma simples podemos dizer que  $w_j$  representa a importância relativa do critério  $g_j$ , a que é usual chamar "peso". Mais correctamente, trata-se de um factor de escalarização ("scaling factor" - ver Vansnick, 1984).

Na prática, a grande dificuldade inerente ao processo de modelização reside na determinação das taxas de substituição em vários pontos no espaço dos critérios (Raiffa, 1969), que permitirão a definição da função de valor global (critério único de síntese) explicitando as preferências do decisor. Na maioria das situações a função de valor  $V$  não é estável no espírito do decisor: quer porque basta que este atribua um pouco mais de importância a um critério para que possa mudar consideravelmente  $V$ , quer porque as apreciações do decisor sobre a importância relativa dos critérios são significativamente variáveis no tempo (Roy et al, 1975).

Que dizer então quando vários decisores ou actores estão presentes e cujas preferências importa modelizar?

Se para um só decisor está longe de ser fácil atribuir valores precisos às taxas de substituição, perante múltiplos actores, portadores de sistemas de valores próprios e muitas vezes conflituosos, torna-se irrealista querer definir uma função de agregação única de consenso. "O Teorema de Arrow mostra, de forma exemplar, como é nesses casos difícil, para não dizer impossível, construir uma tal regra de agregação de carácter geral" (Roy, 1974b).

3) Na abordagem da subordinação de síntese, introduzida pela chamada "Escola Francesa", a modelização das preferências é feita através da construção de uma relação binária em  $A$ , chamada relação de subordinação ( $S$ ), com o significado

seguinte: dadas duas alternativas  $a_k$  e  $a_h$ ,  $a_k$  subordina  $a_h$  ( $a_k S a_h$ ) se existirem argumentos suficientemente fortes para admitir que, aos olhos do decisor,  $a_k$  é pelo menos tão boa como  $a_h$ . Consequentemente,  $a_k$  é indiferente ( $a_k S a_h$  e  $a_h S a_k$ ) ou estritamente preferível a  $a_h$  ( $a_k S a_h$  e  $a_h \not S a_k$ ).

Quando os argumentos em favor da afirmação " $a_k$  é pelo menos tão boa como  $a_h$ " são julgados insuficientes, então  $a_k$  não subordina  $a_h$ . Consequentemente,  $a_h$  é preferível ( $a_k \not S a_h$  e  $a_h S a_k$ ) ou incomparável a  $a_k$  ( $a_k \not S a_h$  e  $a_h \not S a_k$ ).

A relação  $S$  modeliza assim apenas a parte segura das preferências, isto é, a que pode ser estabelecida com razões suficientes, face à informação disponível.  $S$  é uma relação não necessariamente pressuposta transitiva e completa.

O princípio subjacente à abordagem da subordinação de síntese é o expresso pelo axioma da comparabilidade parcial (Roy, 1985), segundo o qual três situações fundamentais de preferência podem ser encontradas no processo de modelização: incomparabilidade (R), preferência estrita (P) e indiferença (I) - distinguindo-se as duas últimas das situações equivalentes na abordagem do critério único de síntese por as relações binárias correspondentes poderem ser não transitivas. Em certas circunstâncias admite-se ainda uma quarta situação, chamada de preferência fraca (Q), por forma a levar em conta casos de indecisão entre P e I.

Os argumentos encontrados pelo analista como favoráveis à validação da subordinação de  $a_h$  por  $a_k$  podem ser por ele consideradas como susceptíveis de envolver um maior ou menor risco ao estabelecer que, de acordo com o sistema de preferências do decisor,  $a_k$  é globalmente pelo menos tão boa como  $a_h$ . Matematicamente, este grau de convicção na subordinação pode ser expresso associando à relação  $S$  uma "função de credibilidade"  $C(a_k, a_h)$ , de domínio  $[0,1]$ , tal que (Roy, 1977):

- $C(a_k, a_h)=1$  expressa uma convicção absoluta na subordinação de  $a_h$  por  $a_k$ ;
- $C(a_k, a_h)=0$  expressa uma convicção absoluta na ausência de subordinação de  $a_h$  por  $a_k$ ;

- $C(a_k, a_h)$  é uma função não decrescente das avaliações parciais de  $a_k$  ( $g_j(a_k)$ ,  $j=1, \dots, m$ ) e não crescente das avaliações parciais de  $a_h$  ( $g_j(a_h)$ ,  $j=1, \dots, m$ ).

Assim,  $C(a_k, a_h)$  varia obviamente em função das diferenças entre as avaliações parciais de  $a_k$  e  $a_h$  segundo os vários critérios  $g_j$ :

$$C(a_k, a_h) = f(g_1(a_k) - g_1(a_h), \dots, g_j(a_k) - g_j(a_h), \dots, g_m(a_k) - g_m(a_h))$$

Teoricamente, a relação de subordinação assim definida ( $S^c$ ) é uma relação binária vaga ("fuzzy"), conceito introduzido por Zadeh (1970) e estudado no âmbito da tomada de decisão por vários autores (Zadeh et al, 1975; Blin, 1977; Orlovsky, 1978; Zimmermann et al, 1984). Roy (1977) apresenta várias formas de construção de relações de subordinação vagas.

Diga-se, entretanto, que muitos métodos multicritério que seguem a abordagem da subordinação de síntese constroem e exploram relações de subordinação não vagas (equivalente a fazer  $C \in \{0, 1\}$ ). Citem-se, de entre os mais conhecidos e aplicados, os métodos ELECTRE I e IS (Buffet et al, 1967; Roy 1968; Roy e Skalka, 1984), ELECTRE II (Roy e Bertier, 1973), Q-Analysis (Duckstein e Kempf, 1979), etc.

Por seu turno, ELECTRE III (Roy, 1978 ; Roy et al, 1986) e PROMETHEE I e II (Brans et al, 1984 e 1986; Brans e Vincke, 1985; Mareschal, 1983) são exemplos de métodos que constroem e exploram relações de subordinação vagas.

Na prática a construção de relações de subordinação, vagas ou não, pode ser baseada em noções variadas, cuja descrição geral pode ser encontrada em (Roy, 1973, 1974a e 1985) e em (Roy, 1977) para o caso de relações vagas: concordância e discordância (procurando modelizar situações não compensatórias - ver Bouyssou e Vansnick, 1985), degradação (Yu, 1973), extrapolação lógica (Roy, 1973), compensação (Roy, 1973; Jacquet-Lagrèze, 1975; Bana e Costa, 1988).

Para além da forma de construção de  $S$ , os métodos de subordinação diferem ainda no processo como exploram essa relação (Roy e Vincke, 1981).

Cada uma das duas etapas de construção e exploração da

relação de subordinação varia e deve ser tratada de acordo com as características próprias do problema em estudo e dos actores envolvidos, que conduzem o analista à escolha do método ou métodos cuja aplicação na fase de síntese considera mais adequada à situação decisional particular que enfrenta.

Winkels e Wascher (1981) e Siskos et al (1983) fazem resenhas bibliográficas bastante completas sobre as abordagens de subordinação.

## 5. O PROBLEMA DA PONDERAÇÃO DOS CRITÉRIOS

Um problema fundamental subjacente à agregação é o da explicitação das ponderações dos vários critérios, sejam elas expressas através de taxas de substituição - como nos métodos das várias abordagens de síntese que se baseiam no conceito de compensação -, sejam elas encaradas como reflectindo directamente os graus de importância relativa dos vários critérios - como nos métodos não compensatórios de subordinação que se baseiam nos conceitos de concordância e discordância.

É pois natural que na literatura se possam encontrar descrições e comparações de muitas técnicas desenvolvidas para determinar os valores dos coeficientes de ponderação dos critérios (Eckenrode, 1975; Hwang e Yoon, 1981; Goicochea et al, 1982; Zeleny, 1982; Vansnick et al, 1984; Solymosi e Dombi, 1986; Vansnick et al, 1986a e b; etc). Em língua portuguesa, Barreto (1985) faz uma resenha descritiva de alguns dos métodos de ponderação mais conhecidas.

Note-se, no entanto, que dadas as diferenças de significação teórica do conceito de ponderação inerente aos diversos métodos de análise multicritério, não tem qualquer sentido querer explicitar os valores dos coeficientes de ponderação independentemente e a montante da selecção do método de agregação mais adequado ao problema em estudo e ao tipo de sistemas de preferência dos actores respectivos. E isto porque o processo de modelização constitui um todo do qual a explicitação das preferências dos actores pelos vários critérios de avaliação é parte não dissociável.

No quadro metodológico da Teoria de Utilidade Multiatri-

buto, Keeney e Raiffa (1976) apresentam um processo prático de ponderação, composto por uma sequência de perguntas, para determinação das taxas de substituição. Por seu turno, Roy et al (1986) desenvolvem um outro processo, com vista à explicitação dos índices de importância dos vários critérios, num quadro metodológico não compensatório, no seio do qual passam a ter sentido questões, por exemplo, do tipo "o critério  $g_j$  é de maior, menor ou da mesma importância comparativamente com o critério  $g_j$ ".

Mas as respostas dos actores têm uma natureza essencialmente qualitativa e/ou imprecisa. Ora, paradoxalmente, tais processos, como a maioria das técnicas de ponderação, são utilizados para obter um vector de valores precisos para os coeficientes de ponderação. O que é uma forte abstracção da realidade que na prática do apoio à decisão somos conduzidos a aceitar por a grande maioria dos métodos de agregação não operar com informação qualitativa nos "pesos".

Excepção feita a alguns poucos métodos desenvolvidos precisamente para ser possível levar em consideração informação imprecisa nos critérios, geralmente do tipo ordinal, como o métodos QUALIFAX (Paelinck, 1976) e ORESTE (Roubens, 1982), ou a REGIME ANALYSIS (Hinloopen et al, 1983b).

Os métodos ELECTRE IV (Roy e Hugonnard, 1982) - no contexto da abordagem da subordinação de síntese -, TRIDENT (Tavares, 1984 e 1986) - no contexto da abordagem do critério único de síntese - e TRIMAP (Clímaco e Antunes, 1987) - no contexto da abordagem do julgamento local interactivo - procuram abordar problemas caracterizados pela ausência de informação sobre as importâncias relativas dos critérios (os dois últimos limitados a 3 critérios).

Na mesma linha, refira-se ainda a abordagem apresentada em (Bana e Costa, 1988) sob o nome de "OUTWEIGH ANALYSIS" e também o método proposto pelo autor na 2ª parte da comunicação que está na origem deste artigo. Este método é um desenvolvimento das ideias introduzidas em (Bana e Costa, 1986a) tendo em vista enfrentar situações multicritério caracterizadas pela presença de vários actores. Aceita-se um quadro metodológico compensatório no seio do qual, em vez de pretender-se, irrealisticamente, encontrar valores únicos para as taxas de substituição, modelizam-se as preferências dos

actores através da explicitação por cada um deles de intervalos de variação dentro dos quais aceitam, de acordo com os seus sistemas de valores individuais, estarem contidos os valores daquelas taxas.

#### REFERÊNCIAS BIBLIOGRÁFICAS

- Alexander, E.R. (1978) - "The design of alternatives in organizational context: A pilot Study", *Administrative Science Quarterly*, 24 (382-404).
- Arbel, A. e Tong, R.M. (1982) - "On the generation of alternatives in decision analysis problems", *The Journal of the Operational Research Society*, 33(4) (337-387).
- Arrow, K.J. (1963) - *Social Choice and Individual Values*, Yale University Press, New Haven.
- Bana e Costa, C.A. (1986a) - "A multicriteria decision aid methodology to deal with conflicting situations on the weights", *EJOR*, 26(1) (22-34).
- Bana e Costa, C.A. (1986b) - "A problemática da rejeição em decisão multicritério", *Investigação Operacional*, 6(1/2) (95-104).
- Bana e Costa, C.A. (1988) - "A methodology for sensitivity analysis in three-criteria problems: a case study in municipal management", *EJOR*, 33 (2) (159-173).
- Barreto, L.S. (1985) - "Ponderação de Atributos", Instituto Superior de Agronomia, Lisboa.
- Bell, D.E.; Keeney, R.L. e Raiffa, H. (1977) - *Conflicting Objectives in Decisions*, Wiley, New York.
- Benayoun, R.; de Montgolfier, J.; Tergny, J. e Larichev, O. (1971) - "Linear programming with multiple objective functions: STEP Method (STEM)", *Mathematical Programming*, 1 (366-375).
- Blin, J.M. (1977) - "Fuzzy sets in multiple criteria decision-making" em M.K. Starr e M. Zeleny (eds.) *Multiple Criteria Decision Making*, North Holland, Amsterdam (129-146).
- Bouyssou, D. (1986) - "Some remarks on the notion of compensation in MCDM", *EJOR*, vol.26-1 (150-160).
- Bouyssou, D. e Vansnick, J.-C. (1985) - "Noncompensatory and generalised noncompensatory preferences structures", *Cahiers du Lamsade*, n°59, Université de Paris-Dauphine, Paris.

- Brans, J.P.; Mareschal, B. e Vincke, P. (1984) - "PROMETHEE: A new family of multicriteria analysis", in J.P. Brans (ed.), *Proceedings of the IFORS 84 Conference*, Washington (408-421).
- Brans, J.P. e Vincke, Ph. (1985) - "A preference ranking organization method, the PROMETHEE method", *Management Science* 31 (647-656).
- Brans, J.P.; Vincke, Ph. e Mareschal, B. (1986) - "How to select and how to rank projects: The PROMETHEE method", *EJOR*, vol.24-2 (228-238).
- Buffet, P.; Grémy, J.P. e Marc, M. (1967) - "Peut-on choisir en tenant compte de critères multiples? Une méthode (ELECTRE) et trois applications, *Révue Metra*, vol.6 (283-316).
- Chankong, V. e Haimes, Y.Y. (1983) - *Multiobjective Decision Making, Theory and Methodology*, North Holland, Amsterdam.
- Clímaco, J.C.; Antunes, C.H. - "TRIMAP - a three criteria interactive linear programming package", a aparecer em *Foundations of Control Engineering*, 1987.
- Cochrane, J.L. e Zeleny, M. (1973) - *Multiple Criteria Decision Making*, University of South Carolina Press, Columbia, South Carolina.
- Cohon, J.L. (1978) - *Multiobjective Programming and Planning*, Academic Press, New York.
- Drucker, P.F. (1974) - *Management: Tasks, Responsibilities, Practices*, Harper & Row, New York.
- Duckstein, L. e Kempf, J. (1981) - "Multicriteria Q-Analysis for plan evaluation", em P.Nijkamp e J.Spronk (eds.) *Multiple Criteria Analysis: Operational Methods*, Gower, Aldershot (87-99).
- Eckenrode, R.T. (1965) - "Weighting multiple criteria" *Management Science*, vol.12-3 (172-191).
- Fishburn, P.C. (1970) - *Utility theory for Decision Making*, Wiley, New York.
- Geoffrion, A. R.; Dyer, J.S. e Feinberg, A (1972) - "An interactive approach for multi-criterion optimization, with an application to the operation of an academic department, *Management Science*, 19-4 (357-368).
- Goicochea, A.; Hansen, R. e Duckstein, L. (1982) - *Multi-objective Decision Analysis with Engineering and Business Applications*, Wiley, New York.
- Hwang, C.-L. e Masud, A.S. (1979) - *Multiple Objective Decision Making - Methods and Applications: A State-of-the-Art Survey*, *Lecture Notes in Economics and Mathematical*

- Systems, vol.164, Springer, New York.
- Hwang, C.-L. e Yoon, K. (1981) - **Multiple Attribute Decision Making, Methods and Applications: A State-of-the-Art Survey**, Lecture Notes in Economics and Mathematical Systems, vol.186, Springer, New York.
- Hinloopen, E.; Nijkamp, P. e Rietveld, P. (1983a) - "Qualitative discrete multiple criteria choice models in regional planning", **Regional Science and Urban Economics**, vol.13 (77-103).
- Hinloopen, E.; Nijkamp, P. e Rietveld, P. (1983b) - "The regime method: a new multicriteria technique", em P.Hansen (ed.), **Essays and Surveys on Multicriteria Decision Making**, Lecture Notes in Economics and Mathematical Systems, vol. 209, Springer, New York (146-155).
- Jacquet-Lagrèze, E. (1975) - "How can we use the notion of semi-orders to build outranking relations in multi-criteria decision making", em D.Wendt e C.Vlek (eds.), **Probability and Human Decision Making**, Reidel, Dordrecht (87-112).
- Jacquet-Lagrèze, E. (1985) - "Basic concepts for multicriteria decision", em G.Fandel e J.Spronk (eds.) **Multiple Criteria Decision Methods and Applications**, Springer, New York (11-26).
- Jacquet-Lagrèze, E. e Shakun, M.F. (1984) - "Decision Support systems for semi-structured buying decisions. EJOR, vol. 16-1 (48-58).
- Keeney, R.L. e Raiffa, H. (1976) - **Decisions with Multiple Objectives, Preferences and Value Tradeoffs**, Wiley, N.Y..
- Mareschal, B. (1983) - **Étude Theorique et Pratique des Méthodes de Surclassement en Analyse Multicritère**, Université Libre de Bruxelles, Bruxelles.
- de Montgolfier, J. (1975) - "Autoroute au forêt?", **Futuribles** N°3 (40-59).
- Nijkamp, P. (1979) - **Multidimensional Spacial Data and Decision Analysis**, Wiley, Chichester/New York, 1977.
- Nijkamp, P. (1981) - "Qualitative evaluation models with conflicting priorities", **Papers of the Regional Science Association**, vol.46 (3-14).
- Nijkamp, P. e Spronk, J. (1981) - **Multiple Criteria Analysis: Operational Methods**, Gower, Aldershot, 1981.
- Nijkamp, P. e Van Delft, A. (1977) - **Multi-Criteria Analysis and Regional Decision-Making**, Martinus Nijhoff, The Hague.
- Nijkamp, P. e Voogd, H. (1985) - "An informal introduction to multicriteria evaluation", em G.Fandel e J.Spronk (eds.)

- Multiple Criteria Decision Methods and Applications**, Springer, New York (61-84).
- Orlovsky, S.A. (1978)** - "Decision-making with a fuzzy preference Relation", **Fuzzy sets and systems**, 1 (155-167).
- Ozernoy, U.M. (1985)** - "Generating alternatives in multiple criteria decision making problems: a survey", em Y.Y.Haimmes and V.Chankong (eds.), **Decision Making with Multiple Objectives**, Lectures Notes in Economics and Mathematical Systems, Vol.242, Springer, New York (322-330).
- Paelinck, J.H.P. (1976)** - "Qualitative multiple criteria analysis, environmental protection and multiregional development", **Papers of the Regional Science Association**, 36 (59-74).
- Raiffa, H. (1969)** - "Preferences for multi-attributed alternatives", **The Rank Corporation, Memorandum RM-5868**.
- Roy, B. (1968)** - "Classement et choix em présence de points de vue multiples (la méthode ELECTRE)", **RIRO**, n<sup>o</sup>8 (57-75).
- Roy, B. (1973)** - "How outranking relation helps multiple criteria decision making", em J.L.Cochrane and M.Zeleny (eds.) **Multiple Criteria Decision Making**, University of South-Carolina Press, Columbia, S.C. (179-201).
- Roy, B. (1974a)** - "Critères multiples et modélisation des préférences ( l' apport des relations de surclassement)", **Révue d' Economie Politique**, n<sup>o</sup>1, (1-44).
- Roy, B. (1974b)** - "La modélisation des préférences: un aspect crucial de l' aide à la decision", **Révue METRA**, vol.13 (135-153).
- Roy, B. (1977)** - "Partial preference analysis and decision-aid: the fuzzy outranking relation concept, em E.Bell; R.Keeney e H.Raiffa (eds.), **Conflicting Objectives in Decisions**, Wiley, New York (40-75).
- Roy, B. (1978)** - "ELECTRE III: un algorithme de classement fondé sur une représentation floue des préférence en présence de critères multiples", **Cahiers du CERO**, 20 (3-24).
- Roy, B. (1985)** - **Méthodologie Multicritère d' Aide à la Décision**, Economica, Paris.
- Roy, B. (1987)** - "Meaning and validity of interactive procedures as tools for decision aid", **EJOR**, vol. 31-3 (297-303).
- Roy, B. e Bertier, P. (1973)** - "La méthode ELECTRE II: une application on media-planning", in M. Ross (ed.), **OR<sup>72</sup>**, North-Holland, Amsterdam (291-302).
- Roy, B. e Skalka, J.M. (1984)** - "ELECTRE IS - aspects méthodologiques et guide d' utilisation", **Documents du Lamsade**,

- n<sup>o</sup>30, Université de Paris-Dauphine, Paris.
- Roy, B. e Vincke, Ph. (1981) - "Multicriteria analysis: survey and new directions", *EJOR*, vol.8-3 (207-218).
- Roy, B. e Hugonnard, (1982) - "Ranking of suburban line extension projects on the Paris metro system by a multicriteria method", *Transportation Research*, vol.16A. n.4 (301-312).
- Roy, B. e Vincke, Ph. (1984) - "Relational systems of preference with one or more pseudo-criteria: some new concepts and results", *Management Science*, vol.30-11 (1332-1335).
- Roy, B.; Vincke, Ph. e Brans, J.P. (1975) - "Aide à la décision multicritère", *Révue Belge de Statistique, d'Informatique et de Recherche Opérationnelle*, Vol.15-4.
- Roy, B.; Présent, M. e Silhol, D. (1986) - "A programming method for determining which Paris metro stations should be renovated", *EJOR*, vol.24-2 (318-335).
- Roubens, M. (1982) - "Preference relations on actions and criteria in multi-criteria decision making", *EJOR* Vol.10-1 (51-55).
- Siskos, J.; Wascher, G. e Winkels, H.M. (1983) - "A bibliography on outranking approaches (1966-1982)", *Cahiers du Lamsade*, n<sup>o</sup>45, Université de Paris-Dauphine, Paris.
- Solymosi, T. e Dombi, J. (1986) - "A method for determining the weights of criteria: the centralised weights", *EJOR*, vol.26-1 (35-41).
- Steuer, R.E., (1986) - "Multiple Criteria Optimization: Theory, Computation and Application, Wiley, New York.
- Szidarowsky, F.; Gershman, M.E. e Duckstein, L. (1986) - *Techniques for Multiobjective Decision Making in Systems Management*, Elsevier, Amsterdam.
- Tavares, L.V. (1984) - "The TRIDENT approach to rank alternative tenders for large engineering projects", *Foundations of Control Engineering*, vol.9-4 (181-191).
- Tavares, L.V. (1986) - "Multicriteria scheduling of a railway renewal program (case study)", *EJOR*, vol.26-3 (395-405).
- Vanderpooten, D. (1987) - "L'approche interactive dans l'aide à la décision multicritère", *Document du Lamsade*, n<sup>o</sup>38, Université de Paris-Dauphine, Paris.
- Vansnick, J.-C. (1984) - "Strength of preference. Theoretical and practical aspects, em: J.P.Brans (ed.), *Operational Research* 84, North-Holland, Amsterdam (449-463).
- Vansnick, J.-C. (1986a) - "On the problem of weights in multiple criteria decision making (the non compensatory approach)", *EJOR*, vol.24-2 (288-294).

- Vansnick, J. C. (1986b) - "De Borda et Condorcet à l'agrégation multicritère", *Cahiers du Lamsade*, n<sup>o</sup>70, Université de Paris-Dauphine, Paris.
- Vansnick, J.-C.; Pirlot, M.; Schepers, G. e Couvreur, P. (1984) - "TACTIC: a noncompensatory decision-aid method with veto", *Working Paper 1984/3*, Université de Mons-Hainaut, Mons.
- Vincke, Ph. (1976) - "Une méthode interactive en programmation linéaire à plusieurs fonctions économiques", *RAIRO* vol.10-6 (5-20).
- Vincke, Ph. (1982) - "Présentation et analyse de neuf méthodes multicritères interactives," *Cahiers du Lamsade* n<sup>o</sup>42, Université de Paris-Dauphine, Paris.
- Von Neumann, J. e Morgenstern, O. (1947) - *Theory of Games and Economic Behaviour*, Princeton University Press.
- Voogd, H. (1982) - "Multicriteria Evaluation for Urban and Regional Planning", Pion, London.
- Winkels, H. M. e Wascher, G. (1981) - "Outranking approaches an integreted survey and a bibliography, *Arbeitsberichte zur Oekonomathematik* n<sup>o</sup>8107, Ruhr-Universität, Bochum.
- Yu, P.L. (1973) - "Introduction to domination structures in multicriteria decision problems", em J.L.Cochrane e M.Zeleny (eds.), *Multiple Criteria Decision Making*, University of South-Carolina Press, Columbia, S.C. (249-261).
- Zadeh, L.A. (1970) - "Similarity relations and fuzzy ordering", *E.R.L. Memo M277*, Electrical Research Laboratory, University of California, Berkeley.
- Zadeh, L.A.; King-Sun Fu e Tanaka, K. (1975) - "Fuzzy sets and their applications to cognitive decision processes", *Papers from the U.S.-Japan Seminar*, University of California, Berkeley, Academic Press, New York.
- Zeleny, M. (1982) - *Multiple Criteria Decision Making*, McGraw-Hill, New York.
- Zimmermann, H.-J.; Zadeh, L.A. e Gaines, B.R. (1984) - *Fuzzy Sets and Decision Analysis*, *Studies in the Management Sciences*, vol.20, North-Holland, Amsterdam.

Zionts, S. (1985) - "Multiple criteria mathematical programming: an overview and several approaches", in G.Fandel e J.Spronk (eds.), **Multiple Criteria Decision Methods and Applications**, Springer, New York (85-128).

Zionts, S. e Wallenius, J. (1976) - "An interactive multiple objective linear programming method for solving the multiple criteria problem", **Management Science**, vol.22-6 (652-663).

# Associação Portuguesa para o Desenvolvimento da Investigação Operacional

## SCIENTIFIC JOURNAL

### "INVESTIGAÇÃO OPERACIONAL"

- Papers in Portuguese,  
English, French, Spanish
- EDITOR: J. P. Paixão

## SPECIALIZED COMMITTEES

### "OPTIMISATION"

- Headed by: J. Borges Gouveia

### "REGIONAL SYSTEMS"

- Headed by: P. Correia

### "INDUSTRIAL SYSTEMS"

- Headed by: J. Assis Lopes

### "AUTOMATIC CONTROL"

- Headed by: L. V. Tavares

### "TRANSPORT SYSTEMS"

- Headed by: J. M. Viegas

### "DECISION SUPPORT SYSTEMS"

- Headed by: J. Climaco

## SPECIALIZED MEETINGS

- 1st Advanced Seminar on Optimization  
9-11 Dec.  
Estoril, 1982
- Workshop on Planning, Control and Industrial Systems Management  
22-24 Feb.  
Sesimbra, 1984
- Seminar on Operations Research at IST  
12 Mars-24 July  
Lisbon, 1985
- Workshop on Purchase of Goods subject to Price Fluctuation  
11-12 April  
Óbidos, 1987
- 1st Seminar on Operations Research and Road Transport of Goods: Difficulties and Contributions  
27-28 Nov.  
Ferreira do Zêzere, 1988
- 1st Cycle on Management Techniques  
Distribution – Production Management –  
Regional and Urban Planning – Forecasting  
and Inventory Control  
Universidade dos Açores  
Açores, 1988

# Associação Portuguesa para o Desenvolvimento da Investigação Operacional

## NATIONAL MEETINGS

### BIENNIAL NATIONAL CONFERENCE

IO/82 – 1<sup>st</sup> National Conference of APDIO

22-24 March

Lisbon, 1982

Organized by Luís Valadares Tavares

IO/84 – 2<sup>nd</sup> National Conference of APDIO

16-18 April

Oporto, 1984

Organized by Rui Campos Guimarães

IO/87 – 3<sup>rd</sup> National Conference of APDIO

11-14 Oct.

Coimbra, 1987

Organized by Mário S. Rosa

## INTERNATIONAL CONFERENCES

### XXI European Meeting On Multicriteria Decision Support Systems

– 28-29 March

Lisbon, 1985

Organized by J. Clímaco

### PMS – Project Management and Scheduling, 11-13 July,

1988, Lisbon

Organized by L. Valadares Tavares

### OR MODELS ON MICROCOMPUTERS

– 25-27 Sept.

Lisbon, 1985

Organized by J. Dias Coelho

### MCDA SUMMER SCHOOL

– The Third International Summer  
School – 16-30 July,

1988, Lisbon

Organized by C. Bana e Costa

### IFAC CONFERENCE ON SYSTEMS ANALYSIS APPLIED TO WATER AND RELATED LAND RESOURCES

– 2-4 Oct.

Lisbon, 1985

Organized by L. Valadares Tavares and  
J. Evaristo da Silva

### UDMS 89

– 13th Urban Data Management  
Symposium – 29 May-2 June,

1989, Lisbon

Organized by A. Morais Arnaud

### II Meeting Portuguese-Spanish on Operations Research

– 15 Sept.

Lisbon, 1986

Organized by L. Valadares Tavares

### SIXTH EURO SUMMER INSTITUTE

– Funchal-Madeira

May/June 1989

Coordinator: J. M. Paixão

### EURO VIII – EIGHTH EUROPEAN CONFE- RENCE ON OPERATIONAL RESEARCH

– 16-19 Sept.

Lisbon, 1986

Organized by Isabel Hall Themido

### NATO ASI

– Operations Research and Management in  
Fishing

Póvoa de Varzim

March/April 1990

Organized by Bernardo Calafate Vascon-  
celos and

Guimarães Rodrigues

## ÍNDICE

*Isabel M<sup>a</sup> Hall Themido*

Modelos de Investigação Operacional para apoiar a tomada de decisão  
em Supermercados..... 3

*Paula Garcia Carranca & João de Quinhones Levy*

Aplicação da programação dinâmica a optimização de sistemas de tratamento..... 17

*J. Romão Eusébio, Lélío Amado, Lucinda Fragoso & J. Paixão*

Desenvolvimento e implementação de um sistema automático de escalas  
de autocarros e tripulações na Rodoviária Nacional..... 35

*Ana Maria Paias*

Sistema interactivo para geração dos horários de condutores de viaturas..... 47

*Maria da Conceição da Fonseca*

Implementação de uma heurística para determinação das rotações de serviços  
do pessoal tripulante..... 55

*Jorge Pinho de Sousa*

Single Machine Scheduling: Minimizing the weighted number of late jobs..... 65

*J. J. Júdice & A. M. Faustino*

The solution of the linear bilevel programming problem by using  
the linear complementarity problem..... 77

*João César das Neves*

Aplicação de um modelo de simulação à transferência de uma biblioteca..... 97

*J. Rodrigues Dias*

Validade de uma solução aproximada para o período de inspecção de sistemas  
com diferentes tipos de taxas de risco..... 107

*Carlos Bana e Costa*

Introdução geral às abordagens multicritério de apoio à tomada de decisão..... 117



Associação Portuguesa para o Desenvolvimento  
da Investigação Operacional.

CÉSUR - Instituto Superior Técnico - Avenida Rovisco Pais  
1000 Lisboa - Telef. 80 74 55