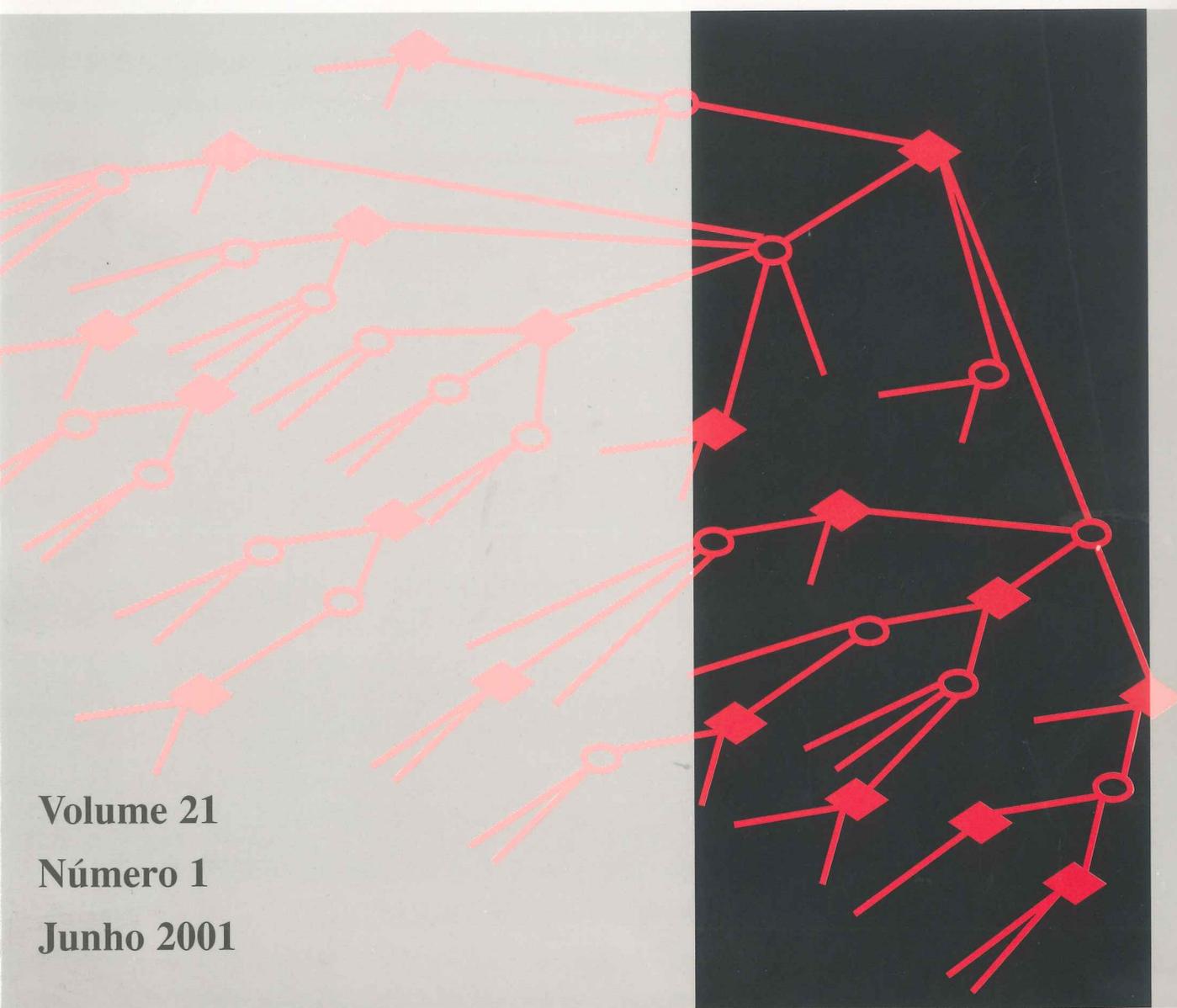


# *Investigação Operacional*



**Volume 21**

**Número 1**

**Junho 2001**

# INVESTIGAÇÃO OPERACIONAL

Propriedade:

APDIO — Associação Portuguesa de 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.*

Patrocinadores

**FCT**

Fundação para a Ciência e a Tecnologia  
MINISTÉRIO DA CIÊNCIA E DA TECNOLOGIA

**Fundação Calouste Gulbenkian**

ISSN n° 0874-5161

Dep. Legal n° 130 761 / 98

Execução Gráfica: J. F. Macedo - Astrografe

700 Ex.

2001/06

# INVESTIGAÇÃO OPERACIONAL

Volume 21 - nº 1 - Junho 2001

Publicação Semestral

Editor Principal: Joaquim J. Júdice  
Universidade de Coimbra

## Comissão Editorial

M. Teresa Almeida Inst. Sup. Economia e Gestão	Edite Fernandes Univ. do Minho	José F. Oliveira Univ. do Porto
C. Henggeler Antunes Univ. de Coimbra	J. Soeiro Ferreira Univ. do Porto	Rui Oliveira Inst. Superior Técnico
Jaime Barceló Univ. de Barcelona	J. Fernando Gonçalves Univ. do Porto	J. Pinho Paixão Univ. de Lisboa
C. Bana e Costa Inst. Superior Técnico	Luís Gouveia Univ. de Lisboa	M. Vaz Pato Inst. Sup. Economia e Gestão
M. Eugénia Captivo Univ. de Lisboa	Rui C. Guimarães Univ. do Porto	A. Guimarães Rodrigues Univ. do Minho
Domingos M. Cardoso Univ. de Aveiro	J. Assis Lopes Inst. Superior Técnico	António J. L. Rodrigues Univ. de Lisboa
João Clímaco Univ. de Coimbra	Carlos Luz Inst. Polit. Setúbal	J. Pinho de Sousa Univ. do Porto
J. Dias Coelho Univ. Nova de Lisboa	Virgílio P. Machado Univ. Nova de Lisboa	Reinaldo Sousa Univ. Católica, Rio Janeiro
João P. Costa Univ. de Coimbra	Manuel Matos Univ. do Porto	L. Valadares Tavares Inst. Superior Técnico
Ruy Costa Univ. Nova de Lisboa	N. Maculan Univ. Fed., Rio Janeiro	Isabel H. Themido Inst. Superior Técnico
J. Rodrigues Dias Univ. de Évora		B. Calafate Vasconcelos Univ. do Porto
Laureano Escudero IBM, Espanha		Luís N. Vicente Univ. de Coimbra

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

Esta Revista é distribuída gratuitamente aos sócios da APDIO. As informações sobre inscrições na Associação, assim como a correspondência para a Revista devem ser enviadas para a sede da APDIO - Associação Portuguesa para o Desenvolvimento da Investigação Operacional - CESUR, Instituto Superior Técnico, Av. Rovisco Pais, 1049 - 001 Lisboa.

Este Volume foi subsidiado por:

**Fundação para a Ciência e Tecnologia (FCT)**

**Fundação Calouste Gulbenkian**

Para efeitos de dactilografia e composição, foram utilizados equipamentos gentilmente postos à disposição pelo Centro de Investigação Operacional (DEIO - FCUL).

Assinatura: 5.000\$00

# UM MODELO MULTI-OBJECTIVO PARA PLANEAMENTO ENERGÉTICO BASEADO NA ANÁLISE INPUT-OUTPUT - APLICAÇÃO AO CASO PORTUGUÊS

**Carla Oliveira**

INESC-Coimbra  
Rua Antero de Quental, 199  
3000-033 Coimbra  
Portugal

**Carlos Henggeler Antunes**

Dept. Engenharia Electrotécnica-Univ. Coimbra  
Polo II, 3000-030 Coimbra  
INESC-Coimbra  
Rua Antero de Quental, 199  
3000-033 Coimbra (cantunes@inescc.pt)  
Portugal

## Abstract

A macro-level energy model and its application to energy planning in Portugal is presented. The study consists in modeling the energy system and its interactions with the whole economy by developing an input-output table, in which the energy sectors are disaggregated. A multiple objective linear programming model is then developed by using the coefficients in the input-output table. The multiple objective linear programming model is aimed at assisting planners and decision makers to study, in an interactive way, alternative policies.

## Resumo

No presente trabalho é desenvolvido um modelo multiobjectivo para planeamento energético baseado na análise *input-output* tendo como referência a situação portuguesa. O estudo consiste na modelação das interações do sistema energético com toda a economia, através da construção de uma matriz *input-output* onde os sectores energéticos são desagregados. A partir dos coeficientes da matriz *input-output* é desenvolvido um modelo de programação linear multiobjectivo. A finalidade do modelo proposto consiste em apoiar os decisores no estudo do impacto macroeconómico resultante da adopção de políticas alternativas.

## Keywords

Energy system, input-output, multiobjective linear programming

## 1. Introdução

Portugal caracteriza-se por uma elevada dependência energética do exterior, visto que, com excepção da energia hidroeléctrica, da biomassa florestal, de alguns resíduos industriais e de

outras fontes de energia renovável, o país é totalmente dependente da importação de combustíveis fósseis.

A formalização *input-output* é uma importante ferramenta de planeamento que permite ter em consideração todas as interações dos sistemas económico e energético, possibilitando determinar a quantidade de combustíveis fósseis necessária para os sectores produtores (nos consumos intermédios) ou, de modo directo, para procura final.

A área de aplicação específica da análise económica *input-output* ao sistema energético centra a sua atenção nos requisitos de energia primária necessários para a produção e consumo numa economia, permitindo avaliar a energia necessária para produzir um bem ou serviço. A utilização de combustíveis fósseis é associada ao nível de actividade de cada sector. Esta estrutura analítica é então expandida, de modo a permitir determinar as emissões de poluentes atmosféricos resultantes da combustão de combustíveis fósseis. Depois de determinados os requisitos de energia primária necessários à actividade económica, o conteúdo de carbono de um combustível e a respectiva produção de dióxido de carbono são ligados ao seu valor calorífico. Utilizando os coeficientes que permitem determinar a quantidade produzida de dióxido de carbono por unidade consumida de combustível, podem ser determinadas as emissões totais de cada sector e da economia como um todo.

Actualmente, nas sociedades modernas tecnologicamente desenvolvidas, as decisões estratégicas são tomadas num ambiente crescentemente complexo e "turbulento", caracterizado por evoluções e mutações constantes da tecnologia, da estrutura dos mercados e das necessidades sociais. A realidade é essencialmente caracterizada por objectivos múltiplos, conflituosos e incomensuráveis. Por este motivo, os modelos matemáticos de apoio à decisão tornam-se mais representativos da realidade se foram tidos em conta vários aspectos distintos de avaliação de um determinado problema. Assim, preocupações ambientais, económicas e sociais, por exemplo, devem ser consideradas de forma explícita e não agregadas num único indicador de carácter económico. Os modelos multiobjectivo permitem captar a diversidade de aspectos de avaliação, geralmente conflituosos e não comensuráveis, onde o decisor se depara com a necessidade de procurar compromissos entre os múltiplos objectivos, permitindo racionalizar a comparação entre diferentes soluções alternativas, uma vez que não existe uma solução admissível que conduza simultaneamente ao melhor valor para todos os aspectos de avaliação (operacionalizados através de funções objectivo). Num contexto multiobjectivo, o conceito de solução óptima, considerado nos problemas com apenas um objectivo, dá lugar ao conceito de soluções não dominadas (soluções admissíveis para as quais não é possível melhorar uma função objectivo sem sacrificar pelo menos uma das outras funções objectivo) (Steuer, 1986; Clímaco et al., 1996).

A abordagem multiobjectivo permite ao decisor a comparação de diferentes planos potenciais alternativos, conduzindo-o a uma melhor percepção dos diferentes aspectos conflituosos de avaliação. A estrutura de preferências do decisor assume particular relevância,

permitindo avaliar e seleccionar um plano considerado satisfatório do conjunto de soluções não dominadas.

No modelo desenvolvido consideram-se objectivos consistentes com o crescimento económico, o bem estar social, o nível de emprego, a conservação de energia e a preservação do meio ambiente. A metodologia utilizada para quantificar as emissões potenciais de CO<sub>2</sub> baseia-se nos princípios da combustão e composição dos combustíveis. Uma vez que há uma ligação entre o conteúdo de carbono e o valor energético do combustível, as emissões totais são determinadas multiplicando a quantidade de combustível sujeita a combustão (proporcional ao nível de actividade) por factores de emissão que se exprimem na quantidade de poluente produzida por unidade de energia consumida. O modelo de programação linear multiobjectivo, descrito na secção 2, contempla seis funções objectivo, cento e quarenta variáveis e duzentas e três restrições repartidas em oito grupos. As funções objectivo quantificam o consumo privado, o emprego, as importações de energia, o PIB, as emissões de CO<sub>2</sub> e a autoprodução de electricidade. Na secção 3 são apresentados alguns resultados ilustrativos, obtidos com o método interactivo STEM.

## **2. Um Modelo Multiobjectivo para Planeamento Energético Baseado na Análise *input-output***

A abordagem clássica de Leontief (Leontief, 1958) baseia-se na construção de um quadro *input-output* que representa os fluxos económicos e se estrutura de modo a fornecer uma apresentação sistemática e concisa de todas as actividades de um país ou região. Esta forma de modelação da economia tem sido utilizada como instrumento valioso na contabilidade nacional, bem como num vasto campo de problemas, nomeadamente de âmbito macroeconómico. A maior parte dos países industrializados dispõe de quadros *input-output* que servem de complemento às suas estatísticas de rendimento nacional e de imprescindível ajuda à realização de planos de desenvolvimento macroeconómico. A análise *input-output* apresenta, assim, grande interesse e utilidade no apoio à decisão em problemas de planeamento económico, fornecendo uma visão detalhada dos agregados macroeconómicos e dos fluxos monetários.

A política macroeconómica é, também, na sua essência, um problema de afectação de recursos, estando relacionada com a distribuição de recursos escassos por várias actividades económicas, de modo a atingir objectivos de natureza distinta, não só de ordem económica mas também de ordem ambiental e social. Neste contexto, a programação linear é uma importante ferramenta para a análise macroeconómica, permitindo auxiliar na escolha da melhor afectação de recursos, em problemas e modelos de política macroeconómica.

A matriz *input-output* utilizada para definir o modelo (Oliveira, 2000) desagrega todas as componentes energéticas nele contempladas, permitindo a distinção entre formas de energia primária e secundária. Nesta matriz, os fluxos de energia são considerados em unidades físicas e os fluxos dos sectores não energéticos são valorizados em unidades monetárias (fig. 1).

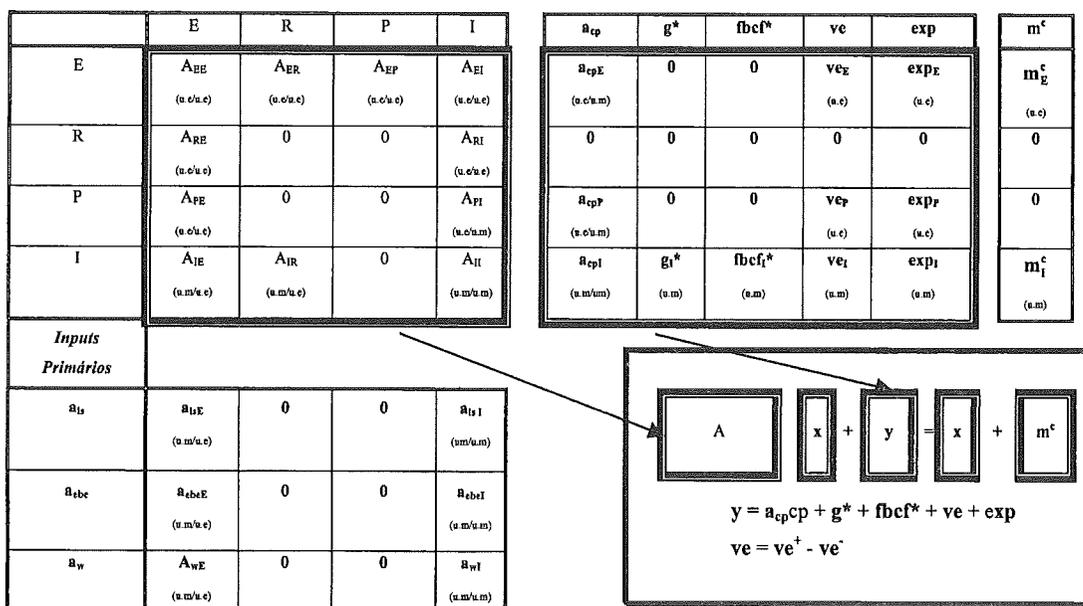


Fig. 1 - Representação esquemática da matriz de coeficientes técnicos utilizada no modelo

A notação utilizada na fig. 1 é a seguinte<sup>1</sup>:

- u.e designa unidades energéticas.
- u.m designa unidades monetárias.
- E designa os sectores energéticos (carvão, petróleo, electricidade, gás de cidade e autoprodução).
- R designa os subprodutos utilizados na autoprodução (gases incondensáveis, hidrogénio, licores sulfíticos, outros resíduos, alcatrão, gás de alto forno, gás de coque e biogás).
- P designa os produtos petrolíferos distribuídos (petróleo bruto, refugos e produtos intermédios, propileno, GPL, gasolinas, petróleos, jets, gasóleo/diesel, fuelóleo, nafta, lubrificantes, asfaltos, parafinas, solventes e coque de petróleo).
- I designa os sectores não energéticos (Indústria e Serviços).
- $A$  é a matriz de coeficientes técnicos de produção no ano base.
- $A_{EE}$  contém as vendas que os sectores energéticos fazem a si próprios, por unidade produzida de energia (por exemplo, consumo de carvão na produção de electricidade).
- $A_{ER}$  contém as contribuições do sector energético para os subprodutos, por unidade produzida de resíduos (por exemplo, quantidade de gases incondensáveis produzidos pelo sector de refinação).

<sup>1</sup> Na notação utilizada ao longo do texto considera-se que as letras minúsculas a negrito designam vectores coluna, as letras minúsculas designam escalares, as letras maiúsculas designam matrizes, as letras assinaladas com um asterisco designam variáveis exógenas e as letras minúsculas a negrito e sublinhadas designam matrizes diagonais, cujos elementos da diagonal principal são o vector correspondente à letra sublinhada.

- $A_{EP}$  contém a quantidade a distribuir dos vários produtos petrolíferos<sup>2</sup>, por unidade distribuída de produtos petrolíferos.
- $A_{EI}$  contém as vendas que o sector energético faz ao sector não energético, por unidade produzida de produtos não energéticos (por exemplo, venda de electricidade ao sector de serviços). Note-se que a venda de produtos petrolíferos não é aqui contabilizada.
- $A_{RE}$  contém a "venda" de subprodutos ao sector energético, por unidade produzida de energia (por exemplo, utilização de gases incondensáveis na autoprodução).
- $A_{RI}$  contém a "venda" de subprodutos aos sectores não energéticos, por unidade produzida de vapor (por exemplo, utilização de licores sulfúricos na produção de vapor).
- $A_{PE}$  contém as vendas de produtos petrolíferos ao sector energético, por unidade produzida de energia (por exemplo, venda de fuelóleo para a produção de electricidade).
- $A_{PI}$  contém as vendas de produtos refinados aos sectores não energéticos, por unidade produzida de produto não energético (por exemplo, venda de propileno à indústria química).
- $A_{IE}$  contém as vendas de produtos não energéticos aos sectores energéticos, por unidade produzida de energia (por exemplo, venda de papel ao sector de produção de energia eléctrica).
- $A_{IR}$  contém a contribuição dos sectores não energéticos para a produção de subprodutos, por unidade produzida de subprodutos (por exemplo, valor monetário da produção de gases incondensáveis do sector químico).
- $A_{II}$  contém as vendas de produtos não energéticos aos sectores não energéticos, por unidade não energética produzida (por exemplo, venda de produtos agrícolas à indústria química).
- $c$  é o vector do consumo privado do ano base, por sector de origem.
- $cp$  é o valor do consumo privado.
- $a_{cp}$  é o vector dos coeficientes de consumo privado, por sector de origem (rácio entre o consumo privado de bens ou serviços de um sector e o consumo privado total do ano base).
- $a_{cpE}$  contém o consumo privado de bens energéticos, por unidade de consumo privado total.
- $a_{cpP}$  contém o consumo privado de produtos petrolíferos, por unidade de consumo privado total.
- $a_{cpI}$  contém o consumo privado de produtos não energéticos, por unidade de consumo privado total.

---

<sup>2</sup> A quantidade a distribuir de cada produto de petróleo é composta por uma quantidade produzida internamente e por uma quantidade importada.

- $\mathbf{g}^*$  é o vector do consumo colectivo.
- $\mathbf{g}_I^*$  contém o consumo colectivo de serviços não mercantis (da administração pública, de educação e investigação, de saúde, etc.).
- $\mathbf{fbcf}^*$  é o vector do investimento em formação bruta de capital fixo (FBCF), por sector de origem.
- $\mathbf{fbcf}_I^*$  contém o investimento em FBCF, por sector não energético.
- $\mathbf{ve}$  é o vector do investimento em variação de existências, por sector de origem.
- $\mathbf{ve}_E$  contém o investimento em variação de existências, por sector energético.
- $\mathbf{ve}_P$  contém o investimento em variação de existências, por sector petrolífero<sup>3</sup>.
- $\mathbf{ve}_I$  contém o investimento em variação de existências, por sector não energético.
- $\mathbf{exp}$  é o vector das exportações, por sector de origem.
- $\mathbf{exp}_E$  contém as exportações, por sector energético.
- $\mathbf{exp}_P$  contém as exportações, por sector petrolífero.
- $\mathbf{exp}_I$  contém as exportações, por sector não energético.
- $\mathbf{m}^c$  é o vector das importações competitivas<sup>4</sup>, por sector de origem.
- $\mathbf{m}_E^c$  contém as importações competitivas, por sector energético<sup>5</sup>.
- $\mathbf{m}_I^c$  contém as importações competitivas, por sector não energético.
- $\mathbf{x}$  é o vector da produção total, por sector de origem.
- $\mathbf{y}$  é o vector da procura final, por sector de origem.

No modelo consideram-se 21 sectores económicos e 23 sectores artificiais (sectores distribuidores<sup>6</sup>), cuja informação é organizada da seguinte forma: uma matriz (44 x 44) que representa os fluxos inter e intra-sectoriais; 7 vectores coluna, 6 dos quais apresentam os valores referentes à procura final, por sector de origem e 1 com os valores das importações competitivas; 3 vectores linha para a distribuição dos *inputs* primários. Os sectores são classificados do seguinte modo: sectores energéticos, produtos petrolíferos, sub-produtos e sectores industriais (no sentido lato). O *output* total de cada sector é uma variável de decisão do modelo. A matriz de coeficientes técnicos obtém-se a partir do quadro de transacções *input-output* de um ano base (1995, por ser o ano mais recente para o qual se dispunha de informação

<sup>3</sup> Não se trata propriamente de um sector produtor, mas de um sector que apenas faz a distribuição dos diversos produtos petrolíferos pelas actividades económicas.

<sup>4</sup> Consideram-se importações competitivas ou concorrenciais as que correspondem a produtos com substituto próximo doméstico e importações não competitivas as que correspondem a produtos sem contrapartida doméstica. As importações não competitivas não são consideradas na matriz de coeficientes técnicos de produção, visto que as estatísticas nacionais optaram, por convenção, em agregar os bens (ou serviços) importados, provenientes de determinados sectores, aos seus congéneres, mesmo que não haja produção no país.

<sup>5</sup> As importações totais do sector petrolífero correspondem à soma das importações dos diversos produtos petrolíferos.

<sup>6</sup> O sector de refinação de petróleo apresenta vários *outputs*. Os *inputs* do sector de refinação têm uma estrutura de *inputs* comum, não sendo possível fazer uma afectação de custos separada para cada um dos produtos. Assim, a sua produção entra num sector artificial que não representa uma produção real, mas que permite apenas efectuar uma distribuição pelos sectores consumidores. Por outro lado, verifica-se que o consumo e o valor de produção dos subprodutos não aparecem contabilizados na matriz de transacções nacional. No entanto, no sector de autoprodução (DGE, 1995) é-lhes atribuído um custo. Deste modo, atribui-se um valor à produção destes subprodutos que acresce ao valor de produção do sector produtor e criam-se sectores artificiais para efectuar a sua distribuição.

estatística usada para obter os coeficientes do modelo). É nesta matriz que se encontram as relações existentes entre os sectores da economia, que se traduzem em restrições do modelo de programação linear multiobjectivo.

## 2.1 Restrições do modelo

### *Restrições de Coerência*

As utilizações de determinada actividade para consumo intermédio e procura final de bens (ou serviços) não poderão exceder o total disponível proveniente da produção nacional e das importações competitivas desse mesmo bem (ou serviço):

$$Ax + a_{cp}cp + g^* + fbcf^* + ve^+ - ve^- + exp \leq x + m^c \quad (1)$$

### *Balança de Pagamentos*

A nova apresentação estatística da Balança de Pagamentos (bp) é elaborada, exclusivamente, numa base de transacções e comporta três componentes principais (Banco de Portugal, 1999): a Balança Corrente, A Balança de Capital e a Balança Financeira. A Balança Corrente compreende as transacções, entre residentes e não residentes, associadas ao comércio internacional de mercadorias e serviços e aos rendimentos de trabalho e investimento. Nesta balança incluem-se ainda os valores correspondentes a mudanças de propriedade sem contrapartida (transferências unilaterais), devidas a operações de natureza corrente. Esta balança desagrega-se em mercadorias, serviços, rendimentos e transferências correntes. A Balança de Capital compreende as transferências de capital e a aquisição e/ou cedência de activos não produzidos não financeiros (abrange transacções sobre activos intangíveis). As transferências de capital correspondem a mudanças de propriedade sem contrapartida, relativas ao aumento dos activos do país receptor ou à diminuição dos seus passivos. A Balança Financeira compreende as transacções que envolvam a mudança de titularidade, entre residentes e não residentes, de activos e passivos financeiros e outras variações nos activos e passivos financeiros da economia. Por categorias funcionais, a balança financeira apresenta a seguinte desagregação: investimento directo, investimento de carteira, outro investimento, derivados financeiros e activos de reserva.

O nível das taxas de juro determina os movimentos de capitais de curto prazo. As alterações nestes fluxos podem ser observadas na balança financeira. A taxa de juro é uma variável exógena para os decisores nacionais de política económica (pertencentes à zona do euro). Deste modo, considera-se exógeno o saldo da balança financeira. De facto, considera-se apenas como endógena a parte das exportações e importações de bens e serviços:

$$bp - pe' exp + pm'm^c = bp^* \quad (2)$$

onde  $pe$  é o vector dos preços médios dos produtos energéticos exportados (por  $tep^7$ ), com dimensões convenientes (as unidades físicas em que estão expressos os combustíveis são

<sup>7</sup> Tonelada equivalente de petróleo (representa a energia contida, em média, numa tonelada de petróleo consumida no país).

convertidas em unidades monetárias); de modo similar,  $\mathbf{pm}$  é o vector dos preços médios dos produtos energéticos importados (por tep), com dimensões convenientes;  $\mathbf{bp}^*$  é a componente exógena da Balança de Pagamentos

#### Valor Acrescentado Bruto (vab)

Atendendo a que o Valor Acrescentado Bruto é um importante indicador para o planeamento macro-económico, permitindo quantificar os meios disponíveis gerados no país, pode escrever-se uma restrição com carácter definidor:

$$\mathbf{vab} = \mathbf{a}_w' \mathbf{x} + \mathbf{a}_{ebe}' \mathbf{x} + \mathbf{a}_{is}' \mathbf{x} \quad (3)$$

onde  $\mathbf{a}_w$ ,  $\mathbf{a}_{ebe}$  e  $\mathbf{a}_{is}$  são os vectores dos coeficientes de remunerações pagas aos trabalhadores, do excedente bruto de exploração gerado por cada sector de actividade e de impostos indirectos líquidos de subsídios, respectivamente.

#### Produto Interno Bruto (pib)

Para avaliar o crescimento de um país a nível económico, um dos indicadores mais utilizados é o PIB. O PIB pode ser determinado segundo três ópticas (INE, 1995): a óptica da despesa<sup>8</sup>, a óptica do produto<sup>9</sup> e a óptica do rendimento<sup>10</sup>. Consideram-se as duas primeiras ópticas de definição (as ópticas do produto e do rendimento são semelhantes). Deste modo,

$$\mathbf{pib} = \mathbf{cp} + \mathbf{i}' \mathbf{fbcf}^* + \mathbf{p}' (\mathbf{ve}^+ - \mathbf{ve}^-) + \mathbf{pe}' \mathbf{exp} - \mathbf{pm}' \mathbf{m}^c + \mathbf{i}' \mathbf{g}^* \quad (4)$$

onde  $\mathbf{p}$ , de forma análoga a  $\mathbf{pm}$  e a  $\mathbf{pe}$ , é o vector dos preços médios das variações de existências de produtos petrolíferos (por tep) e  $\mathbf{i}$  é um vector de 1's com dimensão adequada; e

$$\mathbf{pib} = \mathbf{vab} + \mathbf{a}_{iim}' \mathbf{m}^c + \mathbf{a}_{iva} \mathbf{vab} \quad (5)$$

onde  $\mathbf{a}_{iim}$  é o vector das taxas médias de imposto sobre as importações e  $\mathbf{a}_{iva}$  é a taxa média de imposto sobre o valor acrescentado.

A coerência interna do modelo não garante que se verifique a igualdade das três ópticas. Deste modo, consideraram-se no modelo ambas as definições.

#### Restrições de Capacidade Produtiva

A produção total de cada sector deverá ser limitada superiormente pela correspondente capacidade produtiva, não devendo também ser inferior a determinado nível. Desta forma

$$\mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \quad (6)$$

onde  $\mathbf{lb}$  e  $\mathbf{ub}$  são os vectores dos limites inferiores e superiores de capacidade produtiva, respectivamente, para cada sector.

#### Restrições Impostas às Exportações e Importações

Os fenómenos de sobre-especialização são muito frequentes em modelos onde as importações e as exportações são endógenas (e desligadas dos coeficientes técnicos do

<sup>8</sup> PIB = Consumo Privado + Consumo Colectivo + Investimento Total + Saldo da Balança Comercial.

<sup>9</sup> PIB = Valor Acrescentado Bruto + Imposto Sobre o Valor Acrescentado + Impostos Sobre as Importações.

<sup>10</sup> PIB = Remunerações + Impostos Sobre a Produção - Subsídios de Exploração + Excedente Bruto de Exploração + Imposto Sobre o Valor Acrescentado + Impostos Sobre as Importações.

modelo). Deste modo, o modelo será mais realista se tiver em consideração limites superiores para as exportações e para as importações competitivas de alguns sectores.

$$\mathbf{exp} \leq \mathbf{ub}_{\mathbf{exp}} \quad (7)$$

$$\mathbf{m}^c \leq \mathbf{ub}_m^c \quad (8)$$

onde  $\mathbf{ub}_{\mathbf{exp}}$  e  $\mathbf{ub}_m^c$  são os vectores dos limites superiores das exportações e das importações competitivas, respectivamente, por sector de actividade.

#### *Défice Público ou Saldo Global do Sector Público Administrativo (SPA) de sinal negativo*

Atendendo aos critérios de convergência nominais da UE, relativamente à adesão do país à moeda única, o défice público não deverá exceder 3% do PIB. O SPA obtém-se subtraindo as despesas às receitas da administração pública, excluindo os activos e passivos financeiros, as contas de ordem e os saldos de gerências anteriores. Deve ainda retirar-se das transferências do Estado, quando estas não são líquidas, a transferência do Estado para o Fundo da Regularização da Dívida Pública com o objectivo da amortização da dívida.

Considerando que os impostos indirectos à produção líquidos de subsídios, os impostos directos e as contribuições para a segurança social são as únicas rubricas endógenas do défice público, pode escrever-se:

$$-\left[ t_f \left[ \frac{1}{1-k_5} (\mathbf{a}_w' \mathbf{x} + k_4 \mathbf{a}_{ebe}' \mathbf{x}) \right] + t_e \mathbf{a}_{ebe}' \mathbf{x} + c \mathbf{a}_w' \mathbf{x} + \mathbf{a}_{is}' \mathbf{x} + \alpha_1 \mathbf{a}_{iim}' \mathbf{m}^c + \alpha_2 \mathbf{a}_{iva}' \mathbf{vab} \right] + dp^* \leq 0.03 \text{pib} \quad (9)$$

onde  $t_f$  é uma taxa média de imposto sobre o rendimento e propriedade das famílias,  $k_5$  é uma constante ( $[\text{rendimento bruto} - (\text{salários} + \text{excedente bruto de exploração}) / \text{rendimento bruto}$ ),  $k_4$  é a percentagem de excedente bruto de exploração obtida pelas famílias (as componentes mais importantes do rendimento bruto são os salários e o excedente bruto de exploração),  $t_e$  é a taxa média de imposto sobre o rendimento das empresas,  $c$  é uma taxa de média de contribuições para a segurança social,  $\alpha_1$  é a proporção de impostos sobre as importações que cabe ao Estado,  $\alpha_2$  é a proporção de IVA que cabe ao Estado e  $dp^*$  é a componente exógena do défice.

#### *Capacidade de Armazenagem e Existências de Segurança*

$$\mathbf{ve}_p^+ - \mathbf{ve}_p^- \leq \mathbf{ca}^* - \mathbf{e}^{0*} \quad (10)$$

$$\mathbf{ve}_p^+ - \mathbf{ve}_p^- \geq \mathbf{es}^* - \mathbf{e}^{0*} \quad (11)$$

com  $\mathbf{ve}_p^+ \geq \mathbf{0}$  e  $\mathbf{ve}_p^- \geq \mathbf{0}$ ,

onde  $\mathbf{e}^{0*}$  é o vector de existência iniciais,  $\mathbf{es}^*$  é o vector de existências de segurança,  $\mathbf{ca}^*$  é a capacidade de armazenagem existente e  $\mathbf{ve}_p^+$  e  $\mathbf{ve}_p^-$  são os vectores das variações positivas e negativas das existências de produtos petrolíferos.

As restrições (10) obrigam a que variações positivas das existências não excedam a capacidade de armazenagem e as restrições (11) garantem que as reduções de existências não abranjam as existências de segurança.

## 2.2 Funções Objectivo

### O Consumo Privado

O consumo privado do país (cp) é dado pelo consumo total dos residentes (cr) com exclusão do consumo dos residentes fora do país (crf - despesas de turistas portugueses no estrangeiro) e adicionando o consumo de não residentes no país (cnr - despesas de turistas estrangeiros em Portugal). O consumo dos residentes considera-se função do rendimento disponível bruto. Deste modo,

$$cp = \frac{(1-k_2)}{(1-k_1)} \left[ \frac{k_3}{1-k_5} (1-t_f) (a_w 'x + k_4 a_{ebe} 'x) - k_3 c a_w 'x \right] \quad (12)$$

onde  $k_1$  é uma constante (cnr/cp),  $k_2$  é uma constante (crf/cr) e  $k_3$  é a propensão média ao consumo (cr/rendimento bruto disponível), que se assume constante durante o período de planeamento.

A função objectivo considerada como medida de bem estar dos cidadãos é

$$\max cp.$$

### Emprego

Outra medida de bem estar social é o nível de emprego que deverá ser maximizado:

$$\max a_{emp} 'x$$

onde o vector  $a_{emp}$  contém o número de empregados por unidade produzida em cada sector de actividade.

### Importações de Energia

As importações de energia devem ser minimizadas, dada a dependência energética do país:

$$\min i' m_E^c$$

onde  $i$  é um vector de 1's de dimensão conveniente e  $m_E^c$  é o vector das importações competitivas de energia (de acordo com a organização das estatísticas nacionais).

### PIB

Com esta função objectivo pretende-se maximizar o desempenho da economia nacional, medido através do PIB:

$$\max pib.$$

### Emissões de CO<sub>2</sub>

A minimização do impacte da utilização dos recursos energéticos na poluição atmosférica, conduz a considerar a seguinte função objectivo:

$$\min c_t$$

onde  $c_t$  designa as emissões totais de CO<sub>2</sub>.

Para a obtenção de dados mais representativos da realidade foi seguida a metodologia *top-down* do IPCC. O processo de estimação de emissões de CO<sub>2</sub> pode repartir-se em 6 fases (IPCC, 1996): 1) determinar o consumo aparente de combustíveis, por tipo de combustível; 2) converter, se necessário, os valores em TJ; 3) transformar o consumo aparente de cada combustível em conteúdo de carbono através da multiplicação do consumo aparente de cada

combustível pelo correspondente factor de emissão de carbono; 4) determinar a quantidade de carbono de cada combustível destinado a fins não energéticos e dedução desta quantidade do carbono contido no consumo aparente, de modo a determinar o conteúdo real de carbono; 5) corrigir os valores para o cálculo da combustão incompleta do combustível, de modo a determinar a quantidade de carbono realmente oxidado na combustão; 6) converter a quantidade de carbono oxidado em emissões de CO<sub>2</sub>.

Deste modo, as emissões de CO<sub>2</sub> podem definir-se como:

$$c_t = \mathbf{fo}' (\mathbf{fec} \mathbf{fc} A_E \times 10^{-3} - \mathbf{fca} \mathbf{fec} N_E \times 10^{-3}) \frac{44}{12} + \mathbf{fo}' (\mathbf{fec} \mathbf{fc} \mathbf{a}_{ecp} \mathbf{cp} \times 10^{-3} - \mathbf{fca} \mathbf{fec} \mathbf{na}_{ecp} \mathbf{cp}) \frac{44}{12}$$

onde  $A_E = \begin{bmatrix} A_{EE} & A_{ER} & A_{EP} & A_{EI} \\ A_{RE} & 0 & 0 & A_{RI} \\ A_{PE} & 0 & 0 & A_{PI} \end{bmatrix}$  é uma submatriz da matriz de coeficientes técnicos A

com os coeficientes de utilização de energia (teps/unidades de output);  $N_E$  é uma matriz de coeficientes de utilizações não energéticas de energia (TJ/unidade de output);  $\mathbf{fo}$  é o vector das fracções de carbono oxidado;  $\mathbf{fec}$  é uma matriz diagonal, cujos elementos da diagonal principal são os factores de emissão de carbono (tC/TJ);  $\mathbf{fc}$  é uma matriz diagonal cujos elementos da

diagonal principal são os factores de conversão de teps em TJ;  $\mathbf{a}_{ecp} = \begin{bmatrix} \mathbf{a}_{cpE} \\ 0 \\ \mathbf{a}_{cpP} \end{bmatrix}$  é o vector dos

coeficientes de consumo de energia no sector privado;  $\mathbf{na}_{ecp}$  é o vector dos coeficientes de utilização não energética de energia (TJ/unidade de consumo total);  $\frac{44}{12}$  é o rácio entre os pesos moleculares do CO<sub>2</sub> e do carbono.

### *Autoprodução de Electricidade*

O recurso a formas de energia alternativas na autoprodução, e conseqüente valorização e reciclagem dos resíduos, permite não só efectuar economias de energia, mas também minimizar as quantidades a eliminar dos resíduos. Assim, outra função objectivo considerada é a maximização da autoprodução de electricidade:

$$\max x_{aut.}$$

### **2.3 Algumas características e limitações do modelo**

O modelo apresentado possui limitações decorrentes do acesso à informação estatística, que tornam os dados menos representativos das condições actuais do país. Assim, até à data de elaboração deste estudo, a matriz de transacções mais recente era datada de 1995. O mesmo sucedia em relação ao balanço energético nacional detalhado, sendo o mais recente do mesmo ano.

As dificuldades a nível estatístico surgiram, também, na análise dos quadros de autoprodução (que não apresentavam o detalhe necessário para poder identificar as quantidades

de energia consumidas na co-geração, ou mesmo a quantidade de vapor recuperada e utilizada noutras actividades industriais), dado que os factores de conversão aconselhados nem sempre coincidiam com os utilizados na construção do balanço energético nacional. Por este motivo, surgem na matriz *input-output* alguns valores diferentes dos considerados no balanço energético nacional de 1995.

A hipótese de imutabilidade da utilização de combustíveis assumida nesta análise, resultante das características inerentes a este tipo de modelos estáticos, é pouco realista. Com o progresso tecnológico é natural que comecem a utilizar-se novas formas de energia que ofereçam rendimentos superiores, com preços mais atractivos e mais ecológicas. O aparecimento do gás natural canalizado, em finais de 1996, é disso exemplo; porém, este combustível não é contemplado no estudo, visto que o gás natural não era considerado na estrutura produtiva do ano base (1995), para o qual estavam disponíveis os dados mais recentes.

Relativamente à quantificação dos gases que provocam o efeito de estufa, o método utilizado no presente trabalho, para o cálculo das emissões de dióxido de carbono, permite apenas estimar de modo "grosseiro" as emissões deste tipo de gases. O valor determinado para as emissões destes gases seria mais aproximado dos valores realmente verificados se a informação estatística disponível sobre a utilização de combustíveis fosse mais detalhada. Poder-se-ia considerar, alternativamente, um método que dividisse o consumo de combustíveis com base no conhecimento de tecnologias suficientemente homogéneas para permitir a utilização de factores de emissão mais representativos. A única alteração a efectuar ao modelo *input-output* clássico consistiria em separar a procura de um determinado combustível por um dado sector da economia nas diversas tecnologias utilizadas na combustão.

No trabalho apresentado utiliza-se um único vector de coeficientes técnicos para cada sector, o que traduz a hipótese de não alteração tecnológica. Poder-se-ia também ter em conta a existência de processos alternativos de fabrico, considerando-se vectores alternativos de produção para cada sector de actividade. Porém, os dados necessários para alimentar um modelo deste tipo são escassos ou mesmo inexistentes.

### 3. Alguns Resultados Ilustrativos

A partir deste modelo multiobjectivo *input-output* para planeamento energético alimentado com dados estatísticos publicados pela Direcção Geral de Energia e pelo Instituto Nacional de Estatística, para o ano base de 1995, bem como estimativas para o ano 2000 foram obtidos alguns resultados ilustrativos (Oliveira, 2000).

O processo de pesquisa de soluções iniciou-se com o cálculo das soluções não dominadas que optimizam separadamente cada uma das funções objectivo. Esta informação é útil para obter uma visão de carácter global da gama de variação de cada função objectivo, bem como das características das soluções (em princípio bem distintas) que optimizam cada uma delas.

Os valores das funções objectivo para as soluções não dominadas que optimizam separadamente cada função objectivo estão organizados na tabela 1.

Os valores do consumo privado e do PIB estão expressos em milhões de escudos, o valor do emprego é dado em milhares de pessoas, os valores da importação de energia e da autoprodução estão quantificados em teps, e os valores da poluição estão expressos em gigagramas de dióxido de carbono.

Solução	Consumo (10 <sup>6</sup> esc.)	Emprego (10 <sup>3</sup> )	Imp. Energia (tep)	PIB (10 <sup>6</sup> esc.)	Poluição (Gg CO <sub>2</sub> )	Autoprodução (tep)
1	<b>12758731</b>	5466	22168566	19462057	62949	333314
2	12755082	<b>5472</b>	22163756	19458519	62506	323349
3	12357513	5278	<b>19320771</b>	18850937	58475	306016
4	12756361	5457	22165006	<b>19465134</b>	62331	326990
5	12355369	5275	20021180	18849107	<b>58402</b>	305730
6	12667011	5435	19517785	19322142	59276	<b>336406</b>

Tabela 1 - Tabela de óptimos individuais

As soluções não dominadas que optimizam cada função objectivo separadamente possuem as seguintes características principais:

- *Solução 1* (que maximiza o consumo privado): os valores obtidos para os níveis de importação de energia e de emissões de CO<sub>2</sub> são os mais elevados (relativamente a estas seis soluções); o *output* do sector de autoprodução aproxima-se do respectivo óptimo; os *outputs* da electricidade, carvão, petróleo, papel, outras indústrias transformadoras, água, porcelanas e vidro e de quase todos os sub-productos (excepto o biogás) são os mais elevados.
- *Solução 2* (que maximiza o emprego): todos os objectivos atingem valores próximos dos obtidos nas soluções 1 (max cp) e 4 (max pib); os *outputs* obtidos para a agricultura, porcelanas e vidro (sectores intensivos em mão de obra), petróleos (a produção agrícola é intensiva neste combustível), GPL (a produção de porcelanas e vidro é intensiva neste combustível) e propileno são os mais elevados (relativamente às seis soluções apresentadas).
- *Solução 3* (que minimiza as importações de energia): A Balança de Pagamentos atinge o seu valor mais baixo (a redução dos níveis de *output* conduz, em geral, a níveis mais elevados de importação); todos os objectivos atingem valores próximos dos obtidos na solução 5 (min c<sub>j</sub>); os *outputs* dos sectores agrícola, construção, serviços, transportes, electricidade, gás de cidade e quase todos os hidrocarbonetos (excepto o fuelóleo) atingem os valores mais baixos (para estas seis soluções). Os níveis de importação de petróleo bruto e de produtos intermédios são mais baixos na solução 5 do que na solução 3. Este facto resulta de um maior nível de produção de fuelóleo na solução 3, relativamente à solução 5 (que resulta num ligeiro aumento da autoprodução), conduzindo aos níveis de importação mais baixos de electricidade.

- *Solução 4* (que maximiza o PIB): A Balança de Pagamentos e o VAB atingem os valores mais elevados; os valores dos *outputs* alcançados para os serviços e a construção (refira-se a contribuição dos serviços para o VAB na ordem dos 60%, em 1995, e a forte componente do sector de construção e obras públicas nos gastos das administrações públicas), o gás de cidade (o sector de serviços é o mais intensivo na utilização desta forma de energia), gasóleo (para exportações) e indústria química (*output* absorvido principalmente em consumos intermédios) são os mais elevados (no âmbito destas seis soluções).
- *Solução 5* (que minimiza as emissões de CO<sub>2</sub>): o valor obtido para o VAB é o mais baixo (nestas seis soluções); os valores da autoprodução, PIB, consumo privado e emprego são os mais baixos. Os *outputs* alcançados na produção de produtos petrolíferos, água e vapor, electricidade (altamente dependente de hidrocarbonetos), minerais (inclui siderurgia), químicas, porcelanas e vidro, papel, alimentares, transportes, carvão e sub-produtos são os mais baixos. O nível de importações de electricidade é elevado, denotando o impacte negativo da sua produção nas emissões de CO<sub>2</sub>.
- *Solução 6* (que maximiza a autoprodução de electricidade): os valores do consumo privado, PIB e emprego são relativamente elevados (próximos das soluções 1, 2 e 4); o valor da importação de energia encontra-se próximo do respectivo óptimo e as emissões de CO<sub>2</sub> são inferiores quando comparadas com as das soluções 1, 2 e 4; os *outputs* são mais elevados no sectores agrícola (tendo como resultado a produção máxima de biogás) e minerais (sector responsável pela produção de gás de coque, alcatrão e gás de alto forno, sub-produtos utilizados na autoprodução). O *output* mais baixo do sector de produção de electricidade e um valor menor da importação desta forma de energia mostram a diminuição da dependência das importações de electricidade, quando a auto-produção é maximizada.

### 3.1 O método STEM

O método interactivo STEM foi seleccionado para apoio à decisão neste estudo de planeamento energético baseado num modelo multiobjectivo *input-output* por possuir um protocolo de interacção simples e pela facilidade de implementação através do recurso a uma "package" de programação linear (Benayoun et al., 1971; Steuer, 1986, cap. 13; Clímaco et al., 1996).

Em cada fase de cálculo é determinada a solução que minimiza uma distância ponderada de Tchebycheff à solução ideal. Em cada interacção do STEM o decisor especifica as quantidades que está disposto a sacrificar nas funções objectivo com valores que considera satisfatórios, de modo a melhorar as restantes. O problema a resolver em cada iteração reflecte as escolhas efectuadas pelo decisor em iterações precedentes, através da redução da região admissível.

Quando os valores das funções objectivo são considerados satisfatórios, o processo termina; caso contrário, o decisor deve especificar as funções que pretende relaxar e de que quantidade, de modo a melhorar as restantes funções objectivo. A região admissível é então reduzida com a introdução destas limitações nos valores das funções objectivo. Nas experiências efectuadas foi utilizada uma versão mais flexível do método STEM, em que é permitido relaxar mais do que uma função objectivo em cada iteração e a mesma função objectivo em várias iterações.

Para obtenção dos resultados ilustrativos seguidamente descritos supõe-se a existência de um decisor hipotético, que expressa as suas preferências face à informação que lhe é apresentada, conduzindo assim o processo de pesquisa de novas soluções.

Inicialmente são calculadas as soluções eficientes que optimizam separadamente cada uma das funções objectivo, sendo apresentada ao decisor a tabela 1.

Em seguida, resolve-se o problema escalar para determinar a solução que minimiza uma distância ponderada de Tchebycheff à solução ideal. A solução obtida na primeira iteração é

$$\mathbf{z}^{(1)} = (12528068, 5350, 19400616, 19113601, 58737, 306455).$$

Face a esta solução vamos considerar:

- que o decisor considera satisfatório o valor do PIB, admitindo prescindir de 1000000 milhões de escudos no seu valor, de modo a melhorar as restantes funções objectivo. A solução obtida  $\mathbf{z}^{(2)}$ , resultante do relaxamento do valor do PIB, não revela alterações. Neste problema qualquer relaxamento isolado no PIB a partir da 1ª iteração conduz sempre à mesma solução  $\mathbf{z}^{(1)}$ .
- Suponhamos que o decisor passa também a considerar o valor do consumo privado satisfatório, pelo que admite relaxar o seu valor em 600000 milhões de escudos. A solução resultante do relaxamento do consumo privado e do PIB é  $\mathbf{z}^{(3)} = (12439977, 5350, 19389387, 18975919, 58737, 306455)$ . Nesta solução há um "aproveitamento" do relaxamento dessas duas funções objectivo, verificando-se apenas uma melhoria nos níveis de importação de energia.
- Suponhamos que o decisor, para melhorar os restantes objectivos, passa a considerar como satisfatório o nível de emprego, relaxando o seu valor em 100 milhares de pessoas. A solução obtida na nova iteração é  $\mathbf{z}^{(4)} = (12439977, 5313, 19359699, 18977937, 58603, 306455)$ . Os resultados obtidos mostram que o relaxamento do PIB não foi "aproveitado", tendo mesmo o seu valor melhorado relativamente à solução anterior. No entanto, a deterioração dos valores do consumo e emprego permitiu uma melhoria dos valores da poluição e da importação de energia. Refira-se, por último, que um maior relaxamento do emprego não se traduz numa alteração da solução obtida nesta iteração.
- Supondo que o decisor considera esta última solução satisfatória, o processo termina.

Da análise dos valores obtidos nas soluções das quatro iterações do método STEM é possível retirar algumas indicações: a solução calculada na segunda iteração  $z^{(2)}$  possui valores para as funções objectivo iguais aos obtidos na primeira iteração. No entanto, verificam-se diferenças nas importações de produtos da indústria metalúrgica, que apresentam níveis superiores na solução  $z^{(1)}$ , e nas importações de produtos da indústria metalo-electro-mecânica, que apresentam valores superiores na solução  $z^{(2)}$ . Esta situação é compensada pelo investimento em variação de existências em ambas as soluções.

Na terceira iteração, onde se relaxam o PIB e o consumo privado em simultâneo, obtêm-se para a solução  $z^{(3)}$  diminuições dos níveis de importações das indústrias química, metalúrgica, metalo-electro-mecânica, carvão, petróleo bruto, produtos intermédios, GPL, naftas, asfaltos, parafinas e solventes. Por outro lado, regista-se um aumento na importação da electricidade e de papel. Em termos de produção sectorial, atingem-se os níveis mais baixos de produção (relativamente às quatro soluções calculadas com o método STEM) nos sectores de minerais, de outras indústrias, de construção, serviços, água e vapor de água, gás de cidade, de quase todos os subprodutos (excepto o hidrogénio que passa a apresentar uma produção positiva), dos petróleos e asfaltos (ambos com o mesmo valor na iteração seguinte) e os níveis mais elevados no sector agrícola. O investimento em variação de existências passa a apresentar valores mais altos nos seguintes sectores: agricultura, porcelana e vidro, cimento e papel. As exportações apresentam os valores mais baixos (relativamente às soluções obtidas com o STEM) para os minerais, gásóleo e lubrificantes, enquanto os valores mais altos se reflectem nos sectores das madeiras, electricidade e *jets*.

Na última iteração, onde são relaxados, em simultâneo, o consumo privado, o PIB e o nível de emprego, há a registar na solução  $z^{(4)}$ , relativamente à iteração anterior, o seguinte: os níveis de importações totais apresentam os valores mais elevados, situação que não se deve à importação de energia, que é inferior relativamente à solução anterior (as importações de electricidade e petróleo registam uma diminuição que compensa o ligeiro aumento das importações de carvão), mas às importações nos sectores químico, metalo-electro-mecânico e de transportes; em termos de produção sectorial verifica-se que a produção nos sectores agrícolas, de petróleo (devido a uma ligeira redução nos valores da produção de fuelóleo) e alimentar atinge os níveis mais baixos (relativamente às soluções obtidas nas quatro iterações do método STEM). O valor obtido para o VAB é ligeiramente superior ao obtido na iteração anterior, enquanto o saldo da Balança de Pagamentos apresenta o valor mais baixo (para estas quatro soluções).

Embora tivesse sido realizada uma vasta recolha de informação estatística (particularmente junto do INE e da DGE) que possibilitasse alimentar o modelo multiobjectivo com dados próximos da realidade, foi ainda necessário estimar vários parâmetros para os quais não foi possível obter valores. Neste contexto, os resultados ilustrativos, aqui brevemente descritos, devem ser considerados como um exercício (o mais realista possível) em torno das

potencialidades oferecidas por modelos e métodos interactivos de apoio à decisão multiobjectivo no sentido de ajudar os decisores na tomada de decisões mais fundamentadas.

#### 4. Conclusões

O sector energético tem grande importância no conjunto das actividades económicas, quer pelo seu impacte directo na procura final, quer pelo impacte no aparelho produtivo, com as consequências que daí derivam ao nível do emprego, do abastecimento interno, das relações comerciais com o exterior e, em geral, do bem estar social.

A realização deste trabalho procurou proporcionar uma perspectiva global sobre as questões relacionadas com o planeamento energético, tendo em conta, nomeadamente, os impactes ambientais resultantes da utilização das diversas fontes energéticas. O modelo de planeamento proposto é um modelo de programação linear multiobjectivo com base na metodologia *input-output*, particularmente adaptada ao estudo da problemática energética. Este modelo pretende ser um instrumento de apoio aos decisores na análise do impacte macroeconómico resultante da adopção de políticas alternativas. Para o efeito, foram devidamente tidas em conta as especificidades do sistema energético português.

No caso vertente, embora se tenha considerado um nível razoável de desagregação da actividade económica, as incertezas e omissões existentes na base de dados, que nunca será possível eliminar na totalidade, bem como as hipóteses simplificadoras admitidas na estimação de alguns valores para o ano 2000 relativamente à componente exógena do modelo, reforçam o facto de as soluções destes modelos não poderem ser encaradas como prescrições definitivas com carácter normativo, mas sim como referências para encontrar planos de acção bem fundamentados.

O modelo comporta seis funções objectivo, cento e quarenta variáveis e duzentas e três restrições, apoiando-se numa tabela *input-output* que considera a economia do país repartida em vinte e um sectores económicos e em vinte e três sectores energéticos artificiais.

O modelo permite ainda obter informação sobre outros indicadores macroeconómicos, com o detalhe permitido pela informação estatística disponível, como o VAB e o saldo da Balança de Pagamentos.

O modelo de planeamento energético proposto é suficientemente flexível para poder ser facilmente alterado, de modo a incorporar novos combustíveis que entretanto seja necessário considerar no sistema energético nacional, bem como algumas questões importantes que em seguida se indicam como sugestões para desenvolvimento futuro.

O primeiro passo consistirá em desagregar o sector de produção de energia eléctrica nas suas diversas tecnologias de produção: as produções hídrica (muito dependente dos níveis de precipitação), térmica-carvão, térmica-fuelóleo, térmica-gasóleo e térmica-gás natural.

Poderão ainda ser considerados processos alternativos de fabrico no sector de refinação de petróleo, devido à flexibilidade existente na selecção da proporção em que cada um dos produtos refinados surge.

Por fim, a mensuração do impacte ambiental resultante da utilização de combustíveis poderá incorporar, para além do dióxido de carbono, outros gases nefastos para o ambiente.

### Referências

- [1] Banco de Portugal, Nova apresentação das Estatísticas da Balança de Pagamentos (1995).
- [2] Benayoun, R., Montgolfier, J., Tergny, J., Larichev, O., Linear Programming with Multiple Objective Functions: Step Method (STEM), *Mathematical Programming* 1 (1971) 366-375.
- [3] Clfmaco, J., Antunes, C.H., Alves, M.J., Programação Linear Multiobjectivo-Métodos Interactivos, "Software" e Aplicações, Faculdade de Economia da Universidade de Coimbra (1996).
- [4] DGE, Auto-Produtores de Energia Eléctrica, Divisão de Estatística (1995).
- [5] INE, Contas Nacionais-Total do País (1995).
- [6] IPCC, Revised 1996 IPCC Guidelines for National Greenhouse Gas Inventories: Reference Manual, In <http://www.iea.org/ipcc/invs6.htm> (1996).
- [7] Leontief, W., *La estructura de la economia americana (1914-1939)*, Bosch, Barcelona (1958).
- [8] Oliveira, C., Um modelo multiobjectivo para planeamento energético baseado na análise input-output, Dissertação de Mestrado em Gestão da Informação nas Organizações, Faculdade de Economia da Universidade de Coimbra (2000).
- [9] Steuer, R., *Multiple Criteria Optimization: Theory, Computation and Application*, Wiley (1986).

# ANÁLISE MULTIOBJECTIVO NO PLANEAMENTO AGREGADO DA PRODUÇÃO

**Carlos Gomes da Silva**

Escola Sup. Tecnologia e Gestão de Leiria  
2401-951 Leiria

**José Figueira**

**João Lisboa**

Faculdade de Economia  
Universidade de Coimbra  
3004-512 Coimbra

## **Abstract**

This paper presents a mixed integer multiple objective linear programming model in the branch of the aggregate production planning applied to a Portuguese firm producing construction materials. The multiple objective model was developed with the following objectives in mind: (1) to maximise profit; (2) to minimise back orders; (3) to minimise workforce fluctuations. This model allows the inclusion of certain aspects related to the partial inflexibility of the workforce, legal restrictions on the workload, workforce size (workers to be hired and downsized), workers in training, limitation of production and inventory levels, and the quantity of products to be subcontracted. The purpose of this study is to define the level and type of workers, the amount of regular hours and overtime, the inventory level for each type of product, and the level of subcontracted products in order to meet the forecast demand, for a planning period of 12 months. Also presented is an interactive method, which provides support for the decision-maker in knowledge and exploitation of the possible solutions and in selection of the "best solution".

## **Resumo**

Este artigo apresenta um modelo de programação linear inteira mista multiobjectivo para o planeamento agregado da produção (PAP), aplicado a uma empresa portuguesa dedicada à produção de materiais de construção. O modelo multiobjectivo foi desenvolvido tendo conta os seguintes objectivos: (1) maximização do lucro; (2) minimização das encomendas com entrega atrasada; (3) minimização da flutuação do número de trabalhadores. Este modelo permite contemplar certos aspectos relacionados com a inflexibilidade parcial dos trabalhadores, restrições legais ao despedimento, número de trabalhadores contratados e despedidos, número de trabalhadores em formação, limitação da produção e das existências e subcontratação de produtos. O objectivo deste estudo é definir o nível e o tipo de trabalhadores, o número de horas normais e extraordinárias, o nível de existências de cada produto e o nível da subcontratação de cada um dos produtos de modo a satisfazer a procura prevista num período de planeamento de 12 meses. É ainda apresentado um método interactivo que permite apoiar o agente de decisão (AD) no conhecimento e exploração das soluções possíveis e na selecção da "melhor solução".

## **Keywords**

Aggregate planning production, mixed integer multiple objective linear programming, Interactive methods

## 1. Introdução

O planeamento agregado da produção (PAP) tem vindo a merecer uma importância crescente por parte dos responsáveis pela gestão da produção. De facto, devido à significativa alteração do ambiente operativo das empresas (nomeadamente, a introdução de novos processos de gestão como o JIT e o TQM, as exigências de uma clientela cada vez mais sofisticada, o fácil acesso às novas tecnologias e a crescente evolução da cooperação entre empresas (tem-se verificado uma mudança de atitude dos gestores relativamente aos problemas do planeamento da produção.

O problema consiste em calcular, período a período e para um determinado nível da capacidade fabril, as quantidades de produto a fabricar, os níveis de mão-de-obra a utilizar e o número de existências, de modo a minimizar os custos de produção da empresa e a satisfazer a sua procura.

A partir dos anos 50, com o trabalho de Holt et al. (1955) que deu origem à conhecida Regra de Decisão Linear, têm sido efectuados numerosos estudos sobre este assunto, tendo sido propostas diversas metodologias de abordagem deste problema, com maior ou menor sofisticação, dando origem, umas, a soluções denominadas óptimas e outras, a soluções não óptimas. Diversos autores têm estabelecido uma classificação sistemática dos principais modelos existentes, assim como os subsequentes desenvolvimentos que vão surgindo. A este propósito, destacam-se os trabalhos de Saad (1982), Nam e Longendram (1992) e Pan e Kleiner (1995).

A maioria dos trabalhos publicados, relata situações em que a solução encontrada para o problema do planeamento agregado utiliza como critério de optimização os custos de produção da empresa. A utilização da programação multiobjectivo tem sido menos utilizada, apesar de revelar bastantes potencialidades, tendo em conta os meios informáticos actualmente disponíveis e as vantagens que esta metodologia pode proporcionar ao gestor, nomeadamente a possibilidade de introduzir mais do que um objectivo na busca da "melhor" solução de compromisso. Os estudos realizados por Masud e Hwang (1980) e MacLeod e Reeves (1992) são um exemplo deste tipo de aplicação.

O trabalho aqui apresentado tem como objectivo desenvolver uma abordagem multiobjectivo aplicada ao problema do PAP, utilizando três objectivos para avaliação do desempenho da empresa: (1) a maximização do lucro, (2) a minimização do número de encomendas com entrega atrasada e (3) a minimização das flutuações do número de trabalhadores.

Para demonstrar a aplicação do modelo utilizámos os dados fornecidos por uma empresa portuguesa que se dedica à exploração de granito para a construção civil. Esta empresa possui um número anual de vendas da ordem do meio milhão de contos e tem actualmente 36 trabalhadores. A sua produção é totalmente absorvida pelo mercado interno, enfrentando uma competição bastante forte, onde os custos de produção e o cumprimento dos prazos de entrega são condições essenciais para poder sobreviver.

Para tornar esta abordagem atractiva ao agente de decisão (AD), criámos também um sistema de apoio à decisão (SAD) para permitir ao gestor encontrar uma solução sem necessitar de ter conhecimentos específicos de programação multiobjectivo. A descrição deste SAD será apresentada num outro trabalho a publicar brevemente.

Após termos situado o problema que nos propomos estudar, iremos em seguida explicitar um modelo de programação linear inteira mista multiobjectivo aplicado ao PAP. Na Secção 3, apresentamos um método interactivo para a sua resolução, na Secção 4, apresentamos os resultados obtidos com este estudo e, na Secção 5, iremos referir as principais conclusões deste trabalho.

## 2. Um modelo de programação linear inteira mista multiobjectivo aplicado ao planeamento agregado da produção

A abordagem multiobjectivo permite contemplar a existência de diferentes pontos de vista emergentes do problema de decisão, sem forçar a sua conversão para uma unidade comum (ver Roy, 1996).

A análise das perdas e ganhos nos objectivos constitui uma abordagem muito interessante para o AD, que evolui no domínio do problema, possibilitando o aprofundamento da compreensão da complexidade da realidade em questão, permitindo reter para a decisão, um conjunto de soluções alternativas "escondidas" numa abordagem mono-objectivo, que se lhe impõe analisar.

O papel do AD é fundamental quer ao nível da definição das funções objectivo, quer da avaliação das soluções resultantes, projectando sobre estas as suas preferências.

Seguidamente apresenta-se a notação utilizada no modelo multiobjectivo proposto em Gomes da Silva (2000) e Gomes da Silva et al. (2000).

### 1. Conjuntos gerais:

- $m$ , número de tipos de trabalhadores ( $j = 1, \dots, m$ );
- $q$ , número de produtos ( $i = 1, \dots, q$ );
- $n$ , número de períodos ( $t = 1, \dots, n$ ).

### 2. Parâmetros :

- $c_{it}$  - Custo de produção unitário do produto  $i$  no período  $j$ , excluindo todos os custos seguintes.
- $ca_{it}$  - Custo com o atraso na entrega de uma unidade do produto  $i$  no período  $t$ .
- $cr_{jt}$  - Custo com a remuneração normal de um trabalhador da categoria  $j$  no período  $t$ .
- $ce_{it}$  - Custo com a existência em armazém de uma unidade do produto  $i$  no período  $t$ .
- $cd_{jkt}$  - Custo com o despedimento de um trabalhador da categoria  $j$ , contratado no período  $k$  e despedido no período  $t$ .
- $cf_{jt}$  - Custo com a formação de um trabalhador da categoria  $j$  no período  $t$ .
- $ch_{jt}^{ext}$  - Custo por hora extraordinária de um trabalhador da categoria  $j$  no período  $t$ .

$pr_{it}$  - Preço de venda unitário do produto  $i$  no período  $t$ .

$cs_{it}$  - Custo de subcontratação de uma unidade do produto  $i$  no período  $t$ .

$h_{jt}^{emax}$  - Número máximo de horas extraordinárias do trabalhador da categoria  $j$  no período  $t$ .

$h_{jt}^{nmax}$  - Número máximo de horas normais do trabalhador da categoria  $j$  no período  $t$ .

$h_{ji}$  - Número de horas necessárias, por trabalhador da categoria  $j$ , para produzir uma unidade do produto  $i$ .

$t_{j0}$  - Número de trabalhadores da categoria  $j$ , no início do período de planeamento ( $t = 0$ ).

$t_{jt}^{max}$  - Número máximo de trabalhadores da categoria  $j$  no período  $t$ .

$t^{max}$  - Número máximo total de trabalhadores (assume-se igual em todos os períodos).

$\mu$  - Número legal de períodos, após contratação, em que um trabalhador não pode ser despedido.

$\alpha_j$  - Número de períodos de formação do trabalhador da categoria  $j$ .

$a_{i0}$  - Número de unidades do produto  $i$  em atraso no início do planeamento (período 0).

$\beta_j^{norm}$  - Percentagem do horário de trabalho normal efectuado por um trabalhador em formação da categoria  $j$ .

$\beta_j^{ext}$  - Percentagem do horário de trabalho extraordinário efectuado por trabalhador em formação da categoria  $j$ .

$d_{it}$  - Procura em unidades do produto  $i$  no período  $t$ .

$e_{i0}$  - Existências do produto  $i$  no período 0.

$e^{max}$  - Capacidade máxima de armazenamento de produtos acabados (assume-se constante em todos os períodos).

$\xi_{jk}$  - Proporção dos trabalhadores da categoria  $j$ , existentes no início do planeamento, passíveis de despedimento no período  $k$ . ( $\xi_{jk}=1$ , com  $k > 0$ )

$P_{it}^{max}$  - Capacidade produtiva máxima em unidades do produto  $i$  no período  $t$ .

$P^{max}$  - Capacidade global de produção, em unidades de produto, em cada período (assume-se constante em todos os períodos).

$s_{it}^{max}$  - Número máximo de unidades do produto  $i$  a subcontratar no período  $t$ .

$v_i$  - Espaço de armazenamento ocupado por cada unidade de produto  $i$ .

Variáveis de decisão:

$A_{it}$  - Unidades do produto  $i$  em atraso no período  $t$ .

$E_{it}$  - Existências do produto  $i$  no período  $t$ .

$H_{jt}^n$  - Número de horas de trabalho normal dos trabalhadores da categoria  $j$  no período  $t$ .

$H_{jt}^e$  - Número de horas de trabalho extraordinário dos trabalhadores da categoria  $j$  no período  $t$ .

$T_{jt}$  - Número de trabalhadores da categoria  $j$  a utilizar no período  $t$ .

$TF_{jt}$  - Número de trabalhadores da categoria  $j$  em formação no período  $t$ .

$TC_{jt}$  - Número de trabalhadores da categoria  $j$  contratados no período  $t$  ( $TC_{j0} = t_{j0}$ ).

$TD_{jkt}$  - Número de trabalhadores da categoria  $j$  contratados no período  $k$  e despedidos no período  $t$ .

$P_{it}$  - Produção do produto  $i$  no período  $t$ .

$S_{it}$  - Unidades do produto  $i$  subcontratadas no período  $t$ .

As variáveis  $T_{jt}$ ,  $TF_{jt}$ ,  $TC_{jt}$  e  $TD_{jkt}$  são inteiras.

As funções objectivo maximização do lucro, minimização das encomendas com entrega atrasada e a minimização das flutuações do número de trabalhadores foram utilizadas para avaliar as diferentes alternativas (soluções) que se colocam ao AD. As expressões matemáticas destas funções são as seguintes:

### 1. Maximização do Lucro:

$$\begin{aligned} \text{Max } z_1 = & \sum_{i=1}^q \sum_{t=1}^n pr_{it}(d_{it} - A_{it} + A_{it-1}) - \sum_{i=1}^q \sum_{t=1}^n ce_{it}E_{it} - \\ & \sum_{i=1}^q \sum_{t=1}^n ca_{it}A_{it} - \sum_{j=1}^m \sum_{t=1}^n cf_{jt}TF_{jt} - \sum_{j=1}^m \sum_{t=1}^n ch_{jt}^{ext}H_{jt}^e - \sum_{i=1}^q \sum_{t=1}^n c_{it}P_{it} - \\ & \sum_{j=1}^m \sum_{t=1}^n cr_{jt}T_{jt} - \sum_{j=1}^q \sum_{k=0}^n \sum_{t=1}^n cd_{jkt}TD_{jkt} - \sum_{i=1}^q \sum_{t=1}^n cs_{it}S_{it} \end{aligned}$$

### 2. Minimização das encomendas com entrega atrasada:

$$\text{Min } z_2 = \sum_{t=1}^n \sum_{i=1}^n A_{it}$$

### 3. Minimização da flutuação do número de trabalhadores:

$$\text{Min } z_3 = \sum_{t=1}^n \sum_{j=1}^m TC_{jt} + \sum_{j=1}^m \sum_{k=0}^n \sum_{t=1}^n TD_{jkt}$$

As restrições existentes delimitam o âmbito de construção das alternativas do planeamento agregado e podem ser agrupadas em três grandes famílias: número de trabalhadores, quantidades produzidas e níveis de existências em armazém. Estas reflectem as principais limitações de ordem técnica, operacional e legal que se colocam no PAP.

- Restrições respeitantes ao número de trabalhadores:

1. Definição dos trabalhadores por categoria e período.

$$T_{jt} = T_{jt-1} + TC_{jt} - \sum_{k=0}^t TD_{jkt} \quad j = 1, \dots, m; t = 1, \dots, n$$

2. Limitação legal para despedimento.

$$\sum_{\substack{w=k \\ 1 \leq w \leq n}}^{k+\mu-1} TD_{jkw} = 0 \quad j = 1, \dots, m; k = 0, \dots, n$$

3. Limitação do despedimento:

$$\sum_{\substack{w=1 \\ l \geq k}}^n TD_{jkw} \leq \xi_{jk} TC_{jk} \quad j = 1, \dots, m; k = 0, \dots, n$$

4. Número máximo de trabalhadores por categoria e período.

$$T_{jt} \leq t_{jt}^{\max} \quad j = 1, \dots, m; t = 1, \dots, n$$

5. Número total máximo de trabalhadores.

$$\sum_{j=1}^m T_{jt} \leq t^{\max} \quad t = 1, \dots, n$$

6. Trabalhadores em formação.

$$TF_{jt} = \sum_{\substack{k=t-\alpha_j+1 \\ k \geq 1}}^t TC_{jk} - \sum_{\substack{k=t-\alpha_j+1 \\ w \geq 1}}^t \sum_{\substack{l=t-\alpha_j+1 \\ k \geq 1}}^t TD_{jkw} \quad j = 1, \dots, m; k = 0, \dots, n$$

7. Limitação do horário normal.

$$H_{jt}^n \leq h_{jt}^{n\max} (T_{jt} - TF_{jt}) + \beta_j^{\text{norm}} h_{jt}^{n\max} TF_{jt} \quad j = 1, \dots, m; t = 1, \dots, n$$

8. Limitação do horário extraordinário.

$$H_{jt}^e \leq h_{jt}^{e\max} (T_{jt} - TF_{jt}) + \beta_j^{\text{ext}} h_{jt}^{e\max} TF_{jt} \quad j = 1, \dots, m; t = 1, \dots, n$$

• Restrições respeitantes às quantidades a produzir:

9. Equilíbrio da produção.

$$P_{it} + S_{it} + A_{it} = d_{it} + A_{it-1} + E_{it} - E_{it-1} \quad i = 1, \dots, q; t = 1, \dots, n$$

10. Produção global máxima.

$$\sum_{i=1}^q P_{it} \leq p^{\max} \quad t = 1, \dots, n$$

11. Produção máxima por produto e período.

$$P_{it} \leq p_{it}^{\max} \quad i = 1, \dots, q; t = 1, \dots, n$$

12. Subcontratação máxima por produto e período.

$$S_{it} \leq s_{it}^{\max}, \quad i = 1, \dots, q; t = 1, \dots, n$$

13. Limitação da produção pela disponibilidade horária.

$$\sum_{i=1}^q h_{ji} P_{it} = H_{jt}^n + H_{jt}^e \quad j = 1, \dots, m; t = 1, \dots, n$$

- Restrições respeitantes aos níveis de existências em armazém:

14. Capacidade máxima de armazenamento de produtos acabados.

$$\sum_{i=1}^q v_i E_{it} \leq e^{\max}, \quad t = 1, \dots, n$$

O modelo contém, assim, um total de  $n(8m+3+3q)+2m$  restrições e  $n(m(5+n)+4q)$  variáveis, das quais  $mn(4+n)$  são inteiras. O acréscimo do número de restrições é de  $8m+3+3q$ ,  $8n+2$  e  $3n$  para variações unitárias de  $n$ ,  $m$  e  $q$ , respectivamente. Para o número de variáveis, o aumento unitário de  $n$ ,  $m$  e  $q$  conduz a acréscimos de  $n(5+n) + mn$ ,  $n(5+n)$  e  $4n$ , respectivamente.

### 3. Um método interactivo

Uma vez especificado o modelo multiobjectivo, colocou-se o problema da escolha do método de análise do problema. A opção por um método interactivo fundamentou-se na possibilidade de um processo de decisão flexível, em que o AD assume um papel fundamental.

Para o efeito, recorreremos ao método interactivo desenvolvido por Gomes da Silva (2000) e Gomes da Silva et al. (2000), tendo por base as seguintes premissas: quando o AD tem um primeiro contacto com algumas soluções é-lhe difícil não só afirmar se uma dada solução é satisfatória ou não e fixar mínimos de satisfação para alguns objectivos, mas também especificar a quantidade a prescindir em alguns objectivos de modo a melhorar o nível de pelo menos um dos restantes objectivos. A informação de que o AD dispõe no início do processo de interacção não é suficiente para fundamentar aquelas opções.

O método desenvolvido está, assim, dividido em duas fases e tem por base a expansão e contracção de um conjunto gerador (GE) sempre assistidas pelo AD:

1. Na primeira fase, intitulada fase de aprendizagem orientada, o AD depara-se com algumas soluções não dominadas, obtidas através da minimização de uma função ponderada e aumentada a um ponto de referência (ver Steuer, 1986). Este ponto de referência pode ser fixado pelo AD, bem como os ponderadores utilizados na função. O AD pode ainda introduzir restrições no espaço dos objectivos de modo a evitar a apresentação de soluções

com níveis que considera intoleráveis em certas funções objectivo e/ou permitir ainda uma análise condicionada. Se a solução apresentada for considerada aceitável ocorre a expansão de GE. O objectivo desta fase é que o AD "aprenda" o seu problema de um modo flexível e adquira um conjunto de conhecimentos que lhe permita fundamentar as respostas às questões colocadas na segunda fase do método.

2. A segunda fase, intitulada fase de pesquisa orientada, tem início quando o AD pretender aprofundar o conhecimento de GE. Nesta fase, pretende-se fornecer ao AD mais informação sobre as soluções pertencentes a GE e, eventualmente, gerar novas soluções a partir deste conjunto. São disponibilizados cinco modos de interacção: (1) avaliar soluções; (2) procurar soluções vizinhas; (3) comparar soluções; (4) eliminar soluções; e (5) regressar à fase de aprendizagem orientada. Vejamos em detalhe cada um destes modos de interacção:

- No modo de interacção avaliar soluções, o AD escolhe uma solução de GE e especifica o nível mínimo aceitável para os objectivos com níveis não satisfatórios, bem como a quantidade a prescindir em, pelo menos, um dos objectivos com níveis satisfatórios. Após a obtenção da respectiva solução, o AD pode refazer as especificações indicadas. Se a solução apresentada for considerada satisfatória é adicionada a GE, ocorrendo, assim, a sua expansão.
- No modo de interacção procurar soluções vizinhas, o AD escolhe uma solução de GE na vizinhança da qual pretende conhecer novas soluções. Se o AD considerar que a solução vizinha é satisfatória, ela é adicionada a GE ocorrendo a expansão deste conjunto.
- No modo de interacção comparar soluções, o AD escolhe as soluções do conjunto GE que pretende comparar. A escolha destas soluções baseia-se no interesse em pormenorizar a sua distinção, particularmente frequente quando os níveis das funções objectivo são próximos. Ao proceder à sua comparação o AD pode não eliminar nenhuma delas, por considerá-las incomparáveis, ou, eliminar uma delas, por: (1) considerá-la "menos satisfatória" relativamente à outra; ou (2) porque as duas soluções lhe são indiferentes. Neste modo, nunca pode ocorrer a expansão do conjunto GE.
- No modo eliminar soluções, o AD pode excluir directamente qualquer uma das soluções do conjunto GE. A razão para a escolha deste modo reside no facto de o AD considerar desinteressante uma ou mais soluções, relativamente ao conhecimento que acumulou, não pretendendo considerá-las geradoras de outras soluções.
- No modo regressar à fase de pesquisa livre, o AD pretende aumentar o seu conhecimento sobre a região não dominada, pelo que regressa ao início do método, podendo expandir o conjunto GE. A opção por esta interacção pode dever-se à eliminação de todos os elementos de GE ou ao interesse em incluir novas soluções não geradas a partir da vizinhança de nenhum dos actuais elementos de GE, ou da sua avaliação.

Se o conjunto GE possuir apenas uma solução e o AD não pretender passar a nenhum dos modos de interacção disponíveis em que pode eventualmente ocorrer a expansão desse conjunto, então essa solução será a "solução final". Perante o processo de "descoberta", exploração e avaliação de soluções, o AD considera o único elemento de GE "superior" a todos os outros que lhe foram apresentados.

Na Figura 1 está representada a estrutura genérica do método interactivo desenvolvido.

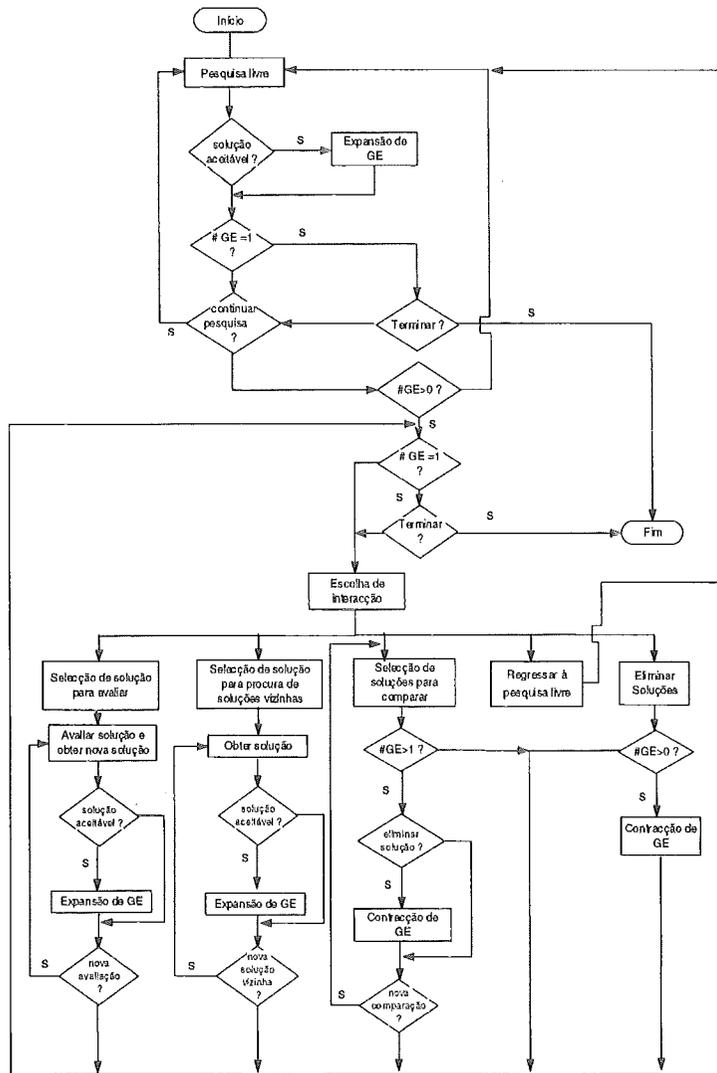


Figura 1 - Esquema do método interactivo desenvolvido

#### 4. Resultados

Os resultados apresentados nesta secção resultam da utilização dos dados da empresa de granitos para a construção civil referida na introdução. O horizonte de planeamento é de 12 meses, e os produtos e os trabalhadores foram agregados em 3 grupos homogéneos e 3 tipos ou especialidades, respectivamente.

Para este caso de aplicação, em que  $n$  (número de períodos de planeamento) = 12,  $m$  (número de tipos de trabalhadores) = 3 e  $q$  (número de produtos) = 3, o problema

multiobjectivo proposto assume a seguinte dimensão: 438 restrições, 756 variáveis, sendo 576 inteiras.

Seguidamente, apresenta-se o resultado do processo de aplicação do método interactivo apresentado.

Inicialmente, foi fornecida ao AD a matriz dos óptimos individuais e, com base nesta, a correspondente matriz das percentagens. O objectivo desta informação é explicitar os limites da optimização de cada uma das funções objectivo (Tabela 1), e o posicionamento relativo dos restantes valores (Tabela 2). Este posicionamento relativo é calculado do seguinte modo:  $c_{ij} = \frac{100}{z_i^* - z_i^{\text{nad}}} (z_{ij} - z_i^{\text{nad}})$ , em que  $z_i^*$  é o valor óptimo da  $i$ -ésima função objectivo,  $z_i^{\text{nad}}$  corresponde ao pior valor da linha  $i$  da matriz dos óptimos individuais e  $z_{ij}$  é o elemento  $ij$  desta matriz. Assim, por exemplo, o elemento da 3ª linha e 2ª coluna da tabela 2 é:  $c_{32} = \frac{100}{0-10} (8-10) = 20$ .

	$z_1^*$	$z_2^*$	$z_3^*$
$Z_1$ (em $10^3$ \$)	32.942,8623	32.110,5222	-14.023,6492
$Z_2$ (em $m^2$ )	3.621,2578	3.621,2578	49.247,0582
$Z_3$ (em nº trab.)	10	8	0

Tabela 1 - Matriz dos óptimos individuais

	$z_1^*$	$z_2^*$	$z_3^*$
$Z_1$ (em $10^3$ \$)	100,0000	98,2278	0,0000
$Z_2$ (em $m^2$ )	100,0000	100,0000	0,0000
$Z_3$ (em nº trab.)	0,0000	20,0000	100,0000

Tabela 2 - Matriz das percentagens dos óptimos individuais

#### 4.1 Primeira fase do método interactivo

Na primeira fase do método interactivo, o AD atribuiu ponderadores às funções objectivo e introduziu restrições no espaço dos objectivos. O resultado deste processo é apresentado na Tabela 3. Foram geradas 8 soluções que o AD decidiu guardar, embora nesta fase fosse possível rejeitar algumas das soluções apresentadas. Estas soluções foram obtidas sequencialmente da seguinte forma:

- Numa primeira interacção, o AD manteve como ponto de referência o ponto ideal perturbado, (32.946,1566; 3.620,8956; -0,0001), e definiu o vector (0,9;0,08;0,02) como ponderador, pretendendo, assim, privilegiar a "proximidade" com o valor óptimo do lucro. A solução obtida foi  $S_1 = (32.128,7560; 3.621,2578; 8)$ . Por se tratar do início do protocolo de interacção o AD não viu qualquer razão para rejeitar esta solução, apesar do nível muito pouco satisfatório do valor da flutuação dos trabalhadores. Esta solução foi integrada em GE. Assim, temos  $GE = \{S_1\}$ .

- Numa segunda interacção, o AD decidiu aumentar muito sensivelmente a "proximidade" com o valor óptimo da função objectivo flutuação do número de trabalhadores, reduzindo a "proximidade" com a função objectivo encomendas com entrega atrasada. Assim, o vector ponderador definido foi (0,9;0,05;0,05). Obteve-se, então, a solução  $S_2 = (31.390,4585; 3.992,4447; 7)$ . Nesta solução, a percentagem do valor óptimo de cada uma das funções objectivo é 96,69%, 99,19% e 30%, respectivamente. O AD decidiu também juntar  $S_2$  a GE, i.e.,  $GE = \{S_1, S_2\}$ .
- Numa terceira interacção, o AD manteve o ponto de referência e o vector ponderador, mas está interessado em que a nova solução não apresente um valor superior a 6 para a flutuação do número de trabalhadores. A solução obtida foi  $S_3 = (29.166,0702; 5.550,2021; 6)$ . Esta solução apresenta ainda níveis muito reduzidos (em % dos óptimos) para as encomendas com entrega atrasada, mas no entanto o AD integrou-a em GE.
- Numa quarta interacção, o AD definiu como vector de ponderadores (0,33; 0,33; 0,33), não privilegiando nenhum dos desfasamento do ponto de referência, e continuou a reduzir o valor da flutuação do número de trabalhadores, impondo agora, um máximo de 5. Obteve-se a solução  $S_4 = (15.663,0702; 18.428,7785; 3)$ , que também foi introduzida em GE. Nesta solução, o valor da flutuação do número de trabalhadores melhorou significativamente, correspondendo o seu valor a 70% do valor óptimo.
- Numa quinta interacção, o AD continuou preocupado com a flutuação do número de trabalhadores e pretendeu conhecer o nível do lucro e das encomendas com entrega atrasada compatíveis com uma flutuação do número de trabalhadores não superior a 1. Definiu-se como vector ponderador: (0,49; 0,49; 0,02). O ponto de referência não se alterou. Na solução obtida,  $S_5 = (-3.754,3751; 39.273,000; 1)$ , o lucro apresenta valores negativos e o valor dos atrasos corresponde apenas a 21,86% do valor óptimo. Esta solução também foi guardada, apresentando GE a seguinte composição:  $GE = \{S_1, S_2, S_3, S_4, S_5\}$ .
- Numa sexta interacção, o AD conhecia já algumas soluções correspondentes à melhoria contínua da flutuação do número de trabalhadores. O AD decidiu, então, inverter o processo de pesquisa e procurar soluções com elevado valor para o lucro. Como ponto de referência mantém o ponto ideal perturbado e redefine o vector ponderador que passa a ser igual a (0,999; 0,005; 0,005). A solução obtida foi  $S_6 = (32.942,8623; 3.621,2578; 10)$ . Esta solução foi também integrada em GE, embora o AD tivesse considerado menos satisfatório o valor para a flutuação do número de trabalhadores.
- Numa sétima interacção, o AD pretendeu conhecer uma solução em que o lucro se aproxime dos 20000 contos, as encomendas com entrega atrasada dos 3500m<sup>2</sup> e a redução dos trabalhadores dos 5. Considerou-se o seguinte vector ponderador: (0,33; 0,33; 0,33). A solução obtida foi  $S_7 = (27.547,7773; 6.740,0970; 5)$ , tendo também sido incluída em GE.

- Finalmente, obteve-se  $S_8 = (4.840,8903; 3.041,9654; 2)$ , mantendo-se o ponto ideal perturbado como ponto de referência. O AD privilegiou a "proximidade" com o melhor valor da flutuação do número de trabalhadores, seguindo-se a "proximidade" com o lucro e finalmente com as encomendas com entrega atrasada, traduzido-se no seguinte vector ponderador:  $(0,25; 0,15; 0,6)$ . Após incluir esta solução em GE, este conjunto apresenta a seguinte composição:  $GE = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8\}$ . O AD considerou que já reuniu um conjunto representativo de soluções para o seu problema. A Tabela 3 resume as soluções de GE.

Solução	$Z_1$ (em $10^3$ \$)	$Z_1$ (em $m^2$ )	$Z_3$ (em nº trab.)	% óptimo $Z_1$	% óptimo $Z_2$	% óptimo $Z_3$
$S_1$	32.128,7560	3.621,2578	8	98,2666	100,0000	20,0000
$S_2$	31.390,4585	3.992,4447	7	96,6947	99,1865	30,0000
$S_3$	29.166,4391	5.550,2021	6	91,9593	95,7723	40,0000
$S_4$	15.663,0702	18.428,7785	3	63,2083	67,5457	70,0000
$S_5$	-3.754,3751	39.273,0000	1	21,8651	21,8588	90,0000
$S_6$	32.942,8623	3.621,2578	10	100,0000	100,0000	0,0000
$S_7$	27.547,7773	6.740,0970	5	88,5129	93,1643	50,0000
$S_8$	4.840,8903	30.041,9654	2	40,1659	42,0926	80,0000

Tabela 3 - Resultado do processo de interacção com o AD - 1ª fase

Nas Figuras 2, 3 e 4 é apresentada a evolução das soluções de GE.

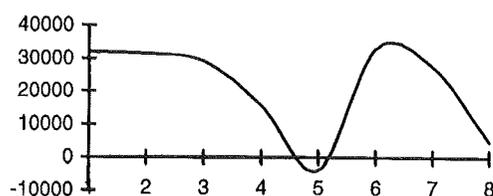


Figura 2 - Evolução do lucro ( $Z_1$ )

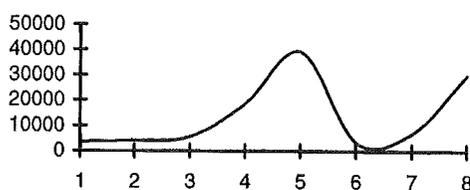


Figura 3 - Evolução das encomendas com entrega atrasada ( $Z_2$ )

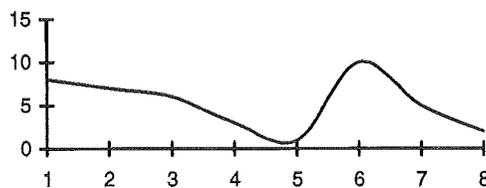


Figura 4 - Evolução das flutuações do número de trabalhadores ( $Z_3$ )

#### 4.2 Segunda fase do método interactivo

Ao confrontar-se com todas as soluções do conjunto GE o AD comparou as soluções  $S_2$  e  $S_3$ . Da comparação eliminou a solução  $S_3$ , pelo que foi excluída de GE. Também, excluiu directamente  $S_5$  e  $S_8$  por possuírem valores para o lucro bastante baixos, comparativamente com os correspondentes valores óptimos. Após a eliminação de todas estas soluções o conjunto GE reduziu-se a  $GE = \{S_1, S_2, S_4, S_6, S_7\}$ .

A solução  $S_4$ , apresenta níveis percentuais homogêneos, justificando a pesquisa na sua vizinhança. Uma solução vizinha encontrada foi  $S_9 = (22.640,6152; 11.074,9704; 4)$ . Nesta solução, o lucro e as encomendas com entrega atrasada melhoraram simultaneamente e a perda verificada na flutuação do número de trabalhadores não é drástica, pelo que a solução foi guardada.

De novo, perante o conjunto GE, o AD escolheu a solução  $S_1$  para avaliar. Nesta solução, o nível do lucro foi considerado satisfatório, estando disposto a prescindir de 3.500 contos, o nível da flutuação do número de trabalhadores foi considerado não satisfatório, exigindo-se que a flutuação do número de trabalhadores não exceda 6 trabalhadores. Após resolução do problema correspondente obteve-se novamente a solução  $S_3$  que foi acrescentada ao conjunto GE ( $GE = \{S_1, S_2, S_3, S_4, S_6, S_7, S_9\}$ ), o que demonstra uma certa incoerência na apreciação das soluções por parte do AD.

A solução  $S_2$  apresenta um mau desempenho da flutuação do número de trabalhadores. A existência em GE, de soluções com níveis mais moderados nesta função objectivo, em simultâneo com níveis aceitáveis para os restantes objectivos, permitiu que o AD eliminasse  $S_2$ .

O AD pretende avaliar a solução  $S_4$ , considerando o nível do lucro não satisfatório, exigindo um nível mínimo de 20.000 contos, para tal permite que a flutuação do número de trabalhadores aumente em 1 trabalhador. A solução obtida foi novamente  $S_9$ .

Perante as soluções apresentadas, o AD considerou não satisfatória a flutuação dos trabalhadores superior a 4, o que reduziu o conjunto GE às soluções  $S_4$  e  $S_9$ .

Os níveis próximos na função objectivo flutuação do número de trabalhadores, nestas duas soluções, justificou uma comparação mais atenta. Dessa comparação resultou que o aumento do lucro e a diminuição das encomendas com entrega atrasada tornam aceitável o aumento da flutuação do número de trabalhadores, pelo que  $S_4$  foi eliminada. O AD considerou  $S_9$  a solução final (ver Tabela 4).

Solução	$Z_1$ (em $10^3$ \$)	$Z_1$ (em $m^2$ )	$Z_3$ (em n° trab.)	% óptimo $Z_1$	% óptimo $Z_2$	% óptimo $Z_3$
$S_9$	22.640,6152	11.074,9704	4	78,0647	83,6634	60,0000

Tabela 4 - Resultado do processo de interacção com o AD - 2ª fase

## 5. Conclusões

A análise multiobjectivo permite que o AD contemple explicitamente vários aspectos do problema de decisão. Esta característica foi essencial para a aplicação da programação multiobjectivo ao PAP.

No âmbito do PAP, o lucro ou os custos são habitualmente considerados os únicos critérios de avaliação das diferentes alternativas de planeamento agregado. Consequentemente, apenas são considerados os aspectos passíveis de redução a uma unidade monetária. No entanto, estes não constituem os únicos factores relevantes. O clima laboral e a insatisfação dos clientes são também fundamentais. As consequências de uma política de recursos humanos caracterizada por uma forte flutuação do número de trabalhadores não podem ser descuradas ao nível da produtividade e da estabilidade laboral, bem como a insatisfação dos clientes devido aos atrasos na entrega das encomendas.

Em Nam e Longendram (1992) e Pan e Kleiner (1995) é evidenciado que o nível de agregação representa uma razão da ineficácia prática dos modelos. Neste sentido, trabalhou-se no modelo desenvolvido com um menor nível de agregação relativamente aos trabalhadores e produtos, o que permitiu um maior refinamento das soluções apresentadas. Sendo impossível contemplar todas as possibilidades de acção e as restrições existentes no PAP, incluíram-se vertentes que permitem uma maior adequação à realidade, representada nomeadamente pela existência de subcontratação de produtos, de um período de formação dos trabalhadores recém-contratados, de um período legal de despedimento e de uma limitação da capacidade de armazenamento.

Promover a aprendizagem do AD e flexibilizar o processo de decisão motivaram o desenvolvimento de um método interactivo.

Com a aplicação do método interactivo que desenvolvemos, verificou-se que a informação inicialmente exigida ao AD não é elevada nem complexa, tanto mais que esta não tem que ser, obrigatoriamente, fundamentada. Também o processo interactivo subjacente não contempla imperativamente a coerência do AD nas sucessivas reacções às soluções apresentadas. Este

aspecto está na base da apresentação de soluções bastante diversas, permitindo um maior conhecimento do espaço das soluções possíveis.

A disponibilidade de diferentes modos de interacção na segunda fase do método torna-o suficientemente flexível para intensificar a aprendizagem do problema, atribuindo ao AD um papel activo na apresentação das soluções e possibilitando, ainda, que prossiga sistematicamente na melhoria de uma dada função objectivo, caso seja essa a finalidade da pesquisa.

Um outro aspecto importante, relacionado com o processo interactivo, é o tempo de resposta ao AD. Apesar do modelo apresentar um número não muito reduzido de variáveis inteiras (576 variáveis) e de restrições (438), este tem uma estrutura particular fazendo com que o tempo médio de resposta em cada interacção seja apenas de 15 segundos. Este tempo de resposta foi obtido num Pentium II a 133 MHZ, tendo-se utilizado o optimizador linear LINGO (versão 5.0).

## Referências

- [1] Gomes da Silva, C., Programação linear inteira mista multiobjectivo: aplicação ao planeamento agregado da produção, Dissertação de Mestrado, Faculdade de Economia da Universidade de Coimbra (2000).
- [2] Gomes da Silva, C., Figueira, J. e Lisboa, J., An aggregate production planning model based on mixed integer multiple criteria linear programming, apresentado para publicação. [versão reduzida em: Actas das 50 ésimas jornadas do grupo europeu Apoio Multicritério à Decisão, 28 de Setembro a 2 de Outubro de 1999, Cerisy-la-Salle, Normandia, França, 4p] (2000).
- [3] Holt, C., Modigliani, F. e Muth, J., Deviation of a linear decision rule for production and employment, *Management Science* 2 (1956) 159-177.
- [4] MacLeod, K. e Reeves, G., An application of the AXIS solution framework to multiple objective aggregate production planning, *Decision Sciences* 23 (1992) 1315-1332.
- [5] Masud, A. e Hwang C., An aggregate planning model and application of three multiple objective decision methods, *International Journal of Production Research* 18 (1980) 741-752.
- [6] Mellicamp, J.M. e Love, R., Production switching heuristics for the aggregate planning problem, *Management Science* 24 (1978) 1242-1251.
- [7] Nam, S. e Logendran, R., Aggregate production planning - A survey of models and methodologies, *European Journal of Operations Research* 61 (1992) 255-272.
- [8] Pan L. e Kleiner, B., *Aggregate planning today*, Work Study (MCB University Press) 44 (1995) 4-7.
- [9] Roy, B., *Multicriteria methodology for decision aiding*, Kluwer Academic Publishers, Dordrecht (1996).
- [10] Saad, G., An overview of production planning models: structural classification and empirical assessment, *International Journal of Production Research* 20 (1982) 105-114.
- [11] Steuer, R., *Multiple criteria optimization: Theory, computation, and application*, John Wiley and Sons, New York (1986).

/

,

# DISCRETIZATION METHODS FOR SEMI-INFINITE PROGRAMMING

**A. Ismael F. Vaz**

**Edite M.G.P. Fernandes**

Departamento de Produção e Sistemas  
Escola de Engenharia-Univ. do Minho  
Campus de Gualtar  
4710 Braga-Portugal

**M. Paula S.F. Gomes**

Mechanical Engineering Department  
Mechatronics in Medicine Laboratory  
Imperial College of Science, Tech. and Medicine  
London SW7 2BX United Kingdom

## Abstract

Discretization methods represent an important class of methods to solve semi-infinite programming (SIP) problems. Authors have claim that their algorithms for linear SIP could be extended to nonlinear SIP problems. Nevertheless numerical results on nonlinear SIP using a discretization method are uncommon. In this paper two discretization methods are described and numerical results for linear and nonlinear SIP problems are shown. The methods described are proposed by Hettich [4, 5] and Reemtsen [8] and adapted for nonlinear semi-infinite programming. The problems collection is the one coded in SIPAMPL [9].

## Keywords

Discretization methods, semi-infinite programming.

## 1. Introduction

Semi-infinite programming (SIP) problems appear in many engineering areas such as air control pollution [6], robotics [3], production planning [7, 10] and in Chebyshev approximation theory [4, 5, 8]. A nonlinear semi-infinite programming problem is described in the mathematical form as follows:

$$\begin{aligned}
 & \min_{x \in \mathbb{R}^n} f(x) \\
 & \text{s. t. } g_i(x,t) \leq 0, \quad i = 1, \dots, m \\
 & \quad h_i(x) \leq 0, \quad i = 0, \dots, o \\
 & \quad h_i(x) = 0, \quad i = o + 1, \dots, q \\
 & \quad \forall t \in T.
 \end{aligned} \tag{1}$$

Usually  $T \in \mathbb{R}^p$  is a cartesian product of intervals  $([\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \dots \times [\alpha_p, \beta_p])$ .

These problems are called semi-infinite programming problems due to the constraints  $g_i(x, t) \leq 0$ ,  $i = 1, \dots, m$ . We can think of  $T$  as an infinite index set and therefore (1) is a problem with finitely many variables over an infinite set of constraints.

There are three main classes of methods to solve a SIP problem (see [6]): discretization methods, exchange methods and reduction methods. Discretization methods replace the set  $T$  by a finite set of points (usually a grid of points) and the semi-infinite programming problem (1) will result in a finite programming problem. In exchange methods the infinite constraints are replaced by finite ones and a sequence of finite programming sub-problems are solved. These methods are called exchange because between two sub-problems some constraints are dropped out and others are added into the set of constraints. Reduction methods find all the global (and some local) maxima for the infinite constraints and solve the sub-problem that is obtained from the SIP by using the infinite constraints evaluated in the maxima found.

Discretization methods represent a important class of methods to solve semi-infinite programming problems. A wide collection of semi-infinite problems has been coded in AMPL ([1]). The SIPAMPL [9] interface allows the connection of a SIP solver to the SIP problems database coded in AMPL.

In Section 2 we present some notation used throughout the paper. In Section 3 a description of two discretization algorithms is given. The corresponding changes to the original algorithms in order to solve nonlinear semi-infinite programming problems is given in Section 4 and in Section 5 a description of some implementation details is given. Numerical results obtained from SIPAMPL problems collection are presented in Section 6 and the conclusions are discussed in Section 7.

## 2. Definitions and Some Notation

The following definitions are needed to describe the discretization algorithms presented in the next section.

**Definition 2.1** A grid is a set of the form  $T[h = (h_1, h_2, \dots, h_p)] = T \cap \{t = (t_1, t_2, \dots, t_p) : t_i = \alpha_i + jh_i, j = 0, \dots, n_i; i = 1, \dots, p\}$ , where  $n_i = (\beta_i - \alpha_i)/h_i$ .

**Definition 2.2**  $NLP(T[h])$  is the following nonlinear programming sub-problem:

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } & g_i(x, t) \leq 0, \quad i = 1, \dots, m \\ & h_i(x) \leq 0, \quad i = 0, \dots, o \\ & h_i(x) = 0, \quad i = o + 1, \dots, q \\ & \forall t \in T[h], \end{aligned} \tag{2}$$

**Definition 2.3**  $QP(T[h])$  is the quadratic programming sub-problem:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + c^T x$$

$$\begin{aligned} \text{s.t. } \quad & A(t)x - b(t) \leq 0, \\ & \forall t \in T[h]. \end{aligned} \quad (3)$$

The following sets are crucial in the algorithms procedure.

$R(x_k)$  denotes the set of all points of the grid at iteration  $k$  that makes an infinite constraint active, i.e.,  $R(x_k) = \{t \in T[h^k] : g_i(x_k, t) = 0\}$ .

$S(x_k)$  is the set of all points of the grid at iteration  $k$  that makes an infinite constraint active or violated, i.e.,  $S(x_k) = \{t \in T[h^k] : g_i(x_k, t) \geq 0\}$ .

### 3. The Algorithms

Discretization methods try to solve a SIP problem by replacing the infinite set  $T$  by a finite one that is, usually, a grid of points. These methods do not guarantee an exact or near exact solution to the SIP but will solve the SIP in the final grid. These methods start with an initial approximation to the solution and in order to converge to an approximate solution the methods solve the problem in a number of intermediate grids. In each iteration the methods refine the grid in a predefined way. To minimize the number of constraints in the sub-problem the algorithms use only a selected set of points in each grid. Hettich [4,5] and Reemtsen [8] define two discretization methods that we describe in detail.

The algorithm proposed by Hettich for quadratic SIP is as follows:

**Algorithm 3.1** Hettich ([4,5])

step 0: Define  $T[h^0]$ , set  $\tilde{T}[h^0] = T[h^0]$ . Solve  $QP(\tilde{T}[h^0])$  and let  $x_0$  be the solution. Define  $R = R(x_0)$ .

step k: If  $S(x_{k-1}) \not\subseteq \tilde{T}[h^{k-1}]$

then: Make  $\tilde{T}[h^{k-1}] = R \cup S(x_{k-1})$ . Solve  $QP(\tilde{T}[h^{k-1}])$  and let  $x_{k-1}$  be the solution. If  $R(x_{k-1}) \not\subseteq R$

then: Set  $R = R \cup R(x_{k-1})$ .

else: Add one point from  $\tilde{T}[h^{k-1}] \setminus R$  to  $R$ .

Continue with Step k.

else: If  $k > r$  stop.  $\tilde{T}[h^k] = R \cup S(x_{k-1}) \cup N(S(x_{k-1}))$ . Solve  $QP(\tilde{T}[h^k])$  and let  $x_k$  be the solution. Set  $R = R \cup R(x_k)$ . Go to Step k+1.

where  $r$  is the number of refinements and  $N(S(x_{k-1}))$  contains the neighbours of points of  $S(x_{k-1})$  in the full grid  $T[h^k]$  that make an infinite constraint active or violated.

The algorithm proposed by Reemtsen is presented below with some changes in the notation to meet our requirements.

**Algorithm 3.2** Reemtsen ([8])

step 0: Choose  $\tilde{\varepsilon}_k \in (0,1)$  and  $\tilde{\delta}_k \geq 0$  ( $k = 1, \dots, r$ ). Set  $D_0 = T[h^0]$ . Set also  $i = 0$  and  $k = 1$ .

step 1: Solve  $NLP(D_i)$  and let  $x_i$  be the solution.

step 2: If  $k > r$  stop. Set  $B_{i+1} = T[h^k]$ ,  $\varepsilon_{i+1} = \tilde{\varepsilon}_k$  and  $\delta_{i+1} = \tilde{\delta}_k$ .

If  $x_i$  solves the full grid, i.e.,  $g_j(x_i, t) \leq \delta_{i+1}$  for all  $t \in B_{i+1}$  and  $j = 1, \dots, m$

then: Increment  $k$  by one and continue with step 2.

else: Set  $D_{i+1} = \{t \in B_{i+1} : g_j(x_i, t) \geq -\epsilon_{i+1} f(x_i), j = 1, \dots, m\}$ . Increment  $i$  by one and go to step 1.

$f(x_i)$  is the objective value in  $x_i$  (solution of  $NLP(D_i)$ ).

#### 4. Algorithms Modification

The first algorithm implemented in this paper is obtained from the Algorithm 3.1 where the QP sub-problem is replaced by a NLP sub-problem.

The algorithm described by Reemtsen and presented in section 3 may solve nonlinear SIP problems, although the author only showed numerical results for Chebyshev approximation problems, which are linear SIP. To solve nonlinear SIP the set  $D_{i+1}$  had to be replaced by  $D_{i+1} = \{t \in B_{i+1} : g_j(x_i, t) \geq -\epsilon_{i+1}|f(x_i)|, j = 1, \dots, m\}$ , as for the Chebyshev approximation problems the objective function value is always positive. This could be considered a slight modification but our numerical experiments show that the algorithm is very sensitive to the choice of  $D_{k+1}$  and does not converge when applied to problems where the objective function takes negative values. Replacing the sub-problem solver we are able to solve any problem with the general structure presented in (1).

#### 5. Implementation Details

The SIPAMPL interface is used to connect the discretization algorithms based on NLP sub-problem solvers to its problems database. We suggest the reading of [9] for more details.

To manage the points of a grid an hash table is implemented. This type of tables allows a faster and easy way for searching and inserting new points.

The NPSOL ([2]) solver is used for the NLP sub-problem. The success of the NLP solver is crucial to the success of the SIP algorithm.

Both algorithms have a convergence proof. For the proof see the original papers ([4,5,8]). The convergence proof requires that the NLP solver has global convergence. In both algorithms an unconstrained NLP may be solved, when no finite constraints and no points in the grid that make an infinite constraint active or violated (i.e., no points for which  $g_i(x_k, t) \geq 0, \forall i = 1, \dots, m, \forall t \in T[h^k]$ ) are present. In this case the solution of the NLP sub-problem could be too far away from the SIP solution and the next NLP sub-problem could no longer converge. To avoid the situation where no active or violated points exist, the sub-problem is solved with the constraint that corresponds to  $\max g_i(x_k, t) \geq 0, \forall i = 1, \dots, m, \forall t \in T[h^k]$ .

When no initial guess is known for the problem an extra initial NLP sub-problem is solved with five discrete points per dimension and with a random initial guess (random between 0 and 1 in all dimensions). The solution from this NLP sub-problem is the initial guess to the SIP problem.

## 6. Numerical Results

We tested the modified discretization algorithms, mentioned in Section 4, with all, known to us, SIP problems. The numerical results were obtained on a Pentium III at 450Mhz with 128MB of RAM and a Linux Operating System (Red Hat 5.2) with AMPL Student Version 19991027 (Linux 2.0.18).

In Table 1 some relevant data is shown for all the problems is the SIPAMPL database [9]. Two grid refinements are made and they are as Hettich proposed in [4], i.e.,  $h^1 = h^0/2$ ,  $h^{k+1} = h^k/3$ . In Reemtsen version of the algorithm  $\tilde{\epsilon}_0 = 0.01$ ,  $\tilde{\delta}_k = 0$ , for all  $k$ , were used.  $\tilde{\epsilon}_k$  is updated by the formula

$$\begin{cases} \tilde{\epsilon}_1 = \tilde{\epsilon}_0/(2)^p & \text{for } k = 0 \\ \tilde{\epsilon}_{k+1} = \tilde{\epsilon}_k/(3)^p & \text{for } k \neq 0. \end{cases}$$

The notation used is:

Problem	Problem name (filename with .mod extension in SIPAMPL)
nx	Number of finite (x) variables
nt	Number of finite (t) variables
h	Step size in the first grid. The step size used is $(h, \dots, h)^p$
IG	Number of points in the first (initial) grid
FG	Number of points in the last (final) grid
ACH/ACR	Average number of constraints in the NLP solved in Hettich/Reemtsen version of the algorithm (the number of constraints in the first NLP solved is not counted)
NH/NR	Number of NLP solved in Hettich/Reemtsen version of the algorithm
fxH/fxR	Objective function value in the solution of the final grid in Hettich/Reemtsen version of the algorithm.

For problem **elke2** both methods do not converge to a solution, since NPSOL could not find an admissible solution. For problem **elke7** both methods do not converge from the initial guess  $x_0 = (4.85386, \dots, 4.85386)$  proposed by Haaren-Retagne ([3]). Other guess is used  $x_0 = (2.85386, \dots, 2.85386)$ .

Problem	nx	nt	h	IG	FG	ACH	ACR	NH	NR	fxH	fxR
hettich1	9	2	0.1	121	3721	29.5	53	5	5	3.05e-2	3.06e-2
hettich2	3	1	0.1	21	121	7.3	13	4	3	5.38e-1	5.38e-1
hettich3	5	1	0.1	8	46	10.7	13	4	3	3.52e-3	3.55e-3
hettich4	2	1	0.1	8	46	3	0	3	1	0.75e0	1.00e+0
hettich5	3	2	0.1	225	7225	8	284	3	2	5.36e-1	5.36e-1
hettich6	7	2	0.1	176	5551	19.3	49	4	3	2.80e-2	2.81e-2

hettich7	7	2	0.1	176	5551	18	61	4	3	1.78e-1	1.78e-1
hettich8	5	1	0.1	11	61	8	18	3	2	9.74e-4	5.64e-3
hettich9	29	2	0.1	176	5551	30	45	4	3	3.47e-3	3.47e-3
reemtsen1	11	3	0.1	1331	226981	30.7	77	4	3	1.52e-1	1.52e-1
reemtsen2	10	2	0.1	441	14641	19.3	37	4	3	5.83e-2	5.83e-2
reemtsen3	10	2	0.1	441	14641	16	28	3	2	7.35e-1	7.35e-1
reemtsen4	10	2	0.1	441	14641	18.7	102	4	4	1.14e-2	1.14e-2
reemtsen5	11	3	0.1	1000	216000	140	1510	5	4	8.21e-2	8.68e-2
watson1	2	1	0.1	21	121	1	0	3	1	-2.50e-1	-2.50e-1
watson2	2	1	0.1	11	61	16	0	3	1	2.61e+0	2.62e+0
watson3	2	1	0.1	11	61	1	0	3	1	5.53e+0	5.53e+0
watson4a	3	1	0.1	11	61	3.7	7.5	4	3	6.48e-1	6.49e-1
watson4b	6	1	0.1	11	61	7.3	41	4	3	6.16e-1	6.16e-1
watson4c	8	1	0.1	11	61	10	41	4	3	6.16e-1	6.16e-1
watson5	3	1	0.1	11	61	2	21	3	2	4.30e+1	4.30e+0
watson6	2	1	0.1	11	61	1	0	3	1	9.72e+1	9.72e+1
watson7	3	2	0.1	121	3721	1	0	3	1	1.00e+0	1.00e+0
watson8	6	2	0.1	121	3721	8.3	32.5	4	3	2.44e+0	2.44e+0
watson9	6	2	0.1	441	14641	61	0	3	1	-1.20e+1	-1.20e+1
watson10	3	2	0.1	2601	90601	2	25.5	4	3	2.75e-1	2.75e-1
watson11	3	2	0.1	2601	90601	6.7	1096	4	3	-4.39e+0	-4.39e+0
watson12	3	2	0.1	2601	90601	2.7	134.5	4	3	1.95e+0	1.95e+0
watson13	3	2	0.1	2601	90601	4.8	139	5	3	1.95e+0	1.95e+0
watson14	2	1	0.1	21	121	1	0	3	1	2.20e+0	2.20e+0
priceK	2	1	0.1	32	189	2.7	14.5	4	3	-3.00e+0	-3.00e+0
priceS3	4	3	0.2	1331	226981	6.3	21.5	4	3	-3.67e+0	-3.67e+0
priceS4	4	4	0.6	256	194481	3.3	319.5	4	3	-4.15e+0	-4.09e+0
priceS5	4	5	1.0	243	371293	64.3	155	4	3	-4.47e+0	-4.45e+0
priceS6	4	6	2.0	64	117649	3.7	36.5	4	3	-5.30e+0	-5.15e+0
priceT	4	3	1.0	343	50653	3.7	5	4	2	4.37e+0	4.37e+0
priceT	4	4	2.0	256	130321	21	42.5	4	3	-9.12e-1	-8.99e-1
priceT	4	5	3.5	32	161051	1	0	3	1	-9.33e-1	-9.33e-1
priceT	4	6	5.0	64	262144	1	165	3	2	-9.62e-1	-9.58e-1
priceU	4	6	2.0	64	117649	4.3	7	4	7	-3.45e+0	-3.47e+0
coopeL	2	1	0.1	32	189	1	6	3	2	3.43e-01	3.43e-1
coopeM	2	1	0.1	32	189	2	0	3	1	1.00e+0	1.00e+0

coopeN	2	1	0.1	21	121	1	0	3	1	-2.52e-21	-8.23e-16
leon1	4	1	0.1	11	61	10.7	12	4	3	4.50e-3	4.50e-3
leon2	6	1	0.1	21	121	16.7	18	4	2	4.17e-5	4.17e-5
leon3	6	1	0.1	11	61	13.3	18	4	3	5.08e-4	5.21e-4
leon4	7	1	0.1	21	121	21.3	26	4	3	2.58e-3	2.60e-3
leon5	8	1	0.1	41	241	20.7	32	4	3	1.42e-2	1.43e-3
leon6	5	1	0.1	11	61	12.7	16	4	3	1.54e-4	1.55e-4
leon7	5	1	0.1	11	61	12.7	15	4	3	2.09e-3	2.10e-3
leon8	7	1	0.1	41	241	19.3	28	4	3	5.42e-2	5.42e-2
leon9	7	1	0.1	61	361	16.7	34	4	3	1.63e-1	1.63e-1
leon10	3	1	0.1	21	121	7.3	13	4	3	5.38e-1	5.38e-1
leon11	3	1	0.1	11	61	7.3	10	4	2	4.84e-2	4.84e-2
leon12	2	1	0.1	63	377	2.7	8.5	4	3	-1.00e+0	-1.00e+0
leon13	2	1	0.1	11	61	2.7	3.5	4	3	2.36e-1	2.36e-1
leon14	2	1	0.1	11	61	2.7	4.5	4	3	6.65e-1	6.67e-1
leon15	2	1	0.1	11	61	2.7	9	4	3	-6.67e-1	-6.67e-1
leon16	3	1	0.1	11	61	3.7	19	4	2	1.73e+0	1.73e+0
leon17	3	1	0.1	11	61	1	0	3	1	-2.00e+0	-2.00e+0
ferris1	7	1	0.1	11	61	19.3	22	4	3	4.72e-4	4.88e-4
ferris2	7	1	0.1	11	61	9.0	41	4	3	-1.79e+0	-1.79e+0
lin1	6	2	0.1	121	3721	23.8	29	6	3	-1.82e+0	-1.82e+0
liu1	2	1	0.1	21	121	1	0	3	1	-4.65e+0	-4.65e+0
liu2	2	1	0.1	16	95	2.7	14.5	4	3	-3.36e+0	-3.36e+0
liu3	16	1	0.1	51	301	9.3	131	4	3	1.54e+2	1.54e+2
fang1	50	1	0.1	11	61	1.7	8.5	4	3	4.79e-1	4.79e-1
fang2	50	1	0.1	11	61	14.3	21	4	2	6.93e-1	6.93e-1
fang3	50	1	0.1	11	61	16	0	3	1	1.72e+0	1.72e+0
elke1	9	1	0.01	100	600	150	545	7	3	1.08e+0	1.08e+0
elke3	9	1	0.01	100	600	116.7	540	4	3	1.52e+0	1.52e+0
elke4	9	1	0.01	100	600	50	606	5	3	2.50e+0	2.50e+0
elke5	9	1	0.01	100	600	130	675	4	3	2.40e+0	2.40e+0
elke6	9	1	0.01	100	600	122.5	755	5	3	4.49e+0	4.52e+0
elke7	9	1	0.01	100	600	130	455	5	3	2.91e+0	2.91e+0
li1	10	1	0.5	401	2401	43.3	1601	12	3	1.76e+5	2.15e+5
li2	6	1	0.5	401	2401	29.9	816.5	9	3	3.32e+4	3.08e+4

Table 1 - Numerical results for the modified discretization algorithms

Problem	d	nx	AC	ACH	N	NH	fx
hettich6	2	7	26.4	24	6	5	2.80e-2
hettich6	3	11	40.6	38	7	8	3.47e-3
hettich6	4	16	99.3	88	7	12	6.96e-4
hettich6	5	22	176	90	9	13	1.62e-4
hettich6	6	29	473.3	140	7	15	3.96e-5
hettich6	7	37	1072	221	6	12	1.00e-5
hettich7	2	7	24	23	6	6	1.78e-1
hettich7	3	11	36.3	33	7	5	3.66e-2
hettich7	4	16	65.5	56	9	11	4.67e-3
hettich7	5	22	88.2	79	11	11	7.38e-4
hettich7	6	29	160.9	108	8	12	7.67e-5
hettich7	7	37	384.4	145	6	10	8.79e-6

Table 2 - Comparison with Hettich first results

In [4] Hettich presented some numerical results of Algorithm 3.1. The algorithm was applied to two Chebyshev approximation examples based on two variables polynomials as approximating functions (named hettich6 and hettich7 in SIPAMPL database). Chebyshev approximation problems are linear SIP problems, and the corresponding sub-problems are linear as well. The linear sub-problem solver is a primal-dual simplex method. The data reported from Hettich ([4]) and from the adapted method implemented herein are presented in Table 2. For these runs we use identical parameter values as in the Hettich paper:  $h = (h_1, h_2) = (0.1, 0.15)$  with three refinements, resulting in  $IG = 121$  and  $FG = 32761$ . As Hettich did not mention any initial guess for these problems, the procedure described in Section 4 was used to obtain an initial guess. In the sequence of Table 1 notation, in Table 2:

- AC is the average number of constraints in the NLP, without the constraints for the first grid
- N is the number of NLP sub-problems solved by the modified implementation of Section 4
- d is the approximating polynomial degree. The number of finite variables,  $x$ , depends on the degree of the polynomials.

Algorithm 3.2 was proposed by Reemtsen ([8]) to be used for general nonlinear semi-infinite programming problems. In spite of this, Reemtsen numerical results in [8] were presented only from Chebyshev approximation problems (coded as reemtsen1 to reemtsen5 in SIPAMPL database).

Since Chebyshev approximation problems are linear semi-infinite programming problems, Reemtsen used the quadratic programming DQPROG routine of the IMSL library to solve the sub-problems and no initial guesses were included. The results obtained by Reemtsen and the ones given by the modified algorithm implemented herein are collected in Table 3. Identical parameter values are used:  $\tilde{\delta}_i = 0$ , for all  $i$  and the refinements are  $h^{k+1} = h^k/2$ . Once again the notation is extended:

- d/t is a parameter in the problem (changes the number of  $x$  variables)
- $\varepsilon$  is the initial  $\tilde{\varepsilon}_k$
- l is the number of grids used ( $r + 1$ )

Problem	d/t	nx	h	$\varepsilon$	IG	FG	l	AC	ACR	N	NR	fx
reemtsen1	2	11	0.2	0.02	216	68921	3	48.7	24	4	4	1.52e-1
reemtsen1	3	21	0.2	0.03	216	68921	3	79.7	40	7	7	3.11e-2
reemtsen1	4	36	0.2	0.03	216	68921	3	129.6	67	12	12	4.88e-3
reemtsen1	5	57	0.2	0.04	216	68921	3	219.3	102	10	11	7.09e-4
reemtsen2	2	10	2/3	0.01	16	591361	8	20.3	11	7	12	5.84e-2
reemtsen3	2	10	2/3	0.01	16	591361	8	22	16	7	9	7.35e-1
reemtsen4	2	10	2/3	0.01	16	591361	8	18.5	9	9	6	1.14e-2
reemtsen4	3	17	2/7	0.01	64	201601	8	132.6	15	14	6	2.75e-3
reemtsen4	4	26	2/7	0.02	64	201601	8	817.4	36	20	13	7.40e-4
reemtsen4	5	37	2/7	0.02	64	201601	8	24	73	5	13	2.13e-4
reemtsen5	2	11	0.25	0.01	125	274625	4	137.5	74	5	5	8.89e-2
reemtsen5	3	21	0.25	0.01	125	274625	4	269.6	59	15	12	4.81e-2

Table 3 - Comparison with Reemtsen first results

## 7. Conclusions and Future Work

The implemented methods proved to be robust. All problems (linear and nonlinear), except one (elke2 in SIPAMPL database), were solved and no major problems were encountered. Problems in Table 1 were solved in a small amount of time (less than 30 seconds).

The nonlinear Reemtsen version of a discretization algorithm solves in general less sub-problems than the nonlinear Hettich version. Hettich version always solves at least one sub-problem in a grid. In Reemtsen version when the solution to the first sub-problem solves the following grids the algorithm will not solve another sub-problem. This explains the zeros in ACR column of Table 1.

The final performance of the algorithms depends on the performance of the method used to solve the sub-problems. The numerical results published by Hettich and Reemtsen (in Tables 2 and 3) were for linear/quadratic SIP and involve linear/quadratic sub-problems programming. The nonlinear modified Section 4 implementations yield worst results, as expected, in the sense

that a larger average number of constraints is encountered than with Hettich/Reemtsen algorithms. Even so the efficiency of the NLP solver gives, now and then, a smaller number of finite sub-problems than Hettich/Reemtsen implementations.

In all discretization methods an exact (or near exact) SIP solution is not guaranteed, but a solution on the finest grid is obtained.

This suggests that these methods could be used as an initial procedure in a SIP method with a local and faster convergence. The active points in the final grid (points such that  $g_i(x^*, t) = 0$  for any  $i = 1, \dots, m$ ) could be used as initial guesses for the local maxima in a reduction type method.

## References

- [1] Fourer, R., Gay, D.; and Kernighan, B.W., A modeling language for mathematical programming, *Management Science* 36 (1990) 519-554.
- [2] Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H., User's guide for NPSOL: A fortran package for nonlinear programming, Stanford University (1986).
- [3] Haaren-Retagne, E., A semi-infinite programming algorithm for robot trajectory planning, Ph.D. thesis, University of Trier (1992).
- [4] Hettich, R., An implementation of a discretization method for semi-infinite programming, *Mathematical Programming* 34 (1986) 354-361.
- [5] Hettich, R. and Gramlich, G., A note on an implementation of a method for quadratic semi-infinite programming, *Mathematical Programming* 46 (1990) 249-254.
- [6] Hettich, R. and Kortanek, K.O., *Semi-infinite programming: Theory, methods, and applications*, SIAM Review 35 (1993) 380-429.
- [7] Li, Y. and Wang, D., A semi-infinite programming model for Earliness/Tardiness production planning with simulated annealing, *Mathl. Comput. Modelling* 26 (1997) 35-42.
- [8] Reemtsen, R., Discretization methods for the solution of semi-infinite programming problems, *Journal of Optimization Theory and Applications* 71 (1991) 85-103.
- [9] Vaz, A.I.F., Fernandes, E.M.G.P. and Gomes, M.P.S.F., SIPAMPL: Semi-infinite programming with AMPL, Technical report ALG/EF/1-2000, Universidade do Minho, Braga, Portugal, April 2000 (to be published).
- [10] Wang, D. and Fang, S.-C., A semi-infinite programming model for Earliness/Tardiness production planning with a genetic algorithm, *Computers and Mathematics with Applications* 31 (1996) 95-106.

# A NEW IMPROVEMENT FOR A K SHORTEST PATHS ALGORITHM<sup>1</sup>

**Enesto de Queirós Vieira Martins†**

Departamento de Matemática  
FCT-Univ. Coimbra  
3001-454 Coimbra-Portugal

**Marta Margarida Braz Pascoal**

**José Luis Esteves dos Santos**

Centro de Informática e Sistemas  
e  
Departamento de Matemática  
FCT-Univ. Coimbra  
3001-454 Coimbra-Portugal

## Abstract

The K shortest paths problem is a well known network optimization problem where it is intended to rank the K shortest paths between an initial and a terminal node in a network. The first algorithm for solving this problem appeared by the fifties and since then several other algorithms have been proposed. These algorithms can be divided into two classes: one based on the Optimality Principle and another based on the determination of a tree of shortest paths. Moreover, in the first of these classes can be considered labeling algorithms and deletion path algorithms.

In this paper an improvement for a known deletion path algorithm is presented which results in the improvement of its running time complexity when the worst case analysis is considered, from  $O(Km)$  to  $O(Kn \log n)$ . Comparative computational experiments regarding the proposed improvement and its version are also reported, allowing possible conclusions about the obtained performances when the average case is considered.

## Keywords

path, ranking, deletion path algorithms.

## 1. Introduction

The K shortest paths problem is a well known network optimization problem where it is intended to rank the K shortest paths between an initial and a terminal node in a network. The first algorithm for solving this problem was proposed by Hoffman and Pavley in 1959, [9], and ever since other papers presenting algorithms have been proposed among which we refer

---

<sup>1</sup> The research of Marta Pascoal and José Santos was developed within CISUC and partially supported by the portuguese Ministry of Science and Technology (MCT), under PRAXIS XXI Project of JNICT.

† Sadly, the author passed away in November, 2000.

[6, 7, 10, 11, 13, 16, 17]. A very complete bibliography on the problem, due to Eppstein, can be found in the URL address <http://liinwww.ira.uka.de/bibliography/Theory/k-path.html>.

These algorithms can be divided into two classes: one based on the Optimality Principle generalization and another on the determination of a tree formed by shortest paths, [14, 15]. Moreover, among the algorithms based on the Optimality Principle there can be considered labeling algorithms and deletion path algorithms (also known as algorithms of alternatives).

In this paper an improvement for a known deletion path algorithm is presented resulting in a decrease of its running time complexity when the worst case analysis is considered from  $O(Km)$  to  $O(Kn \log n)$ . Comparative computational experiments are also reported, allowing possible conclusions about the obtained performances when the average case is considered.

The reader is assumed to be familiarized with some basic network optimization concepts and, mainly, with the deletion path algorithm known as MS algorithm, [16]. Nevertheless, in Sections 2 and 3 some of the notation and definitions used in this paper are introduced and the problem is described. Section 4 contains a brief overview on deletion path algorithms being the improvement of MS algorithm, [16], described in the following one. Finally, in Section 6 the computational complexity of MS algorithm improvement is studied and some comparative numerical results with its previous version are presented.

## 2. Notation and Definitions

Let  $(N,A)$  denote a given network, where  $N = \{v_1, \dots, v_n\}$  is a finite set with  $n$  elements called nodes and  $A = \{a_1, \dots, a_m\} \subseteq N \times N$  is a finite set with  $m$  elements called arcs. Sometimes it will simply be used  $i$  to represent node  $v_i$ . Each arc  $a_k \in A$  can also be identified by a pair  $(i,j)$ , where  $i,j \in N$ .

Let  $i$  and  $j$  be two nodes of  $(N,A)$ ; if all pairs  $(i,j)$  in  $(N,A)$  are ordered, i.e., if all the arcs of the network are directed,  $(N,A)$  is said to be a directed network; if all the pairs  $(i,j)$  are not ordered, then  $(N,A)$  is said to be an undirected network. Without loss of generality, in what follows it will be considered that  $(N,A)$  is a directed network, since each undirected arc can be replaced by two arcs with opposite direction. It will also be assumed that there is at most a single arc between each pair of nodes of the network and there are no arcs of the form  $(i,i)$ , where  $i \in N$ .

Given  $i,j \in N$ , a path from node  $i$  to node  $j$  in  $(N,A)$  is an alternating sequence of nodes and arcs, of the form  $p = \langle i = v_1', a_1', v_2', \dots, a_{\ell-1}', v_{\ell}' = j \rangle$ , where:

- $v_k' \in N$ , for every  $k \in \{1, \dots, \ell\}$ ;
- $a_k' \equiv (v_k', v_{k+1}') \in A$ , for every  $k \in \{1, \dots, \ell-1\}$ .

In order to simplify the notation a path will be represented only by its nodes; that is,  $p = \langle v_1', v_2', \dots, v_{\ell}' \rangle$  and for convenience, sometimes a single node is viewed as a path, the null path. The set of paths defined from  $i$  to  $j$  in  $(N,A)$  will be denoted by  $P_{ij}$ . A loopless path from  $i$  to  $j$  is a path from  $i$  to  $j$  where all the nodes are different and a loop (or cycle) is a path from some node to itself, where all nodes are different except the first one which is also the last.

Given  $x$  and  $y$ , two nodes of a path  $p$ , the subpath of  $p$  from  $x$  to  $y$ , denoted by  $\text{sub}_p(x,y)$ , is simply the path that coincides with  $p$  between these two nodes.

Let  $p \in P_{ix}$  be a path from  $i$  to  $x$  and  $q \in P_{xj}$  a path from  $x$  to  $j$ . The concatenation of  $p$  and  $q$ , denoted by  $p \hat{\diamond} q$ , is a path from  $i$  to  $j$  formed by  $p$  until its node  $x$  and followed by path  $q$  from  $x$  to  $j$ . Given a cycle  $C$  it can be defined the  $k$ -cycle concatenation as  $C^k = C^{k-1} \hat{\diamond} C$ , with  $k \geq 1$  and  $C^0$  a single node of cycle  $C$ .

Let  $c_{ij}$  be a real value associated with arc  $(i,j)$  of  $(N,A)$ , known as the  $(i,j)$  arc cost, and let  $c(p) = \sum_{(i,j) \in p} c_{ij}$  (or, simply,  $c(p) = \sum_p c_{ij}$ ), for a given path  $p$  in  $(N,A)$ , be the  $p$  path cost.

Let  $s$  and  $t$  be two different nodes of  $(N,A)$ , called initial and terminal node, respectively. In order to simplify,  $P$  will denote the set  $P_{st}$ .

In what follows, with no loss of generality, it will assumed that  $P_{si} \neq \phi$  and  $P_{it} \neq \phi$  holds, for every node  $i$  in  $(N,A)$ . It will also be assumed that there are no arcs of the form  $(x,s)$  and  $(t,x)$ , where  $x \in N$ . This is assumed with no loss of generality, since a new initial node  $S$ , a new terminal node  $T$ , and zero the cost arcs  $(S,s)$  and  $(t,T)$  can be added to the network. These new nodes are usually known as super-source and super-sink and there can be easily defined a bijection between  $P_{st}$  and  $P_{ST}$ . As a consequence, nodes  $s$  and  $t$  can be repeated in the path definition and each path has at least three arcs which is important in the proof of some of the theorems presented in what follows.

A path is said to be finite if it is a finite sequence of nodes (and arcs).

### 3. The K Shortest Paths Problem

#### 3.1 The problem

In the classical shortest path problem it is intended to determine a path  $p^*$  from  $s$  to  $t$  in  $(N,A)$  with minimum cost, that is, it is intended to determine  $p^* \in P$  such that  $c(p^*) \leq c(p)$ , for any  $p \in P$ . Given an integer  $K > 1$ , the  $K$  shortest paths problem can be considered a generalization of the previous one where, beyond the determination of the shortest path in  $P$  it also have to be determined the second shortest path in  $P, \dots$ , until the  $K$ -th shortest path in  $P$ . That is, denoting by  $p_i$ ,  $i \in N$ , the  $i$ -th shortest path from  $s$  to  $t$  in  $(N,A)$  it is intended to determine a set of paths,  $P_K = \{p_1, \dots, p_K\} \subseteq P$ , such that:

1.  $p_i$  is determined before  $p_{i+1}$ , for any  $i = 1, \dots, K - 1$ ;
2.  $c(p_i) \leq c(p_{i+1})$ , for any  $i = 1, \dots, K - 1$ ;
3.  $c(p_K) \leq c(p)$ , for any  $p \in P - P_K$ .

#### 3.2 Finiteness, boundness and Optimality Principle

As it is well known, when studying optimal problems, two very important concepts such as finiteness and boundness, have to be studied too, [12]. That is, conditions that assure the existence of a finite and bounded optimal solution have to be established, since only in this case it is possible to compute its optimal solution with an algorithm. Note that both finiteness

and boundness have to be studied jointly since optimal path problems exist where a non finite optimal solution is bounded, [12].

Considering the K shortest paths problem as a generalization of the classical shortest path problem these concepts should also be generalizations of the known ones for the  $K = 1$  problem. Since these notions and related results are already known they will only be briefly referred. For further details the reader may consult [12, 15].

A K shortest paths problem is said to be finite if and only if there exists a finite K-th shortest path and it is said to be bounded if and only if this path has a finite cost. It can be proved that a K shortest paths problem is finite if and only if no cycle in the given network has negative cost and also that it is bounded under the same conditions.

The labeling algorithm for any optimal path problem, which may assume several forms, is supported by the Optimality Principle, [12]. The Optimality Principle for the general problem states that: *Every subpath, from  $s$  to  $y$ , of  $p_k$ , the  $k$ -th shortest path from  $s$  to  $t$  in  $(N,A)$ , is a  $j_y$ -th shortest path from  $s$  to  $y$  in  $(N,A)$ , for some  $1 \leq j_y \leq k$ .*

It can be proved that a K shortest paths problem satisfies the Optimality Principle if and only if there are no cycles with negative cost in the considered network.

Therefore, in order to assure finiteness and boundness of the problem, as well as the Optimality Principle being verified, from now on it will be assumed that  $c(C) = \sum_C c_{ij} \geq 0$  for any cycle  $C$  in the network; that is, all the cycles in the considered network have non negative cost.

#### 4. Deletion Path Algorithms

The first of a sequence of algorithms for ranking shortest paths was proposed by one of the authors in 1984, [11]. This algorithm is based on the successive deletion of paths from a network and it is supported by the Optimality Principle.

This first algorithm can be roughly described considering that the second shortest path from  $s$  to  $t$  in  $(N,A)$  is the shortest path from  $s$  to  $t$  in the network, as long as path  $p_1$  no longer exists in the initial one. Repeating this procedure for  $k > 1$ , the  $k$ -th shortest path from  $s$  to  $t$  in  $(N,A)$  is the shortest path from  $s$  to  $t$  in the network, as long as none of the  $k-1$  shortest paths, that is,  $p_1, \dots, p_{k-1}$ , are allowed to exist in the network. So, denoting by  $(N_1, A_1)$  the given network and by  $(N_k, A_k)$  the network which is obtained from  $(N,A)$  in such a way that its set of paths from  $s$  to  $t$  is  $P - \{p_1, \dots, p_{k-1}\}$ , this algorithm can be sketched as presented below:

**Algorithm 1** - Deletion path algorithm

$(N_1, A_1) \leftarrow (N, A)$ ;

$p_1 \leftarrow$  shortest path from  $s$  to  $t$  in  $(N_1, A_1)$ ;

For  $k \in \{2, \dots, K\}$  Do

$(N_k, A_k) \leftarrow$  Delete  $p_{k-1}$  from  $(N_{k-1}, A_{k-1})$ ;

$p'_k \leftarrow$  shortest path from  $s$  to  $t$  in  $(N_k, A_k)$ ;

$p_k \leftarrow$  path in  $(N,A)$  corresponding to  $p'_k$   
 EndFor

It should be noticed that an easy correspondence may have to be established in order that some path  $p'$  in  $(N_k, A_k)$  can be identified with a path  $p$  in  $(N,A)$ .

The key to this algorithm is the deletion of a path from a network. Shortly it can be said that this deletion is obtained by creating alternative nodes (that is, copies of these nodes) of the path to be removed. However, only some of the arcs related with these nodes are copied, in order that it is not possible to follow that path in the new network. The new computed paths, that is, the determined shortest paths, for instance  $p'$ , correspond to a path  $p$  in the initial network,  $(N,A)$ . Path  $p'$  is said to be an alternative for  $p$  and it may happen that these two paths are misled. The procedure for deleting a path in a network is presented in Procedure 1.1.

**Procedure 1.1** - Delete  $p = \langle s = v_1, v_2, \dots, v_\ell = t \rangle$  from  $(N,A)$  - Martin's algorithm

$(N', A') \leftarrow (N,A);$   
 $N' \leftarrow N' \cup \{v'_2\};$   
 For  $(i, v_2) \in A$  such that  $i \neq s$  Do  
    $A' \leftarrow A' \cup \{(i, v'_2)\}; c_{iv'_2} \leftarrow c_{iv_2}$   
 EndFor  
 For  $j \in \{3, \dots, \ell\}$  Do  
    $N' \leftarrow N' \cup \{v'_j\};$   
    $A' \leftarrow A' \cup \{(v'_{j-1}, v'_j)\}; c_{v'_{j-1}v'_j} \leftarrow c_{v_{j-1}v_j};$   
   For  $(i, v_j) \in A$  such that  $i \neq v_{j-1}$  Do  
      $A' \leftarrow A' \cup \{(i, v'_j)\}; c_{iv'_j} \leftarrow c_{iv_j}$   
 EndFor  
 EndFor

Recall that it was previously assumed, with no loss of generality, that  $\ell \geq 3$ . As it has already been mentioned, this is, in short, the former version of deletion path algorithms. However it is not very efficient, neither in the necessary memory space (since several similar copies of the same nodes and of the same arcs have to be done) nor in the running time, since K shortest path problems have to be solved in networks whose number of nodes and number of arcs is growing up from one network to the following one.

The last improvement to this algorithm, known as MS algorithm, will now be described. In this MS algorithm the number of arcs in the network is constant and therefore the needed memory space decreases from previous versions. In practice, it happens the same with the running time.

The main MS algorithm idea is still the copy of the successive  $p_k$  paths nodes, for a later computation of paths  $p_j$ , with  $j > k$ . In what follows it will be tried to roughly describe the

several improvements to Martin's algorithm. For more complete details the reader is addressed to [1, 2, 3, 16].

As presented in Algorithm 1, for each one of the determined  $p_k$  paths, each of its nodes is analysed and copied. The copy of node  $i \in N_k$  corresponds to a  $j$ -th shortest path from  $s$  to  $i$ , for some  $1 \leq j \leq k$ . This copy consists in a new node, somehow related with the corresponding node in the original network, and new arcs, each one associated to an arc which incides in the node corresponding to  $i$  in the original network.

It can be easily proved that, when determining the shortest path from  $s$  to  $t$  in the networks generated by the algorithm, each node label is permanent. So, in each step of the algorithm, labeling alternative nodes for every  $k$ -th shortest path can be made, instead of solving shortest paths problems. The label of node  $i \in N_k$ , for some  $k \geq 1$ , that is, the cost of the shortest path from the initial node until node  $i$  will be denoted by  $\pi_i$ .

It should be pointed out that given some node  $v_i$  of a path  $p_k$ , if  $v_i$  precedes  $v_\alpha$ , the first node of  $p_k$  where more than one arc incides, then there is no alternative path from  $s$  to  $v_i$  for sub  $p_k(s, v_i)$ . Obviously, there is an alternative for  $v_\alpha$  and for every node following  $v_\alpha$  until  $t$ . Therefore it is enough to analyse the nodes of  $p_k$  following the first one in which more than one arc incides and to which no alternative has been determined yet. Thus the number of copies of nodes is reduced and the algorithm running time is also decreased, since less labels have to be computed (and recomputed). Let  $v_\beta$  be the first of these nodes for which an alternative has not been determined yet. If  $v_\beta$  coincides with  $v_\alpha$ , then it will be the first node to analyse; otherwise it will be analysed as an intermediate node.

The last improvement, the so called MS algorithm, appears by remarking that the set of the arcs inciding in the new alternative nodes is similar to the same set for the corresponding nodes in the previous network whose label is permanent and it is used only once in the determination of the label of its "copy" node. Thus, the replication of these arcs can be avoided since it is enough to associate the information referring the original node to each new node.

The MS algorithm is presented in Procedure 1.2.

**Procedure 1.2** - Delete  $p = \langle s = v_1, v_2, \dots, v_k = t \rangle$  from  $(N, A)$  - MS algorithm

$v_\alpha \leftarrow$  first node of  $p$  in which incides more than one arc;

If ( $v_\alpha$  is defined) Then

$v_\beta \leftarrow$  first node of  $p$  from  $v_\alpha$  for which no alternative has been determined yet;

$(N', A') \leftarrow (N, A)$ ;

$\beta \leftarrow v_\beta$ ;

If ( $v_\alpha = v_\beta$ ) Then

$N' \leftarrow N' \cup \{v_\beta\}$ ;

$A' \leftarrow A' - \{(v_{\beta-1}, v_\beta)\}$ ;

$\pi_{v_\beta} \leftarrow \min\{\pi_i + c_{iv_\beta} : (i, v_\beta) \in A'\}$ ;

```

     $\gamma \leftarrow \beta + 1$ 
  EndIf
  For  $j \in \{\gamma, \dots, \ell\}$  Do
     $N' \leftarrow N' \cup \{v_j\}$ ;
     $A' \leftarrow A' - \{(v_{j-1}, v_j)\} \cup \{(v_{j-1}', v_j)\}$ ;  $c_{v_{j-1}'v_j} \leftarrow c_{v_{j-1}v_j}$ ;
     $\pi_{v_j} \leftarrow \min\{\pi_i + c_{iv_j} : (i, v_j) \in A'\}$ 
  EndFor
Else
  All paths from  $s$  to  $t$  have been determined
EndIf

```

### 5. Improved Algorithm

In this section an improvement of MS algorithm, described in Algorithm 1 and Procedure 1.2, is presented. In this improvement a particular order of the set of arcs inciding in each node of the network is used. This order decreases the number of the performed arithmetic operations, improving not only the theoretical computational complexity of the algorithm but also its practical efficiency. In what follows the tree of the shortest paths from  $s$  to  $i$ , for any  $i \in N$  will be denoted by  $\mathcal{T}_s$ .

Recalling MS algorithm, whenever a path  $p_k$  is determined, the nodes following a well determined node  $x$  have to be analysed, that is, some nodes have to be "copied" and "labeled". So, for such a node  $v_i$ , another node  $v_i'$ , has to be created, being  $\pi_{v_i'} = \min\{\pi_j + c_{jv_i} : (j, v_i) \in A_k\}$  its label. However this labeling procedure demands an exhaustive search over the set of arcs whose head node is  $v_i$ ; that is, all the arcs inciding in  $v_i$  have to be analysed. The technique that will be presented allows that each labeling is an immediate operation, improving the complexity of the previous algorithm.

From MS algorithm it can be concluded that given a node  $v_i \in p_k$  to be copied, its alternative is labeled from another node  $u$  such that  $(u, v_i) \in A_k$  and  $\pi_u + c_{uv_i} \leq \pi_v + c_{vv_i}$ , for any  $(v, v_i) \in A_k$ . Consequently, for any  $k \geq 1$ , let us consider  $A$  sorted as follows. For any  $(i_1, j_1), (i_2, j_2) \in A_k$ .

$$\left\{ \begin{array}{l} j_1 < j_2 \\ \text{or} \\ j_1 = j_2 \text{ and } \left\{ \begin{array}{l} \pi_{i_1} + c_{i_1j_1} < \pi_{i_2} + c_{i_2j_2} \\ \text{or} \\ \pi_{i_1} + c_{i_1j_1} = \pi_{i_2} + c_{i_2j_2} \text{ and } (i_1, j_1) \in \mathcal{T}_s \end{array} \right. \end{array} \right. \Rightarrow (i_1, j_1) \text{ " < " } (i_2, j_2). \quad (1)$$

For arcs  $(i_1, j), (i_2, j) \in A_k$ , inciding in the same node, such that  $\pi_{i_1} + c_{i_1j} = \pi_{i_2} + c_{i_2j}$  and  $(i_1, j), (i_2, j) \notin \mathcal{T}_s$  the used order is not important. It can then be stated that under the

conditions of relation (1),  $v_i'$  is labeled from a node  $u$  such that  $(u, v_i)$  follows the arc  $(v_{i-1}, v_i)$ , preceding  $v_i$  in path  $p_k$  in the ordered set  $A_k$ ; that is,  $u$  is the tail node of the arc inciding in  $v_i$  following  $(v_{i-1}, v_i)$  in the set  $A_k$  ordered according to (1). Therefore, the set of arcs is ordered by non decreasing order of the head node of each arc, as in the reverse star form, [4], and each set of arcs inciding in the same node will be ordered by non decreasing order of the "label" that they can generate; that is, the set of arcs  $(i, j)$  inciding in node  $j \in N$ , will be sorted according to the respective value of  $\pi_i + c_{ij}$ . In what follows this sort will be denominated the dynamic sorted reverse star form. A similar sort was used in the algorithm of Jiménez and Varó, [10].

Nevertheless, some changes have to be taken into account, when obtaining the sequence of networks  $\{(N_i, A_i)\}_{i=1}^k$ , since they can affect the order of the elements in the set of arcs. It is known that creating node  $v_i'$ , the copy of  $v_i$ , can imply deleting the arc preceding  $v_i$  in the analysed path  $(v_{i-1}, v_i)$ , which does not change the order in the network; however, for some of the created alternatives,  $(v_{i-1}, v_i)$  is replaced by  $(v_{i-1}', v_i)$ , which has the same cost of  $(v_{i-1}, v_i)$ . Since the label of  $v_{i-1}'$  was computed in the previous step and in general,  $\pi_{v_{i-1}'} > \pi_{v_{i-1}}$ ; therefore  $\pi_{v_{i-1}'} + c_{v_{i-1}'v_i} > \pi_{v_{i-1}} + c_{v_{i-1}v_i}$ . Thus, when replacing arc  $(v_{i-1}, v_i)$  it is necessary to insert  $(v_{i-1}', v_i)$  in the set of arcs whose head node is  $v_i$ , in such a way that the dynamic sorted reverse star form is maintained, that is, in order that relation (1) is satisfied.

This improvement is presented in Algorithm 2 and in Procedure 2.1.

### Algorithm 2 - Improved algorithm

```

 $(N_1, A_1) \leftarrow (N, A);$ 
 $\mathcal{T}_s \leftarrow$  tree of the shortest paths from  $s$  to  $i$ , for any  $i \in N$ ;
 $p_1 \leftarrow$  path from  $s$  to  $t$  in  $\mathcal{T}_s$ ;          /* shortest path from  $s$  to  $t$  in  $(N_1, A_1)$  */
Define  $(N_1, A_1)$  in the dynamic sorted reverse star form;
For  $k \in \{2, \dots, K\}$  Do
     $(N_k, A_k) \leftarrow$  Delete  $p_{k-1}$  from  $(N_{k-1}, A_{k-1})$ ;
     $p_k' \leftarrow$  shortest path from  $s$  to  $t$  in  $(N_k, A_k)$ ;
     $p_k \leftarrow$  path in  $(N, A)$  corresponding to  $p_k'$ 
EndFor

```

### Procedure 2.1 - Delete $p = \langle s = v_1, v_2, \dots, v_\ell = t \rangle$ from $(N, A)$ - Improved algorithm

```

 $v_\alpha \leftarrow$  first node in  $p$  where more than one arc incides;
If  $(v_\alpha$  is defined) Then
     $v_\beta \leftarrow$  first node of  $p$  from  $v_\alpha$  for which no alternative has been determined yet;
     $(N', A') \leftarrow (N, A)$ ;
     $\mathcal{I} \leftarrow \beta$ ;
    If  $(v_\alpha = v_\beta)$  Then
         $N' \leftarrow N' \cup \{v_\beta'\}$ ;

```

```

 $A' \leftarrow A' - \{(v_{\beta-1}, v_{\beta})\};$ 
 $(i, v_{\beta}) \leftarrow \text{first arc in } A' \text{ inciding in } v_{\beta};$ 
 $\pi_{v_{\beta}} \leftarrow \pi_i + c_{iv_{\beta}};$ 
 $\mathcal{I} \leftarrow \beta + 1$ 
EndIf
For  $j \in \{\mathcal{I}, \dots, \ell\}$  Do
 $N' \leftarrow N' \cup \{v'_j\};$ 
 $A' \leftarrow A' - \{(v_{j-1}, v_j)\} \cup \{(v'_{j-1}, v_j)\}; c_{v'_{j-1}v_j} \leftarrow c_{v_{j-1}v_j};$ 
Insert  $(v'_{j-1}, v_j)$  in  $A'$  in order, satisfying (1);
 $(i, v_j) \leftarrow \text{first arc in } A' \text{ inciding in } v_j;$ 
 $\pi_{v_j} \leftarrow \pi_i + c_{iv_j}$ 
EndFor
Else
All paths from  $s$  to  $t$  have been determined
EndIf

```

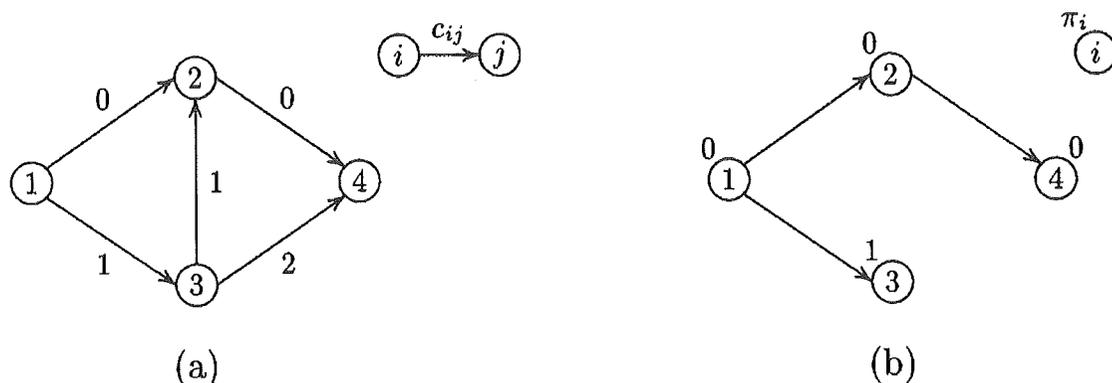


Figure 1 - (a) Network  $(N,A)$ , (b) respective shortest tree rooted at  $s = 1$

$(i,j)$	(1,2)	(3,2)	(1,3)	(2,4)	(3,4)
$\pi_i + c_{ij}$	0	2	1	0	3

Table 1 - The set  $A_1 = A$  sorted according to relation (1)

In what follows, MS algorithm and its improvement will be exemplified using the small network represented in Figure 1(a), where the initial node is  $s = 1$  and the terminal node is  $t = 4$ . Note that neither the super initial nor the super terminal nodes have been included in this network since no arc incides in  $s = 1$  and no arc emerges from  $t = 4$ .

Both algorithms begin by determining the tree of the shortest paths from  $s$  to every node in  $N$ , Figure 1(b), which implies determining  $p_1 = \langle 1,2,4 \rangle$ . After that  $p_1$  is deleted from  $(N,A) = (N_1, A_1)$ , thus determining  $(N_2, A_2)$ , represented in Figure 2, and consequently path  $p_2 = \langle 1,3,2,4 \rangle$ . Note that, the computed path is, in fact,  $\langle 1,3,2',4' \rangle$ , which in the original network

corresponds to the given path,  $p_2$ . In order to determine this path, two new nodes have to be created,  $2'$  and  $4'$ .

For labeling node  $2'$  with MS algorithm

$$\pi_{2'} = \min\{\pi_i + c_{i2} : (i, 2) \in A_2\} = \min\{\pi_i + c_{i2} : (i, 2) \in \{(3, 2)\}\} = 2$$

has to be computed; similarly, for labeling node  $4'$ ,

$$\pi_{4'} = \min\{\pi_i + c_{i4} : (i, 4) \in A_2\} = \min\{\pi_i + c_{i4} : (i, 4) \in \{(2', 4), (3, 4)\}\} = 2$$

has to be computed.

Now, when running MS algorithm improvement, after computing the tree in Figure 1(b) set  $A$  is sorted - see Table 1. Thus, when creating  $2'$ , arc  $(1,2)$  is removed and for labeling  $2'$  arc  $(3,2)$  is chosen since it is the following one which incides in  $2$  (and also the unique in this case) and  $\pi_{2'} = \pi_3 + c_{32} = 2$ . Concerning the creation of  $4'$ , arc  $(2,4)$  is removed and it is replaced by  $(2',4)$ . To maintain the order in  $A_2$ ,  $(2',4)$  must be inserted in the changed set according to relation (1). Since  $\pi_{2'} + c_{2'4} = 2$  and  $\pi_3 + c_{34} = 3$ ,  $(2',4)$  is inserted immediately before  $(3,4)$ . Then, when labeling node  $4'$  arc  $(2',4)$  is chosen, since it is the first arc inciding in node  $4$ , and  $\pi_{4'} = \pi_{2'} + c_{2'4} = 2$ .

So, the second network to be constructed using both algorithms is  $(N_2, A_2)$ , in Figure 2, where  $\xi_i$  denotes the node from which  $i \in N$  has been labeled. However the set of arcs  $A_2$ , obtained when using MS improvement, is sorted in a special way - represented in Table 2.

**6. Computational Experiments**

It will now be studied the behavior of the improved algorithm beginning by looking over its computational complexity. However, rather than the worst case we are interested in studying the performance for the average case, which leads us to present some comparative numerical results obtained by running implementations of MS algorithm and its improvement.

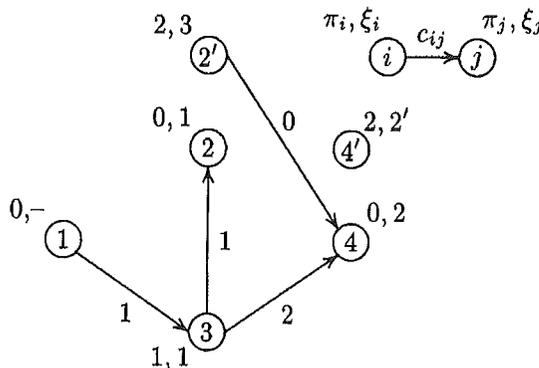


Figure 2 - Network  $(N_2, A_2)$

$(i,j)$	$(3,2)$	$(1,3)$	$(2',4)$	$(3,4)$
$\pi_i + c_{ij}$	2	1	2	3

Table 2 - The set  $A_2$  sorted according to relation (1)

### 6.1 Computational complexity

In MS algorithm improvement, computing the minimum value of  $\pi_i + c_{ij}$  for any arc  $(i,j)$  of the network which incide in node  $j$  is replaced by inserting, in order, an arc in that set, whenever a node is labeled. This operation seldom demands an exhaustive search which happens always with MS algorithm. The computational complexity of this improved algorithm can still be decreased using a suitable data structure for ordered insertions, such as binary heaps. It will now be presented a brief study of the computational complexity for both algorithms when considering the worst case analysis.

The worst case for these two algorithms happens when the shortest path which, in the absence of negative cycles, can be considered a loopless path from  $s$  to  $t$  in  $(N,A)$  contains every node of the network, its cost is zero and besides, there is an arc from  $t$  to  $s$  in the network, also with zero cost. In this case the  $k$ -th shortest path in  $P$  has cost zero and it can be expressed in the form  $p_k = p_1 \diamond C^{k-1}$ , for any  $k \geq 2$ , where  $C = \langle t,s \rangle \diamond p_1$ , that is,  $p_k = p_1 \diamond \underbrace{\langle t,s \rangle \diamond p_1 \dots \diamond \langle t,s \rangle \diamond p_1}_{k-1 \text{ times}}$ . It should be noticed that, even if the existence of cycles  $C$  such that  $c(C) = 0$  is not allowed, in order to avoid solutions of the form  $q_1 \diamond C^k \diamond q_2$ , still an example of a  $k$ -th shortest path with the form given above can be found, (see [14]).

Both algorithms begin by computing the shortest path in the network, which takes  $O(m + n \log n)$  time using heaps and Dijkstras' algorithm, [5,8], considering only non-negative arcs costs. After that, in each step it is necessary to create alternatives for the nodes of  $p_k$ . For MS algorithm the creation of these alternatives requires the analysis of all the arcs inciding in the corresponding node of the initial network. So, in the worst case and for this algorithm, all the arcs in the network have to be analysed and therefore its complexity, in the worst case, is  $O(K m + n \log n)$ .

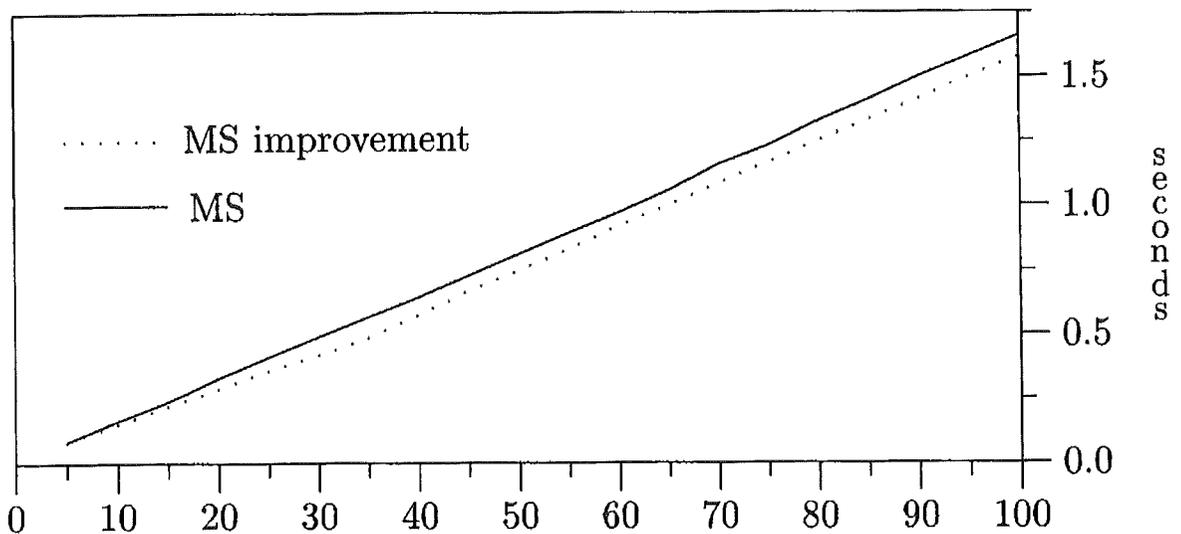


Figure 3 - Random Networks, 10000 nodes and 15000 arcs

For the MS algorithm improvement described in this paper, after determining the shortest path, the set of arcs is sorted. For sorting that set it becomes necessary to sort the set of  $m_i$  arcs which incide in every node  $i$ , which takes  $O(m_i \log m_i)$  using binary heaps. Therefore the running time for this sorting is  $\sum_{i=1}^n m_i \log m_i \leq \sum_{i=1}^n m_i \log n = m \log n$ . After that, it is necessary to sort the set of arcs inciding in every node of each path  $p_k$ , which in the case of complete networks, which is the worst, takes  $O(\log n)$ . It should be noticed that after these sorting every node labeling is immediate. So the running time for this algorithm is  $O(m + K n \log n)$ , or simply  $O(K n \log n)$ , since usually  $K n > m$ .

In what concerns the memory space, if  $p_k$  has also the given form, in MS algorithm the set of arcs always has at most  $m$  elements, and the set of nodes has the original ones and all those created alternatives; that is, the set of nodes has at most  $K n$  elements. So, both MS algorithm and its improvement need  $O(m + K n)$  memory space.

## 6.2 Execution times

As it was said before, besides the worst case it is interesting to analyse the behaviour of the presented algorithm for an average case. From the previous paragraph one would believe that, considering an average case, MS algorithm performance would be much better than its computational complexity. Even so, MS algorithm improvement seems to outperform its previous version, the MS algorithm.

In what follows the implementations, in C language, for the described algorithms are compared. The tests that will be presented ran on a Pentium II, over Linux, with a 266 MHz processor and 64 MB of RAM.

The presented running times are average running times for the resolution of 15 problems for each kind of network. Random networks with 10000 nodes and considering several densities were generated, where the density of a network is the ratio  $d = m/n$ . Density values were small (1.5 and 2, in Figure 3 and Figure 4) and higher (15 and 20, in Figure 5 and Figure 6). Random numbers are generated within a given range, following an uniform distribution.

It should be noticed that MS improvement seems to outperform MS algorithm, specially for high density networks. In fact that is the case in which, for each alternative node created, MS exhaustive search time becomes significant, versus the linear inserting time needed for its improvement.

We must point out the apparent linearity with  $K$  of the running times, although MS improvement presents a lower slope.

The implemented code for MS improvement used linear insertion; so, it is expected that the running times may decrease when using binary heaps. This is intended to be a future work, describing more complete computational experiments, including codes for different algorithms.

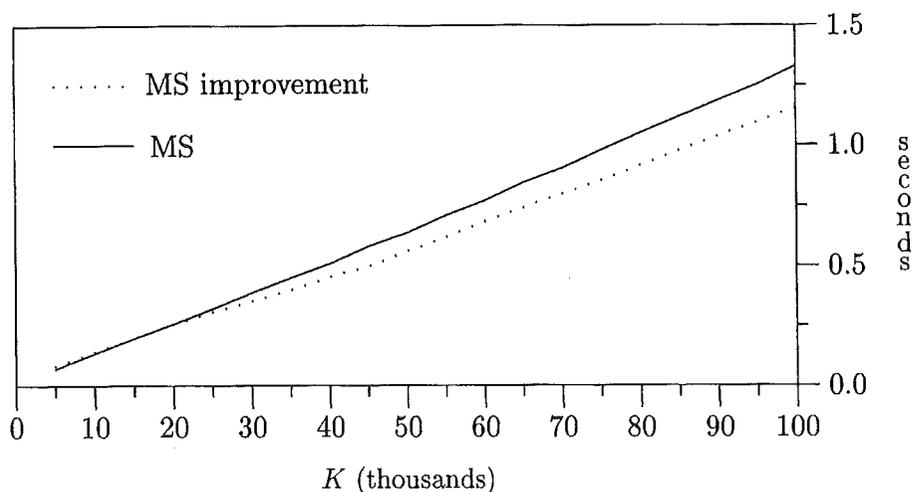


Figure 4 - Random Networks, 10000 nodes and 20000 arcs

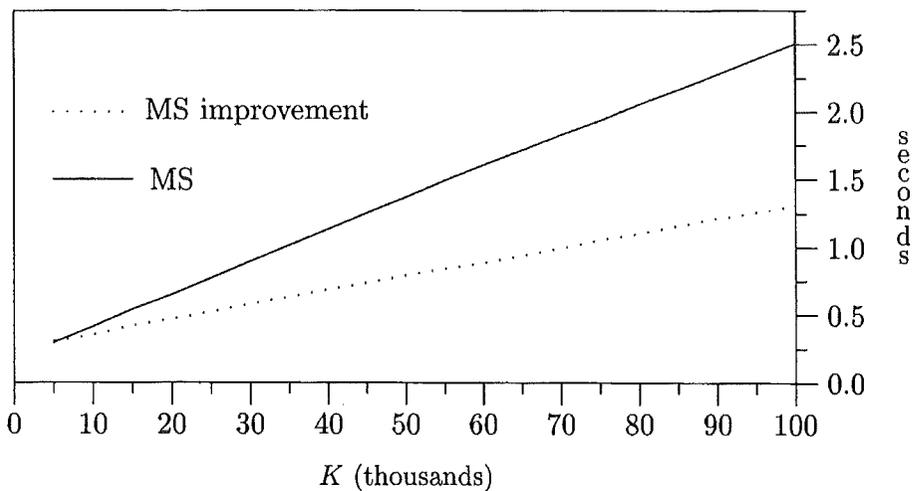


Figure 5 - Random Networks, 10000 nodes and 150000 arcs

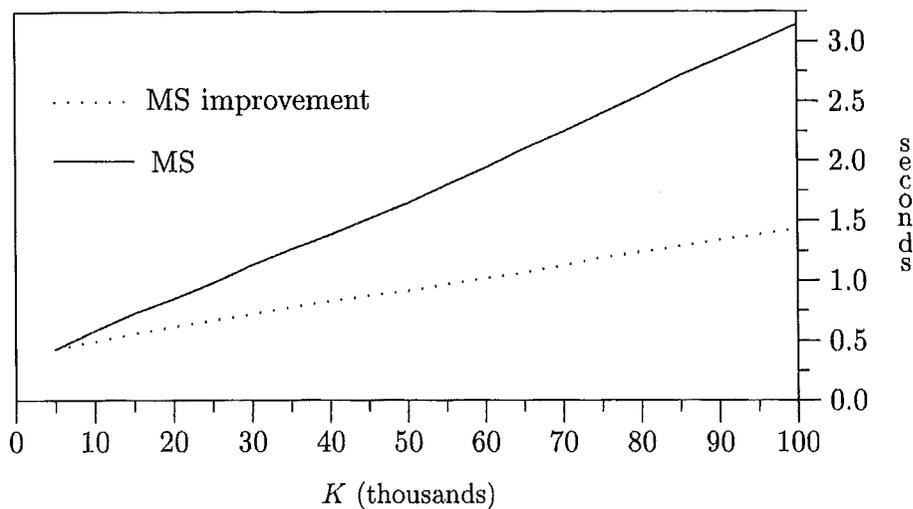


Figure 6 - Random Networks, 10000 nodes and 200000 arcs

## Acknowledgments

The authors acknowledge the referee for his helpful comments on the first version of this paper.

## References

- [1] Azevedo, J.A., Costa, M.E.O.S., Madeira, J.J.E.R.S. and Martins, E.Q.V., An algorithm for the ranking of shortest paths, *European Journal of Operational Research* 69 (1993) 97-106.
- [2] Azevedo, J.A., Madeira, J.J.E.R.S., Martins, E.Q.V. and Pires, F.M.A., A shortest paths ranking algorithm, *Proceedings of the Annual Conference AIRO'90, Models and Methods for Decisions Support*, Operational Research Society of Italy (1990) 1001-1011.
- [3] Azevedo, J.A., Madeira, J.J.E.R.S., Martins, E.Q.V. and Pires, F.M.A., A computational improvement for a shortest paths ranking algorithm, *European Journal of Operational Research* 73 (1994) 188-191.
- [4] Dial, R., Glover, G., Karney, D. and Klingman, D., A computational analysis of alternative algorithms and labelling techniques for finding shortest path trees, *Networks* 9 (1979) 215-348.
- [5] Dijkstra, E., A note on two problems in connection with graphs, *Numerical Mathematics* 1 (1959) 395-412.
- [6] Dreyfus, S.E., An appraisal of some shortest-path algorithms, *Operations Research* 17 (1969) 395-412.
- [7] Eppstein, D., Finding the k shortest paths, *SIAM Journal on Computing* 28 (1998) 652-673.
- [8] Fredman, M.L. and Tarjan, R.E., Fibonacci heaps and their uses in improved network optimization algorithms, *Journal of the Association for Computing Machinery* 34 (1987) 596-615.
- [9] Hoffman, R. and Pavley, R.R., A method for the solution of the n<sup>th</sup> best path problem, *Journal of the Association for Computing Machinery* 6 (1959) 506-514.
- [10] Jiménez, V.M. and Marzal Varó, A., Computing the k shortest paths: a new algorithm and an experimental comparison, In *Proc. 3rd Workshop Algorithm Engineering* (1999).  
(<http://terra.act.uji.es/REA/papers/wae99.ps.gz>).
- [11] Martins, E.Q.V., An algorithm for ranking paths that may contain cycles, *European Journal of Operational Research* 18 (1984) 123-130.
- [12] Martins, E.Q.V., Pascoal, M.M.B., Rasteiro, D.M.L.D. and Santos, J.L.E., The optimal path problem, *Investigação Operacional* 19 (1999) 43-60.  
(<http://www.mat.uc.pt/~eqvm/cientificos/investigacao/Artigos/opath.ps.gz>).
- [13] Martins, E.Q.V., Pascoal, M.M.B. and Santos, J.L.E., A new algorithm for ranking loopless paths, *Research Report, CISUC* (1997).  
(<http://www.mat.uc.pt/~eqvm/cientificos/investigacao/Artigos/mps.ps.gz>).
- [14] Martins, E.Q.V., Pascoal, M.M.B. and Santos, J.L.E., Labeling algorithms for ranking shortest paths, Submitted (1998).  
(<http://www.mat.uc.pt/~eqvm/cientificos/investigacao/Artigos/labeling.ps.gz>).
- [15] Martins, E.Q.V., Pascoal, M.M.B. and Santos, J.L.E., Deviation algorithms for ranking shortest paths, *The International Journal of Foundations of Computer Science* 10 (1999) 247-263.  
(<http://www.mat.uc.pt/~eqvm/cientificos/investigacao/Artigos/deviation.ps.gz>).
- [16] Martins, E.Q.V. and Santos, J.L.E., A new shortest paths ranking algorithm, *Investigação Operacional* 20 (2000) 47-62.  
(<http://www.mat.uc.pt/~eqvm/cientificos/investigacao/Artigos/K.ps.gz>).
- [17] Shier, D., Interactive methods for determining the k shortest paths in a network, *Networks* 6 (1975) 151-159.

# AN ALGORITHM FOR CALCULATING K SHORTEST PATHS WITH A MAXIMUM NUMBER OF ARCS

**Teresa Gomes\***

**Lúci Martins**

**José Craveirinha**

Departamento de Engenharia Electrotécnica/INESC-Coimbra  
Pólo II da Universidade de Coimbra  
Pinhal de Marrocos  
3000 Coimbra  
Portugal  
teresa@dee.uc.pt  
lucia@dee.uc.pt  
jcrav@dee.uc.py

## **Abstract**

In multiexchange telecommunication networks the problem of calculating shortest paths involves, in general, constraints on the maximum number of arcs in any path. In highly connected networks the number of available paths between pairs of nodes is very high and if no additional constraints are introduced, shortest paths (for a given objective function) may occur having a significant number of arcs.

In this paper we present an efficient algorithm for calculating the  $k$  shortest loopless paths with a maximum number of arcs per path, derived from the MPS algorithm [5,6] for calculating the  $k$  shortest loopless paths.

The proposed algorithm will be compared with the original MPS algorithm when both procedures are used for enumerating the  $k$  shortest loopless paths with a maximum number of arcs, in order to show the significant efficiency improvement obtained from the algorithm. The comparison will be made in terms of the total number of generated paths in each algorithm.

## **Keywords**

Routing, shortest paths enumeration

## **1. Introduction**

The calculation of the  $K$ -shortest paths in a network has many applications in science and engineering, namely as a solution to a sub-problem (see ex. [3, 2]). An interesting application in telecommunications as sub-procedure in a multi-objective routing model was recently proposed [1]. In a telecommunication multiexchange network routing context, the determination of optimal/sub-optimal loopless paths (according to predefined criteria) is normally constrained by the maximum number of arcs which may constitute any path. It must be noted that in highly

---

\* Work partially supported by FCT, project PRAXIS/P/EEI/13219/1998, Um estudo sobre encaminhamento dinâmico multi-objectivo e dependente do estado em redes multi-serviço.

connected networks the number of available paths between pairs of nodes is very high and if no additional constraints are introduced, shortest paths (for a given objective function) may occur having a significant number of arcs.

A first (simplistic) approach to the  $K$ -shortest loopless path problem with a maximum ( $D$ ) number of arcs would be to generate the sequence of the shortest paths, using a very efficient algorithm, such as the one proposed by Martins et al. [6] and then to select a subset of  $K$  paths by ignoring those which have more than  $D$  arcs. Note that to the best of our knowledge this algorithm is so far the most efficient one available in the literature. As an alternative an algorithm is proposed in this paper for solving this  $K$  shortest path constrained problem, which is based on the algorithm for loopless paths in [6] which is a variant of the algorithm in [5]. It will be shown that the proposed algorithm is much more efficient than the former (simplistic) approach since it prevents, by construction, the enumeration of paths with more than  $D$  arcs, therefore reducing the number of generated paths which would be disregarded later on.

This paper is organised as follows. In section 2, beyond reviewing the definitions and notation used in [5] some new definitions will be introduced. The foundations of the algorithm will be presented in section 3. The application of the algorithm to various network examples and its comparison with the simple method of rejecting the paths with more than  $D$  arcs in the MPS algorithm, will be performed in section 4. Finally some conclusion on this work and the envisaged application of the algorithm in a bi-objective routing model for telecommunication networks, will be briefly mentioned.

## 2. Definitions and Notation

This section begins by reviewing the basic definitions and notation used in [5]. Let  $(N,A)$  denote the graph defining a network<sup>1</sup> topology, where  $N = \{v_1, \dots, v_n\}$  (or  $N = \{1, \dots, n\}$  to simplify) is the finite set of nodes (or vertices) and  $A = \{a_1, \dots, a_m\}$  is the finite set of arcs or edges, such that each arc is a pair of nodes. If all the pairs are ordered (unordered) the graph is said to be ordered (unordered). Hereafter and unless something explicit is said, the graph is supposed to be ordered. A path  $p$  from  $s$  to  $t$  ( $s, t \in N$ ) is defined by an alternating sequence of nodes and arcs,  $\langle s = v'_1, a'_1, v'_2, \dots, a'_{r-1}, v'_r = t \rangle$ , where  $a'_k \in A$  for any  $k = 1, \dots, r-1$  and  $v'_k \in N$  for any  $k = 1, \dots, r$ ;  $a'_k = (v'_k, v'_{k+1})$  for any  $k = 1, \dots, r-1$ . The value (or cost)  $c(p)$  of a path  $p$  is obtained by associating a cost  $c_{ij} \in \mathbb{R}$  with each arc  $(i,j)$ :

$$c(p) = \sum_{(i,j) \in p} c_{ij} \quad (1)$$

A path from  $s$  to  $t$  in  $(N,A)$  is a loopless path iff all its nodes are different. Let  $p_{ij}$  be a path from  $i$  to  $j$ ; the concatenation of paths  $p_{ij}$  and  $p_{jl}$  is the path,  $p_{ij} \diamond p_{jl}$ , from  $i$  to  $l$ , which coincides with  $p_{ij}$  from  $i$  to  $j$  and with  $p_{jl}$  from  $j$  to  $l$ .

<sup>1</sup> The term network will be reserved to designate a broader entity which includes the graph  $(N,A)$  as a fundamental descriptive structure - e.g. a telecommunication multiexchange network.

Let  $P^K = \{p_1, \dots, p_K\}$  be the set of the  $K$  shortest (unconstrained) loopless paths between a given pair of nodes ( $s$  and  $t$ ). An algorithm enabling to construct  $\mathcal{T}_K$ , a pseudo-tree of the  $K$ -shortest paths (from  $s$  to  $t$ ), algorithm T1, is given in [5]. When a path  $p_k$ ,  $k < K$ , is added to  $\mathcal{T}_{k-1}$ , creating  $\mathcal{T}_k$ , the deviation node of  $p_k$ ,  $v_k$ , is determined as the node whose tail is the first arc where  $\mathcal{T}_{k-1}$  and  $\mathcal{T}_k$  differ, according to algorithm T1. In the context of the MPS algorithm the node  $v_k$  of path  $p$ , from which a new candidate path is generated is the deviation node of that new path. By definition  $s$  is the deviation node of  $p_1$ . The sub-path from  $v_k$  to  $t$  in  $p_k$  is the deviation path of  $p_k$ ,  $p_{v_k t}^k$ . Note that although nodes are repeated in  $\mathcal{T}_K$ , they are all considered as being different in the pseudo-tree. Let  $\mathcal{T}_t$  designate a tree where there is a unique path from any node  $i$  to  $t$  (tree rooted at  $t$ ) and  $c_i(\mathcal{T}_t)$  denote the cost of the path  $p_{it}$  in  $\mathcal{T}_t$ ; the reduced cost  $\tilde{c}_{i,j}$  of arc  $(i,j) \in A$  associated with  $\mathcal{T}_t$  is  $\tilde{c}_{i,j} = c_{ij}(\mathcal{T}_t) - c_i(\mathcal{T}_t) + c_j$ . So all arcs in  $\mathcal{T}_t$  have a null reduced cost. The advantage of using reduced costs was first noted by Eppstein [3] and is shown, by theorems 8 and 9 in [5] and by theorem 2.1 in [6], in the context of the MPS algorithm, to lead to less arithmetic operations and path generation simplification.

Next additional notation and definitions used in the proposed algorithm will be introduced. Let  $|p|$  denote the number of arcs in path  $p$ , from  $s$  to  $t$ , or path length. The notation  $D_p$  will designate a path such that  $D_p = p$  if  $|p| \leq D$ ; otherwise  $D_p$  is the sub-path with origin  $s$  and  $D$  arcs. If  $D = 0$  then  $D_p$  is the empty set  $\emptyset$ . Let  $p$  be a path which contains nodes  $v_i$  and  $v_j$ . The distance from  $v_i$  to  $v_j$  in  $p$  will be given by the number of arcs in  $p$ , from  $v_i$  to  $v_j$ ; if  $v_i, v_j$  are extreme nodes of an arc then the distance is 1; the distance of a node to itself is zero. Let  $p$  be a path from  $s$  to  $t$ , which contains  $v_i$ ; the depth of  $v_i$  in  $p$  is given by the distance from  $s$  to  $v_i$  in  $p$  and will be designated by  $d_p(v_i)$ . Let  $\mathcal{T}_t^*$  be the tree of the shortest paths from all nodes to  $t$ . The height of mode  $v_i$  in  $\mathcal{T}_t^*$  is given by the distance from  $v_i$  to  $t$ ;  $p_{v_i t}^*$  represents the shortest path from  $v_j$  to  $t$  in  $\mathcal{T}_t^*$ . Let the set of arcs  $A$  of  $(N,A)$  be written in terms of  $A(v)$ , the set of arcs the tail node of which is  $v \in N = \{1, \dots, n\}$ , i.e.  $A = A(1) \cup A(2) \cup \dots \cup A(n)$  such that  $A(i) \cap A(j) = \emptyset$  for any  $i \neq j$  ( $i, j \in N$ ). Let  $a'_k \in A(\xi)$  and  $a'_l \in A(\theta)$ . Assuming that  $\xi < \theta$  an order relation ' $<$ ' is defined for the arcs such that  $a'_k < a'_l$  iff  $\xi < \theta$ . Moreover if  $\xi = \theta$  then  $a'_k < a'_l$  if  $\tilde{c}(a'_k) < \tilde{c}(a'_l)$ . This means that the set  $A$  is sorted in such a way that for any two arcs  $(k,j), (i,l) \in A$ ,  $(k,j) < (i,l)$  if  $k < i$  or  $(k = i$  and  $\tilde{c}_{kj} < \tilde{c}_{il})$ . The resulting set  $A = \{a_1, \dots, a_m\}$  is said to be in the sorted forward star form, which leads to the path generation simplification.

### 3. The algorithm

Let one consider that only shortest paths with length less than or equal to  $D$ , be acceptable. Then the  $K$ -shortest path constrained problem which we are seeking to solve for two given nodes  $s, t$  is equivalent to determine a set  $P_D^K = \{p_1, p_2, \dots, p_K\} \subset P \equiv P_{st}$  where  $P_{st}$  is the set of paths which can be defined from  $s$  to  $t$  in  $(N,A)$  such that:

1.  $|p_k| \leq D, k \in \{1, \dots, K\}$
2.  $c(p_k) \leq c(p_{k+1}), k \in \{1, \dots, K\}$

$$3. c(p_k) \leftarrow c(p) \text{ for any } p : |p| \leftarrow D \text{ and } p \in P_{-D}^K$$

The proposed algorithm has the additional (desirable) property that  $p_k$  is identified just before  $p_{k+1}$  for any  $k \in \{1, \dots, K-1\}$ .

Let  $\mathcal{T}_k$  designate the pseudo-tree of the  $k$ -shortest loopless paths from  $s$  to  $t$ , with a maximum of  $D$  arcs and  $X$  the set of the candidate paths to the following shortest path in the context of the algorithm MPS in [5]. The justifications underlying the proposed variant to the algorithm MPS [5] in the version in [6] are now presented.

Let  $p_u$  be the shortest path in  $X$ , and  $v$  its deviation node. From  $p_u$ , paths with deviation node at depth at most  $D-1$ , will be obtained from  $p_u$  only if their deviation node  $v_i$ ,  $v_i \in t$  and  $v_i \in p_{vt}^u$  are such that  $|p_{sv_i}^u| < D-1$  or,  $|p_{sv_i}^u| = D-1$  and  $p_{sv_i}^u \diamond \langle v_i, (v_i, t), t \rangle \in p_u$ , where  $p_{sv_i}^u$  represents the sub-path of  $p_u$  from  $s$  to  $v_i$ . Therefore paths of length greater than  $D$  will be placed in  $X$  if their deviation node is at a depth less than  $D$ . The efficiency of the proposed algorithm stems precisely from not generating paths of length greater than  $D$  which are of no interest, because, when selected, they would be discarded and any path obtained from them would also certainly be discarded too.

The MPS algorithm begins by obtaining the shortest path tree from all nodes to  $t$ . In telecommunication networks, most measures of interest are non negative, therefore the Dijkstra algorithm can be used, for it will be considered that  $c(a) \geq 0$ ,  $a \in A$ ; in fact, any suitable labelling algorithm could be used for this purpose [4]. The inverse Dijkstra algorithm, used as a starting point of the algorithm, will be adapted to associate each node with its height in the shortest paths tree from all nodes to  $t$ . This is easily obtained by giving to  $t$  the height 0 and establishing that whenever the successor of a node  $v_i$  is changed, being for example  $v_j$ , then the height associated with  $v_i$  is also changed and becomes the height of  $v_j$  plus 1.

A final note on the used notation. Given a path  $p$  (from  $s$  to  $t$ ) and a node  $v$  in  $p$ ,  $p_{sv}$  represents the sub-path of  $p$  from  $s$  to  $v$  and  $p_{vt}$  represents the sub-path of  $p$  from  $v$  to  $t$ . Also let  $p_{vt}^*$  represent the shortest path from  $v$  to  $t$  in  $\mathcal{T}_t^*$ .

Next the algorithm for generating the  $K$ -shortest paths of length less or equal to  $D$ , is formalised.

### Algorithm KD

1. Input: the representation of the graph  $(N, A)$  and arc costs  $c_{ij}$ .
2. By using Dijkstra inverse algorithm (including the node heights calculations), obtain the shortest paths from every node to  $t$ , hence constructing the shortest tree rooted at  $t$ ,  $\mathcal{T}_t^*$ .
3. Calculate the reduced cost  $\tilde{c}_{ij}$  for every  $(i, j) \in A$ .
4. Rearrange the arcs of  $(N, A)$  in the sorted forward star form (for the computed  $\tilde{c}_{ij}$ , leading to  $A = \{a_1, \dots, a_m\}$  such that for any  $h \in \{1, \dots, m-1\}$ ,  $c_{ah} \leftarrow \tilde{c}_{ah+1}$  if  $v = \text{tail}(a_h) = \text{tail}(a_{h+1})$ ).
5.  $p \leftarrow$  shortest path from  $s$  to  $t$  ( $p \in \mathcal{T}_t^*$ ).
6.  $k \leftarrow 0$ .

7.  $X \leftarrow \{p\}$  ( $X$  is the set of paths that are candidates to shortest path).
8. While ( $k < K$ ) and ( $X \neq \emptyset$ ) do
  - (a)  $p \leftarrow$  shortest path in  $X$
  - (b)  $X \leftarrow X - \{p\}$
  - (c) If  $|p| \leq D$  and  $p$  has no loops then
    - i.  $k \leftarrow k + 1$
    - ii.  $p_k \leftarrow p$  (the  $k$ -th path which has at most  $D$  arcs, was found)
 EndIf
  - (d) Let  $v$  be the deviation node of  $p$
  - (e) If ( $|p| = D$ ) and ( $|p_{sv}| < D - 1$ )  
 then  $w \leftarrow D - 2$  (try to generate new paths)  
 else If ( $|p| < D$  and  $|p_{sv}| < D$ )  
 then  $w \leftarrow \min(|p| - 1, D - 1)$  (try to generate new paths)  
 else  $w \leftarrow -1$  (no path will be generated from  $p$ )  
 EndIf EndIf
  - (f) If  $w = -1$  then (possibly a path will be placed in  $X$ )
    - i.  $v_i \leftarrow v$
    - ii. Repeat
      - A.  $a_h \leftarrow$  the arc of  $p$  the tail of which is  $v_i$
      - B.  $p_{sv_i} \leftarrow$  sub-path of  $p$  from  $s$  to  $v_i$
      - C. While ( $v_i$  is the tail of  $a_{h+1}$ ) and ( $a_{h+1}$  forms a loop with  $p_{sv_i}$ ) do
        - $h \leftarrow h + 1$
 EndWhile
      - D. If  $v_i$  is the tail  $a_{h+1}$  then
        - $v_j \leftarrow$  head of  $a_{h+1}$
        - $X \leftarrow X \cup \{p_{sv_i} \diamond \langle v_i, a_{h+1}, v_j \rangle \diamond p_{v_j t}^*\}$
 EndIf
      - E. If  $d_p(v_i) = w$   
 then  $w \leftarrow -1$  (no more paths will be obtained from  $p$ )  
 else  $v \leftarrow$  following node in  ${}^w p$  (truncated path  $p$  after  $w$  arcs)  
 EndIf
 Until ( $p_{sv_i}$  has a loop) or ( $w = -1$ )  
 EndIf( $w = -1$ )

#### 4. Analysis of Results

Having in mind to compare the algorithm KD proposed in section 3, (which will be designated hereafter by "KD") with results from the algorithm MPS [6] (hereafter designated by

"MPS") for the purpose of generating the K-shortest loopless paths with a maximum of D arcs, numerous computational experiments were carried out. Strongly meshed networks and fully meshed networks (corresponding to strongly or fully connected graphs) were considered<sup>2</sup>, with 20, 50, 200 and 1000 nodes. The comparison measure was the total number of paths generated by each algorithm until the K-shortest path with at most D arcs was obtained. To facilitate the comparison, values were obtained by averaging the number of generated paths corresponding to solve the problem for pairs of origin-destination nodes in a given network and at least in two networks obtained from different random number sequences, which determine the values of the arc cost (in a given interval) and also the network arcs in the case of strongly meshed networks.

A major factor which influences the relative performance of MPS is the range of variation of arcs costs. For this reason results were presented for two ranges of variation: a moderate variation, from 1 to 10 in 20 node networks up to 1 to 40 in 1000 node networks and a second range of larger relative variations of 1 to 50 in 20 node networks and 1 to 150 for the remaining networks. In MPS the higher is the cost variation, the higher is the average number of arcs of the shortest paths, hence the poorer is the relative performance of MPS, as illustrated in figures 1, 2 and 3.

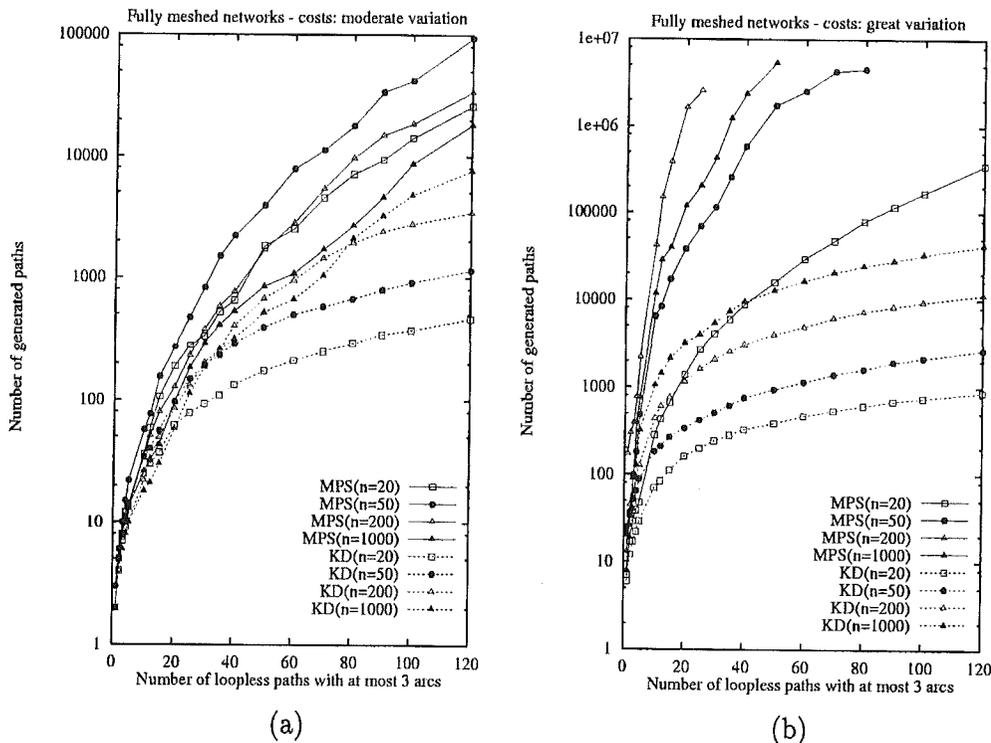


Figure 1 - Comparison of the algorithms for D = 3; (a) n = 20, 50,  $c_{ij} \in [1, 10]$ ; n = 200,  $c_{ij} \in [1, 20]$  and n = 1000,  $c_{ij} \in [1, 40]$  (b) n = 20,  $c_{ij} \in [1, 50]$  and n = 50, 200, 1000,  $c_{ij} \in [1, 150]$

<sup>2</sup> The used program for network generation was kindly borrowed by José Luís Santos.

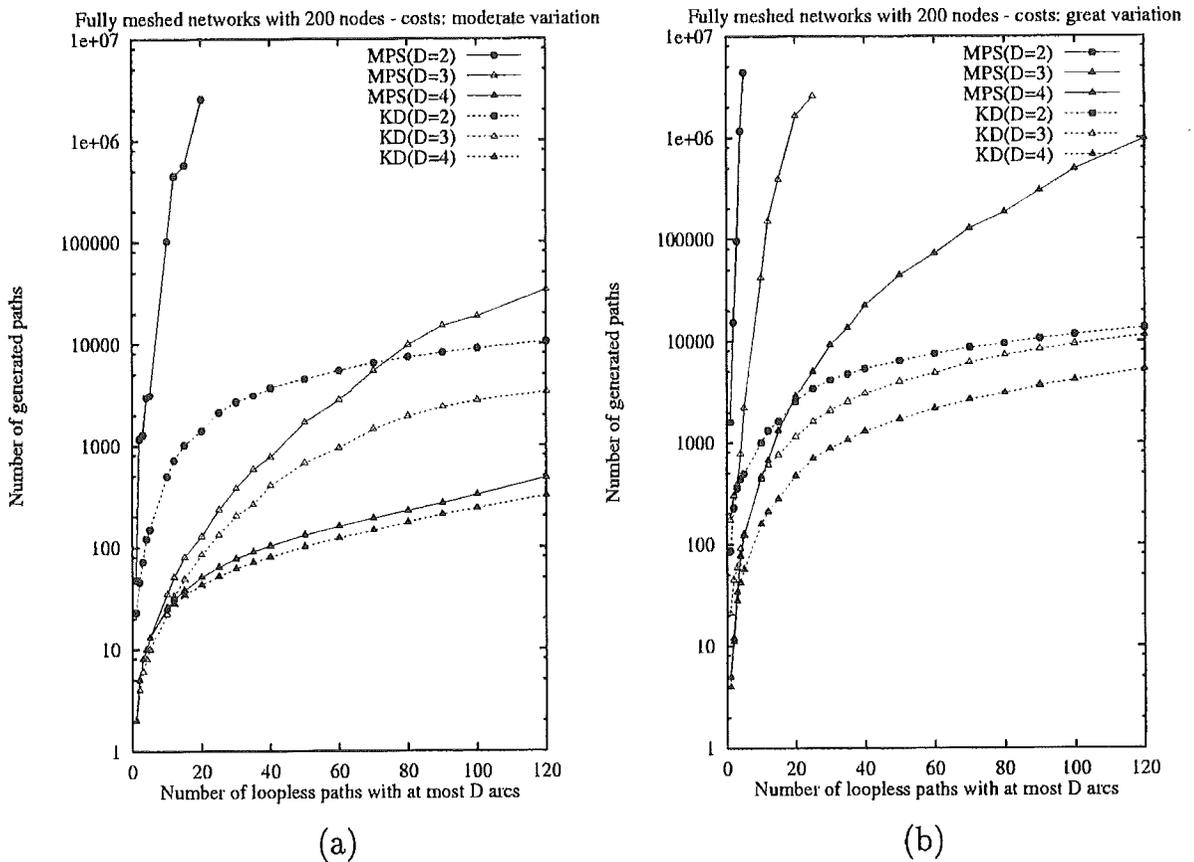


Figure 2 - Comparison of the algorithms for fully meshed networks with 200 nodes and  $D = 2, 3, 4$ ; (a) arc costs  $\in [1, 20]$  (b) arc costs  $\in [1, 150]$

Some of the most significant results are condensed in figures 1, 2 and 3. The presented curves represent averages over the total number of paths generated for obtaining up to 120 shortest paths (whenever they exist) between two pairs of nodes with a maximum of  $D = 2, 3$  or 4 arcs. Note that these are the values of  $D$  (specially  $D = 2, 3$ ) which are of interest to applications in route generation algorithms in multiexchange networks. It should be noted that for  $D = 2$  the use of any of the algorithms is not necessarily advantageous. In fact in many cases it would be more efficient simply to generate all the paths with two arcs and order them by decreasing cost. The considered number of generated paths in the number of paths inserted in the set  $X$  of candidate paths in the course of algorithm execution. To prevent excessive times in some cases (due to swapping) any of the algorithms is stopped when the  $10^7$ -th is generated even if the last shortest path with up to  $D$  arcs hasn't been obtained. This limiting situation occurred some times with the MPS algorithm, but never with the KD algorithm. This has to do with the fact that KD uses much less memory than MPS because the pseudo-tree of the  $K$ -shortest paths generated by KD is much smaller. This is obviously a significant advantage of the KD algorithm.

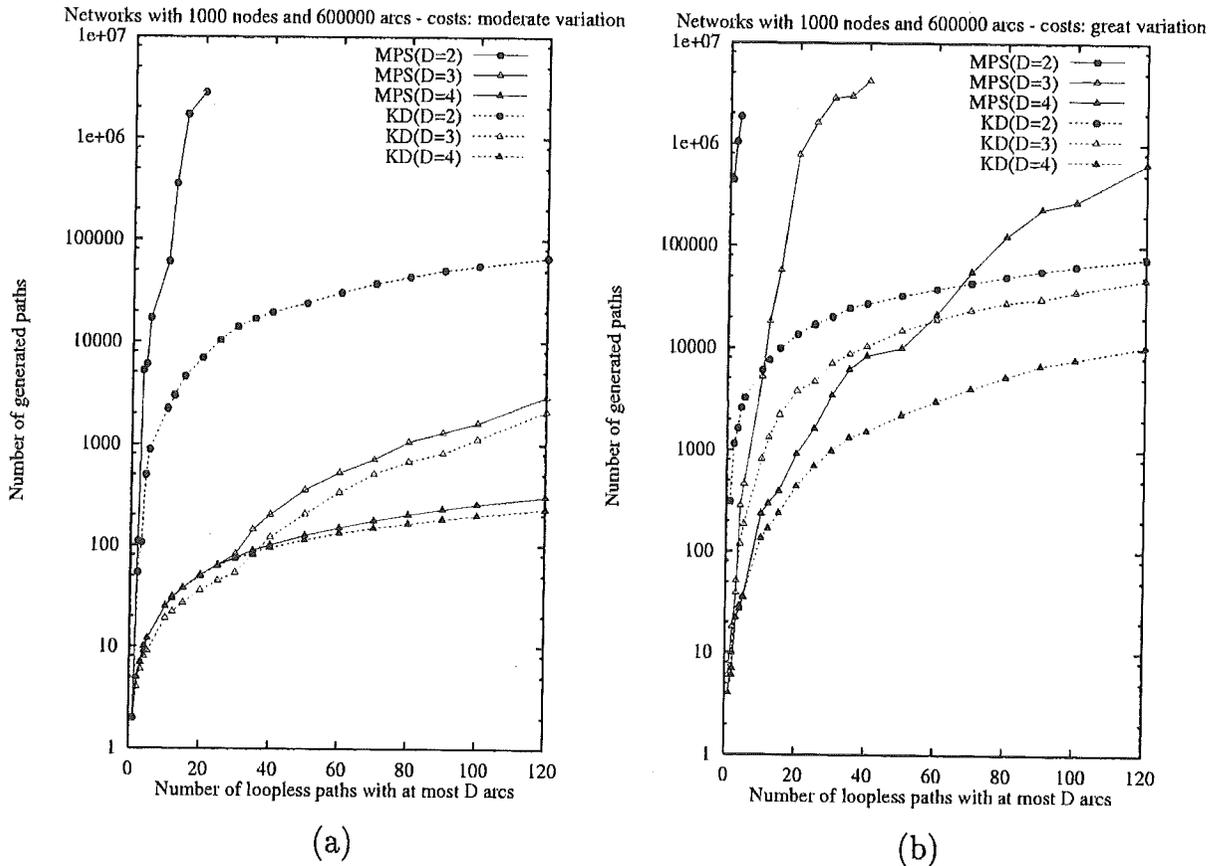


Figure 3 - Comparison of the algorithms for non-fully meshed networks with 1000 nodes and 600000 arcs,  $D = 2, 3, 4$ ; (a) arc costs  $\in [1, 20]$  (b) arc costs  $\in [1, 150]$

It is apparent from the results the quite significant advantage of KD with respect to MPS in most cases. This highly improved efficiency of KD is expectable having in mind that KD does not generate unnecessary paths (that is with more than  $D$  arcs - when such paths are of no potential interest) while it makes the most of the inherent efficiency of MPS. Other advantage of KD stems from the fact that it manages to identify the phase in which there are no more paths with the desired number of arcs. This is particularly important in non fully meshed networks where one seeks to obtain paths with 2 or 3 arcs, a common situation in telecommunication networks.

In general the total number of paths generated by MPS is closer to the equivalent number of KD, when the maximum number of arcs is higher and the relative variation of the arc costs is smaller. This stems from the fact that, in these cases, almost all the first shortest paths generated by MPS have a number of arcs less than or equal to  $D$ .

It was also verified that the relative performance of the algorithms did not depend so much on the network dimensions that on the cost variations. This may be appreciated from figures 1(a) and 1(b) where the results are presented for several fully meshed networks and for two ranges of cost variations, with  $D = 3$ .

## 5. Applications and Conclusions

In some applications of the  $K$  shortest path problem an important constraint is the maximum number of arcs in any chosen path. In particular, for multiexchange telecommunication networks, the determination of optimal or suboptimal (according to some criteria) paths which may be used for routing calls between pairs of nodes (associated with exchanges in terms of network representation) imposes, for technical reasons, that only paths with up to a certain number of arcs, may be considered. This has to do with the necessity of simplifying control and signalling functions and prevent undesirable situations from the point of view of the traffic flow distributions. Note also that multiexchange telecommunication networks are usually either fully meshed or strongly meshed and technology enables the utilisation of sophisticated routing mechanisms. Such mechanisms usually require frequent recalculation (for different network conditions e.g. related to traffic loads or channel occupations) of shortest paths for each pair of origin-destination nodes. An example of this type of application environment for this constrained  $K$ -shortest path problem is a multi-objective routing model proposed in [1]. The basis of this model is to consider two objective functions (to be minimised) in the routing model such as cost and blocking probability or cost and delay. This model involves the selection of non-dominated solutions (paths) of a bi-objective problem taking into account some preference criteria defined on the objective function space. The algorithm used for tackling this problem uses repeatedly the  $K$ -shortest path algorithm [6] and the authors are developing a new variant of the model enabling the choice of alternative paths when for example the first selected path is blocked (ie. at least one arc is unavailable), a technique that is known as alternative routing, also considering as explicit constraint the maximum number  $D$  of arcs per path ( $D = 2, 3, 4$ ).

An algorithm was therefore proposed in this paper for solving the  $K$ -shortest path problem with a constraint on the number of arcs per path, which is based on the MPS algorithm for loopless paths in [6] (which is a variant of the algorithm in [5]). It was shown that the proposed algorithm (algorithm KD) is much more efficient in most cases than the simplistic approach of ignoring the paths selected by MPS with more than  $D$  arcs. This was confirmed by extensive computational results some of which were discussed in the paper. This highly improved efficiency of KD is expectable having in mind that KD does not generate unnecessary (that is with more than  $D$  arcs - when such paths are of no potential interest) while it makes the most of the inherent efficiency of MPS. Other significant advantage of KD is the fact that it uses much less memory than the MPS algorithm, because the pseudo-tree of the shortest paths generated by KD is much smaller. Also KD, unlike MPS, in non-fully meshed networks manages to identify the phase in which there are no more paths with the desired number of arcs. This is particularly important in non fully meshed networks where one seeks to obtain paths with 2 or 3 arcs, a common situation in telecommunication networks.

The experimental results also suggested that in general the total number of paths generated by the two algorithms is closer when the maximum number of arcs is higher and the relative

variation of the arc is smaller. This stems from the fact that, in these cases, almost all the first shortest paths generated by MPS have a number of arcs less than or equal to  $D$ .

### **Acknowledgements**

The authors are very grateful to Professor Ernesto Martins, Marta Pascoal and José Luís Santos for their explanations concerning the MPS algorithm and its implementation.

### **References**

- [1] Antunes, C.Hengeller, Craveirinha, J., Clímaco, J. and Barrico, C., A multiple objective routing algorithm for integrated communication networks, In *Teletraffic Engineering in a Competitive World - ITC 16*, Elsevier (1999) 1291-1300.
- [2] Clímaco, J.C.N. and Martins, E.Q.V., A bicriterion shortest path algorithm, *European Journal of Operational Research* 11 (1982) 399-404.
- [3] Eppstein, D., Finding the  $k$  shortest paths, *SIAM Journal on Computing* 28 (1998) 652-673.
- [4] Martins, E., Pascoal, M., Rasteiro, M. and Santos, J., The optimal path problem, *Investigação Operacional* 19 (1999) 43-60.
- [5] Martins, E., Pascoal, M. and Santos, J., Deviation algorithms for ranking shortest paths, *International Journal of Foundations of Computer Science* 10 (1999) 247-263.  
<http://www.mat.uc.pt/~marta/research.html>.
- [6] Martins, E., Pascoal, M. and Santos, J., An algorithm for ranking loopless paths, *Research Report 99/007, CISUC* (1999). <http://www.mat.uc.pt/~marta/Publicacoes/mps2.ps.gz>.

# APLICAÇÃO DO MÉTODO DE PARTIÇÃO E GERAÇÃO DE COLUNAS AO PROBLEMA DO FLUXO MULTICOMODIDADE

**Filipe Alvelos**

**J.M. Valério de Carvalho**

Departamento de Produção e Sistemas  
Universidade do Minho  
4710-053 Braga  
Portugal  
falvelos@dps.uminho.pt  
vc@dps.uminho.pt

## **Abstract**

We present a branch-and-price algorithm to solve the minimum cost multicommodity flow problem with general integer variables and multiple origins and destinations for each commodity.

A path-based formulation is used. The linear relaxation is solved using column generation, doing the implicit pricing by solving shortest path problems (subproblem) with modified costs.

Obtaining the optimal integer solution implies allowing columns to be generated in the nodes of the branch-and-bound tree. However, the branching constraints may destroy the structure of the subproblem. We develop a branching rule based on arcs flows that maintains the subproblem structure. The modified costs can now be negative, because of the branching constraints of type "greater-than-or-equal-to", and the network of the subproblem may have negative cycles. Based on flow decomposition theorem we propose a procedure to deal with that issue efficiently.

We present computational results for instances with different characteristics, such as number of nodes, number of commodities and density of the network. Also, we compare the results obtained by the branch-and-price algorithm with the results obtained by solving the arc formulation with CPLEX.

## **Resumo**

Neste artigo é apresentado um novo algoritmo, baseado no método de partição e geração de colunas ("branch-and-price"), que permite a obtenção de uma solução ótima do problema do fluxo multicomodidade inteiro de custo mínimo com múltiplas origens e destinos para cada comodidade.

Na resolução da relaxação linear é utilizada uma formulação baseada em fluxos em caminhos e utilizado o método de geração de colunas, correspondendo o subproblema ao problema do caminho mais curto entre cada par origem-destino de cada comodidade.

Para obter uma solução ótima inteira torna-se necessário permitir a geração de colunas nos nodos da árvore de pesquisa do método de partição e avaliação sucessivas ("branch-and-bound"). É utilizada uma regra de partição baseada em fluxos em arcos que preserva a estrutura do subproblema nos nodos da árvore de pesquisa. Com base no Teorema da Decomposição de Fluxos, é utilizado um procedimento que evita o cálculo do caminho mais curto numa rede com circuitos de comprimento negativo.

São comparados os resultados de testes computacionais obtidos com o método desenvolvido e os obtidos resolvendo a formulação baseada em fluxos em arcos por um programa que utiliza exclusivamente a biblioteca de funções CPLEX.

## **Keywords**

Branch-and-price, Multicommodity Flows, Column Generation.

## 1. Introdução

O problema do fluxo multicomodidade de custo mínimo define-se sobre uma rede orientada em que várias comodidades partilham a capacidade dos arcos de forma a serem transportadas dos nodos onde estão disponíveis para os nodos onde são requeridas. Quando se pretende obter uma solução inteira para os fluxos em todos os arcos, é designado por problema do fluxo multicomodidade inteiro (PFMCI) de custo mínimo.

O problema do fluxo multicomodidade de custo mínimo pode ser encarado como uma extensão do problema do fluxo de custo mínimo, já que, não considerando as capacidades dos arcos, se obterá um conjunto de problemas de fluxo de custo mínimo independentes. No entanto, ao contrário do que acontece com esse problema, a solução óptima da relaxação linear do PFMCI de custo mínimo não é necessariamente inteira. Esta observação marca uma clara diferença entre os dois problemas. Existem diversos algoritmos que obtêm uma solução óptima do problema do fluxo inteiro de custo mínimo em tempo polinomial; pelo contrário, o problema do fluxo multicomodidade inteiro de custo mínimo é um problema NP-difícil [14].

Para além da importância teórica dos modelos de fluxo multicomodidade (poder-se-á referir que um dos resultados mais relevantes na optimização de problemas com estrutura angular em blocos e restrições adicionais - o método da decomposição de Dantzig-Wolfe [9] - foi inspirado no trabalho de Ford e Fulkerson sobre o problema do fluxo máximo multicomodidade [11]), o interesse que têm despertado, desde os finais da década de 50, deve-se à multiplicidade de áreas em que encontram aplicação. A título exemplificativo referem-se aplicações de modelos de fluxo multicomodidade em problemas de distribuição/transporte, redes de computadores e planeamento de produção [1,7,19,29].

São conhecidas múltiplas abordagens à resolução da relaxação linear do PFMCI e de variantes deste problema. A generalidade destas abordagens podem ser classificadas em métodos de decomposição orientada pelos preços [1,3,5,18,22], métodos de decomposição orientada pelos recursos [1] e métodos de partição de base [1,3,4,8,10,21].

Neste artigo é apresentado um novo método de resolução para a obtenção de uma solução óptima inteira do PFMCI, baseado no método de partição e geração de colunas ("branch-and-price"). Este método tem-se revelado uma ferramenta poderosa na resolução exacta de problemas de programação inteira e de optimização combinatória. Embora o método da geração de colunas seja conhecido há cerca de quatro décadas (tem as suas raízes no método de decomposição de Dantzig-Wolfe), a sua combinação com o método de partição e avaliação sucessivas ("branch-and-bound") para obtenção de soluções óptimas inteiras apenas na última década recebeu o interesse devido ao seu potencial [17].

A utilização do método de partição e geração de colunas torna-se incontornável na obtenção de soluções inteiras de problemas cuja formulação tem um número enorme de variáveis. Para uma discussão sobre as razões para se procurarem formulações com essa característica, ver [6].

No caso particular do PFMCI em análise, algumas das vantagens da aplicação do método de partição e geração de colunas consistem na facilidade de incorporação de novas restrições no modelo (como a existência de limites ao número de arcos que um caminho pode conter, relevante em problemas em redes de computadores em que a fiabilidade e os tempos de atraso em nodos sejam factores críticos no desempenho do sistema [15], ou a existência de janelas temporais associadas aos vértices, relevante em problemas de distribuição/transporte) e no menor número de restrições que se obtém ao utilizar uma formulação com um número enorme de variáveis.

A grande maioria dos problemas que têm sido investigados traduzem-se em modelos com variáveis binárias. Em [5] é utilizado um método de partição e geração de colunas na resolução do problema do fluxo multicomodidade binário. Nesse problema, é assumido que todo o fluxo de uma comodidade deve ser encaminhado por um único caminho. Esta simplificação permite a utilização de variáveis binárias, cada uma delas associada a um caminho diferente entre a origem e o destino da comodidade.

No problema a que se refere este artigo, existem múltiplos pares origem-destino para cada comodidade. O fluxo que emana de um nodo pode ser repartido por vários caminhos diferentes e conduzido para nodos diferentes. Esta generalização implica a utilização de uma metodologia de resolução diferente, utilizando variáveis inteiras gerais.

A literatura sobre a aplicação do método de partição e geração de colunas a problemas com variáveis inteiras gerais restringe-se, segundo o nosso melhor conhecimento, a experiências sobre o problema de corte [23,24,25,26,27,28].

Na nossa perspectiva, as principais contribuições deste trabalho são as seguintes:

- a utilização do método de partição e geração de colunas para a obtenção de soluções exactas para o problema do fluxo multicomodidade com variáveis inteiras gerais;
- a definição de uma regra de partição, que, conjugada com uma formulação baseada em caminhos e circuitos, permite manter a estrutura do subproblema em todos os nodos da árvore de pesquisa, nomeadamente naqueles cujo problema mestre restrito inclui restrições de partição do tipo " $\geq$ ";
- a utilização da introdução de colunas correspondentes a circuitos no problema mestre restrito para evitar o recurso à resolução de um problema de caminho elementar mais curto em grafos com circuitos de comprimento negativo, que é um problema NP-completo; é dada uma interpretação para o uso destas colunas, como correspondendo a cortes no espaço dual;
- a implementação de um código baseado na biblioteca de funções CPLEX [16] para a resolução da formulação baseada em fluxos em arcos e a realização de testes computacionais comparativos da eficiência do método desenvolvido.

Na Secção 2 deste artigo, depois de definido o problema do fluxo multicomodidade inteiro e estabelecida a notação utilizada, são apresentadas duas formulações para o PFMCI, uma baseada em fluxos em arcos e outra baseada em fluxos em caminhos. São referidas as relações

existentes entre as duas formulações. O teorema da decomposição de fluxos é abordado nesse contexto, servindo posteriormente de suporte ao método desenvolvido.

Na Secção 3 é apresentado o método de geração de colunas, e é demonstrada a validade da sua aplicação à resolução da relaxação linear do PFMCI.

Na Secção 4 é introduzido o método de partição e geração de colunas. São discutidas diferentes regras de partição e as suas consequências na estrutura do subproblema. É estabelecida uma regra de partição que preserva a estrutura do subproblema com base no Teorema da Decomposição de Fluxos. São ainda abordados outros aspectos relativos ao método desenvolvido.

Na Secção 5 é descrita a implementação do método, e discutido, com base em resultados de testes computacionais, o desempenho de diferentes alternativas. Os resultados dos testes computacionais, para instâncias geradas aleatoriamente e com diferentes características, são comparados com os resultados obtidos com um programa que usa exclusivamente uma biblioteca de funções comercial.

Na Secção 6 são apresentadas as principais conclusões extraídas da realização deste trabalho e possíveis melhorias ao método desenvolvido.

## 2. Formulações para o PFMCI

### 2.1 Definições e Notação Utilizada

Uma instância do PFMCI caracteriza-se pelos seguintes parâmetros:

$N$  - conjunto de nodos,  $N = \{1, 2, \dots, n\}$ ;

$A$  - conjunto de arcos,  $A = \{1, 2, \dots, m\}$ ;

$K$  - conjunto de comodidades,  $K = \{1, 2, \dots, h\}$ ;

A cada arco  $ij \in A$ , está associado um nodo origem  $i$  e um nodo destino  $j$ . A cada nodo  $i$  e a cada comodidade  $k$  está associado o parâmetro  $b_i^k$ . Se  $b_i^k > 0$ , o nodo  $i$  é uma origem para a comodidade  $k$ , tendo uma oferta de  $b_i^k$  unidades. Se  $b_i^k < 0$ , o nodo  $i$  é um destino para a comodidade  $k$ , tendo uma procura de  $-b_i^k$  unidades. Se  $b_i^k = 0$ , o nodo não é origem nem destino para a comodidade  $k$ , designando-se por nodo de transbordo puro.

A cada arco  $ij$  e a cada comodidade  $k$  está associado o parâmetro  $c_{ij}^k$  que corresponde ao custo de um fluxo unitário da comodidade  $k$  atravessar o arco  $ij$ . Considera-se  $c_{ij}^k \geq 0$ ,  $\forall ij \in A$ ,  $\forall k \in K$ , embora valores negativos trouxessem alterações menores ao procedimento desenvolvido [2].

A cada arco  $ij$  está associada a capacidade  $u_{ij}$ , que limita o fluxo total que atravessa o arco. As ofertas e as procuras das diferentes comodidades, assim como a capacidade dos arcos, são expressas na mesma unidade de medida. Cada unidade de fluxo de uma comodidade  $k$  que atravessa um arco  $ij$  consome uma unidade de capacidade.

O conjunto de todos os caminhos simples entre todos os pares origem-destino da comodidade  $k$  é representado por  $P^k$ , e o conjunto de todos os nodos origem ou destino da comodidade  $k$  por  $Q^k$ . Se o arco  $ij$  faz parte do caminho  $p$  da comodidade  $k$ ,  $\delta_{ij}^{pk}$  toma o valor 1,

e 0, caso contrário. Se o nodo  $i$  é o nodo origem do caminho  $p$  da comodidade  $k$ ,  $\delta_i^{pk}$  toma o valor 1; se  $i$  é o nodo destino, o valor -1; se não, o valor 0. O custo unitário do caminho  $p$  da comodidade  $k$ , representa-se por  $c_k^p = \sum_{ij \in A} \delta_{ij}^{pk} c_{ij}^k, \forall p \in P^k, \forall k \in K$ .

O conjunto de todos os circuitos simples da comodidade  $k$  é representado por  $C^k$ . Se o arco  $ij$  faz parte do circuito  $c$  da comodidade  $k$ ,  $\delta_{ij}^{ck}$  toma o valor 1, e 0, caso contrário. O custo unitário do circuito  $c$  da comodidade  $k$ , representa-se por  $c_c^k = \sum_{ij \in A} \delta_{ij}^{ck} c_{ij}^k, \forall c \in C^k, \forall k \in K$ .

**2. 2 Formulação Nodo-Arco**

Uma formulação para o PFMCI pode ser obtida com base na definição das variáveis de decisão como sendo o fluxo em cada arco de cada comodidade. Essas variáveis, designadas por variáveis originais, são representadas por  $x_{ij}^k$ . A formulação assim obtida é designada por formulação nodo-arco e apresentada de seguida.

$$\text{Min} \quad \sum_{k \in K} \sum_{ij \in A} c_{ij}^k x_{ij}^k$$

sujeito a:

$$\sum_{ij \in A} x_{ij}^k - \sum_{ij \in A} x_{ji}^k = b_i^k, \forall i \in N, \forall k \in K \tag{1}$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij}, \forall ij \in A \tag{2}$$

$$x_{ij}^k \geq 0, \forall ij \in A, \forall k \in K \text{ e inteiras.}$$

As restrições (1) implicam que haja conservação de fluxo nos nodos para todas as comodidades. São as restrições que definem que se está perante uma rede onde não há perdas nem ganhos: para cada comodidade, o fluxo que entra num nodo difere do fluxo que sai do nodo pela oferta/procura do mesmo.

As restrições (2) não permitem que o fluxo de cada arco, para todas as comodidades, ultrapasse um determinado valor - a capacidade do arco. Estas restrições são designadas por restrições de capacidade mútua ("bundle constraints"). São estas restrições que impedem que o problema possa ser decomposto em problemas de fluxo de custo mínimo independentes (um problema para cada comodidade).

**2. 3 Formulação Arco-Caminho**

Uma outra formulação válida para o PFMCI, designada por formulação arco-caminho, é a seguinte:

$$\text{Min} \quad \sum_{k \in K} \sum_{p \in P^k} c_p^k y_p^k$$

sujeito a:

$$\sum_{p \in P^k} x_i^k y_p^k = b_i^k, \forall i \in Q^k, \forall k \in K \quad (3)$$

$$\sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^{pk} y_p^k \leq u_{ij}, \forall ij \in A \quad (4)$$

$$y_p^k \geq 0, \forall k \in K, \forall p \in P^k \text{ e inteiras.}$$

Esta formulação obtém-se, definindo cada variável de decisão,  $y_p^k$ , como sendo o fluxo que atravessa o caminho  $p$  da comodidade  $k$ . As restrições (3) implicam a conservação de fluxo nos nodos para todas as comodidades e as restrições (4) traduzem a existência de capacidade mútua nos arcos.

Ao contrário da formulação nodo-arco, a formulação arco-caminho tem um número de variáveis que cresce exponencialmente com a dimensão da rede. Por outro lado, o número de restrições em instâncias com nodos de transbordo puro será menor na formulação arco-caminho.

As duas formulações são equivalentes: a formulação arco-caminho pode ser obtida aplicando uma decomposição de Dantzig-Wolfe à formulação nodo-arco depois de adicionados nodos super-origem e super-destino e transferindo as restrições de conservação de fluxo para o subproblema [2].

A equivalência entre as duas formulações decorre também da aplicação do teorema da decomposição de fluxos [1], estendido para o caso multicomodidade, e da Proposição 1, que garante que os fluxos nos circuitos são nulos na solução ótima. Estes dois resultados são apresentados de seguida:

### Teorema da Decomposição de Fluxos

i) Um conjunto de fluxos em caminhos e circuitos é equivalente ao conjunto de fluxos em arcos que se obtém através de:

$$x_{ij}^k = \sum_{p \in P^k} \delta_{ij}^{pk} y_p^k + \sum_{c \in C^k} \delta_{ij}^{ck} y_c^k, \forall ij \in A, \forall k \in K,$$

onde  $y_p^k$  toma o valor do fluxo no caminho  $p$  da comodidade  $k$  e  $y_c^k$  toma o valor do fluxo no circuito  $c$  da comodidade  $k$ .

ii) A cada conjunto de fluxos em arcos corresponde um conjunto de fluxos (não necessariamente único) de fluxos em caminhos e circuitos.

iii) Todos os caminhos com fluxo positivo têm como origem um nodo com oferta e como destino um nodo com procura.

Se assumirmos que os custos unitários são todos não negativos, não haverá circuitos de custo negativo, e solução ótima obedece à seguinte caracterização:

**Proposição 1:** Numa instância do PFMCI em que todos os circuitos têm custo não negativo e o conjunto de soluções válidas é não vazio, existe uma solução ótima para a qual

$$y_c^k = 0, \forall k \in K, \forall c \in C^k.$$

O mesmo se aplica para a solução óptima da relaxação linear do PFMCI.

**Demonstração:** As variáveis de circuito ( $y_c^k$ ) não influenciam as restrições de conservação de fluxo (3). Como as restrições (4) são do tipo menor ou igual, qualquer variável  $y_c^k$  positiva pode ser anulada de tal forma que as restrições continuam a ser respeitadas e o valor da função objectivo diminui monotonamente, já que todas as variáveis de circuito têm como coeficiente na função objectivo um valor não negativo. ♦

A combinação destes dois resultados serve de base ao desenvolvimento do método, em particular à definição da regra de partição, conforme será apresentado na Secção 4.

### 3. Resolução da Relaxação Linear do PFMCI pelo Método de Geração de Colunas

#### 3.1 O Método de Geração de Colunas

O método de geração de colunas baseia-se na consideração implícita de um número que se espera elevado de variáveis. Em problemas com um número enorme de variáveis, como o PFMCI na formulação arco-caminho, será a única abordagem de resolução possível.

Neste método, em cada iteração, é resolvido um problema mestre restrito (PMR) e um (vários) subproblema(s). O PMR é inicializado com um número restrito de variáveis do problema original. Depois de otimizado, a informação relativa ao valor das variáveis duais é transferida para o subproblema, o que permite avaliar a atractividade das variáveis que não fazem parte do PMR. Caso haja variáveis atractivas, as colunas a elas correspondentes são inseridas no PMR e este é reotimizado, repetindo-se de seguida o mesmo processo. Caso contrário, a solução óptima do PMR é a solução óptima do problema original.

#### 3.2 Resolução da Relaxação Linear do PFMCI pelo Método de Geração de Colunas

Na aplicação do método de geração de colunas à resolução da relaxação linear do PFMCI, o problema mestre restrito é inicializado com um número reduzido de caminhos. Depois de obtida a solução óptima desse problema é avaliada a atractividade de caminhos que dele não fazem parte. Esta avaliação é feita com base na transferência do valor das variáveis duais para o subproblema. O subproblema consiste em determinar o caminho mais curto entre todos os pares origem-destino de todas as comodidades numa rede com custos modificados, o que agora se justifica.

Representando por  $w_{ij}$  a variável dual associada à restrição de capacidade mútua do arco  $ij$  e por  $\pi_i^k$  a variável dual associada à restrição de conservação de fluxo do nodo  $i$  para a comodidade  $k$ , o custo reduzido de um caminho é dado por

$$\bar{c}_p^k = \sum_{ij \in A} \delta_{ij}^{pk} (w_{ij} + c_{ij}^k) - \pi_o^k + \pi_d^k, \quad (5)$$

onde o índice  $o$  e o índice  $d$  representam os nodos origem e destino, respectivamente, do caminho  $p$ .

Os caminhos que fazem parte do PMR têm custo reduzido não negativo, o que decorre das condições de optimalidade. A solução óptima do PMR não será óptima para a relaxação linear do PFMCI, se existir algum caminho não considerado no PMR com custo reduzido negativo.

Na equação (5) as parcelas  $-\pi_0^k + \pi_d^k$  são constantes para uma mesma comodidade  $k$  e para caminhos que tenham a mesma origem e o mesmo destino. Cada uma das restantes parcelas está associada a um arco, sendo constante para todos os caminhos de uma mesma comodidade que o inclua.

Considerando uma comodidade  $k$ , o caminho com menor custo reduzido entre uma origem  $o$  e um destino  $d$  é o caminho mais curto entre  $o$  e  $d$  numa rede em que os custos dos arcos são dados por  $w_{ij} + c_{ij}^k, \forall ij \in A$ . Note-se que  $w_{ij} \geq 0, \forall ij \in A$ .

Este caminho é atractivo se

$$\sum_{ij \in A} \delta_{ij}^{pk} (w_{ij} + c_{ij}^k) < \pi_0^k + \pi_d^k.$$

Caso contrário, existe a garantia de que não existem caminhos atractivos para a comodidade  $k$  para o par origem-destino  $o-d$ .

Determinar o caminho mais atractivo de cada comodidade corresponde assim a resolver um problema de caminho mais curto entre todos os pares origem-destino.

#### 4. Resolução do PFMCI pelo Método de Partição e Geração de Colunas

##### 4.1 O Método de Partição e Geração de Colunas

Como referido em [6], numa primeira abordagem poderá parecer que o método de partição e geração de colunas ("branch-and-price") não envolve nada mais do que a junção do método de geração de colunas e do método de partição e avaliação sucessivas ("branch-and-bound"): em cada nodo da árvore de pesquisa do método de partição e avaliação sucessivas resolve-se até à optimalidade o problema de programação linear utilizando o método de geração de colunas. Note-se que se não fosse permitida a geração de colunas nos nodos da árvore de pesquisa, poder-se-ia obter uma solução não óptima ou concluir da impossibilidade do problema inteiro sem tal corresponder à verdade.

No entanto, o problema a resolver em cada nodo da árvore inclui restrições de partição que não estavam presentes na relaxação linear do problema. Estas restrições podem alterar ou destruir a estrutura do subproblema, tornando ineficiente ou impossível a avaliação da atractividade das variáveis que não fazem parte do PMR a resolver em cada nodo de pesquisa.

Assim, o desafio que se coloca ao aplicar o método de partição e geração de colunas a um dado problema é o de definir regras de partição compatíveis com a estrutura do subproblema.

##### 4.2 Regras de Partição e Preservação da Estrutura do Subproblema

A definição de regras de partição com base nas variáveis da formulação arco-caminho, fluxos em caminhos, pode conduzir a que seja possível que o subproblema indique como variável atractiva uma variável que já faz parte do PMR, designando-se a coluna a ela associada

como uma coluna regenerada. A regeneração pode ocorrer por as variáveis duais associadas às restrições de partição não poderem ser incluídas nos custos modificados dos arcos, já que estão associadas a caminhos.

No pior caso resolver o problema da regeneração num nodo da árvore de pesquisa de profundidade  $k$  corresponde a resolver o problema dos  $k$  caminhos mais curtos. Assim, a regra de partição baseada em fluxos em caminhos traduz-se numa alteração da estrutura do subproblema.

Com base no Teorema da Decomposição de Fluxos, é possível definir uma outra regra de partição que, baseando-se nas variáveis originais, não altera a estrutura do subproblema, continuando este a ser o problema do caminho mais curto entre todos os pares origem-destino para cada comodidade.

De acordo com o Teorema da Decomposição de Fluxos e com a Proposição 1, o fluxo de uma comodidade  $k$ , num arco  $ij$ , é dado por

$$x_{ij}^k = \sum_{p \in P^k} \delta_{ij}^{pk} y_p^k \tag{6}$$

Com base em (6) é possível definir uma regra de partição que utiliza o valor das variáveis originais, da seguinte forma:

$$\sum_{p \in P^k} \delta_{ij}^{pk} y_p^k \leq \lfloor x_{ij}^k \rfloor \tag{7}$$

e

$$\sum_{p \in P^k} \delta_{ij}^{pk} y_p^k \geq \lceil x_{ij}^k \rceil, \tag{8}$$

onde  $x_{ij}^k$  tem um valor fraccionário.

Utilizando a regra de partição agora definida e representando  $U^s$  como o conjunto de restrições de partição do tipo (7), de índice  $u$ , e  $L^s$  como o conjunto de restrições de partição do tipo (8), de índice  $l$ , presentes num nodo da árvore de pesquisa  $s$ , o PMR a ele associado é:

$$\text{Min} \quad \sum_{k \in K} \sum_{p \in P^k} c_p^k y_p^k$$

sujeito a:

$$\sum_{p \in P^k} \gamma_i^{pk} y_p^k = b_i^k, \forall i \in Q^k, \forall k \in K$$

$$\sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^{pk} y_p^k \leq u_{ij}, \forall ij \in A$$

$$\sum_{p \in P^k} \delta_{ij}^{pk} y_p^k \leq \lfloor x_{ij}^{ku} \rfloor, \forall u \in U^s \tag{9}$$

$$\sum_{p \in P^k} \delta_{ij}^{pk} y_p^k \geq \lceil x_{ij}^{kl} \rceil, \forall l \in L^s \quad (10)$$

$$y_p^k \geq 0, \forall k \in K, \forall p \in P^k,$$

onde  $R^k$  é o conjunto dos caminhos da comodidade  $k$  que fazem parte do problema mestre restrito,  $R^k \subseteq P^k, \forall k \in K$ .

O custo reduzido de um caminho  $p$ , com origem  $o$  e destino  $d$ , de uma comodidade  $k$  é agora

$$\bar{c}_p^k = \sum_{ij \in A} \delta_{ij}^{pk} (w_{ij} + \sum_{u \in U^s} w_{ij}^{ku} - \sum_{l \in L^s} w_{ij}^{kl} + c_{ij}^k) - \pi_o^k + \pi_d^k,$$

onde  $w_{ij}^{ku}$  e  $w_{ij}^{lu}$  são as variáveis duais associadas às restrições de partição (9) e (10), respectivamente. De notar que estas variáveis estão associadas a arcos  $o$  que permite a inclusão do seu valor nos custos modificados da rede do subproblemas.

O subproblema de cada comodidade continua assim a ser o problema do caminho mais curto entre todos os pares de vértices numa rede com custos modificados. O custo modificado de um arco  $ij$ , de uma comodidade  $k$ , é dado por

$$w_{ij} + \sum_{u \in U^s} w_{ij}^{ku} - \sum_{l \in L^s} w_{ij}^{kl} + c_{ij}^k.$$

A contribuição negativa das parcelas relativas às restrições de partição do tipo " $\geq$ " implica a possibilidade de existência de circuitos de comprimento negativo na rede de um subproblema. Note-se que as variáveis duais associadas às restrições de capacidade mútua e de partição são não negativas.

O problema do caminho mais curto numa rede com circuitos de comprimento negativo é um problema NP-completo [14]. Para tornar este problema é apresentado em [2] um algoritmo baseado na resolução de sucessivos problemas de caminho mais curto em redes que diferem da original por remoção de arcos que façam parte de circuitos com comprimento negativo. Os resultados de testes computacionais aí apresentados demonstram a ineficiência deste método quando comparado com o que agora se apresenta.

#### 4.3 Resolução dos Problemas dos Nodos da árvore de Pesquisa

A abordagem adoptada até ao momento considera apenas variáveis relativas a caminhos, baseando-se a sua validade na Proposição 1. Uma implicação dessa abordagem é que, na resolução, usando geração de colunas, do problema de um nodo da árvore de pesquisa, podem existir circuitos de comprimento negativo nas redes dos subproblemas.

Na Figura 1, onde junto aos arcos se indicam restrições de partição do tipo " $\geq$ " para uma mesma comodidade, tal é exemplificado com a representação de parte de um problema de um nodo de uma árvore de pesquisa. Ao ser resolvido o subproblema da comodidade em causa, a rede poderá ter circuitos de comprimento negativo, já que a soma dos custos modificados dos arcos 12, 23 e 31 poderá ser negativa.

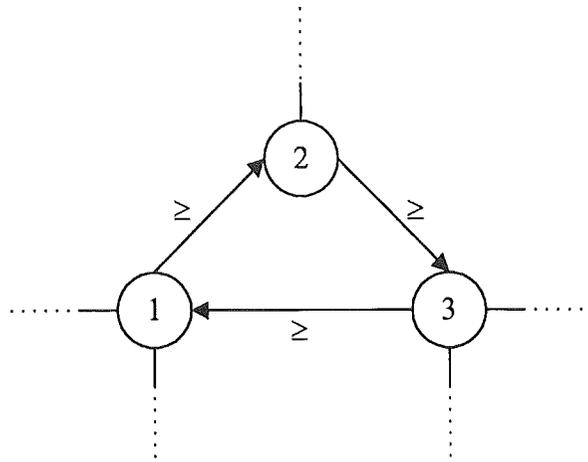


Figura 1 - Exemplo de um circuito que poderá ter comprimento negativo na resolução de um subproblema.

Uma mesma solução pode ser representada de diferentes formas de acordo com a alínea ii do teorema da decomposição de fluxos. No caso do Exemplo, o fluxo no circuito 1231 pode ser representado por fluxos em, por exemplo, dois caminhos, um que inclua os arcos 12 e 23 e outro que inclua o arco 31; ou considerando a variável associada ao circuito 1231.

A abordagem utilizada consiste em considerar as variáveis associadas a circuitos. O problema a resolver num nodo da árvore de pesquisa é:

$$\text{Min} \quad \sum_{k \in K} \left( \sum_{p \in P^k} c_p^k y_p^k + \sum_{c \in C^k} c_c^k y_c^k \right)$$

sujeito a:

$$\sum_{k \in K} \left( \sum_{p \in P^k} \delta_{ij}^{pk} y_p^k + \sum_{c \in C^k} \delta_{ij}^{ck} y_c^k \right) \leq u_{ij}, \forall ij \in A$$

$$\sum_{p \in P^k} \gamma_i^{pk} y_p^k = b_i^k, \forall k \in K, \forall i \in Q^k$$

$$\sum_{p \in P^k} \delta_{ij}^{pk} y_p^k + \sum_{c \in C^k} \delta_{ij}^{ck} y_c^k \leq \lfloor x_{ij}^{ku} \rfloor, \forall u \in U^s \tag{11}$$

$$\sum_{p \in P^k} \delta_{ij}^{pk} y_p^k + \sum_{c \in C^k} \delta_{ij}^{ck} y_c^k \geq \lceil x_{ij}^{kl} \rceil, \forall l \in L^s \tag{12}$$

$$y_p^k \geq 0, \forall k \in K, \forall p \in P^k,$$

$$y_c^k \geq 0, \forall k \in K, \forall c \in C^k.$$

De salientar a inclusão das variáveis associadas a circuitos nas restrições de partição (11) e (12) de acordo com o Teorema da Decomposição de Fluxos e com a regra de partição definida com base nas variáveis originais.

O custo reduzido de uma variável associada a um circuito  $c$  de uma comodidade  $k$  é:

$$\bar{c}_c^k = \sum_{ij \in A} \delta_{ij}^{ck} (w_{ij} + \sum_{u \in U^s} w_{ij}^{ku} - \sum_{l \in L^s} w_{ij}^{kl} + c_{ij}^k).$$

Note-se que as variáveis associadas a circuitos não fazem parte de restrições de conservação de fluxo. O custo reduzido de uma variável associada a um circuito é igual à soma dos custos modificados dos arcos que o integram. A existência de um circuito de comprimento negativo no subproblema de uma comodidade corresponde assim à existência de uma variável associada a um circuito atractiva.

Utilizando um algoritmo de correcção de rótulos [13] é possível resolver os subproblemas de um nodo da árvore de pesquisa de forma eficiente. Estes algoritmos determinam o caminho mais curto ou detectam a existência de um circuito de comprimento negativo, identificando os arcos que o compõem. Assim, cada subproblema de um nodo da árvore de pesquisa pode retornar ao PMR uma coluna correspondente a um circuito, caso tenha sido detectado um circuito de comprimento negativo, ou uma coluna correspondente a um caminho, caso não tenha sido detectado nenhum circuito de comprimento negativo.

Durante a resolução da relaxação linear e numa solução óptima do PFMCI as variáveis associadas a circuitos têm necessariamente o valor nulo, de acordo com a Proposição 1. Numa solução do problema de um nodo da árvore de pesquisa tal pode não ser verdadeiro devido às restrições de partição. Duas situações podem ocorrer:

- a solução óptima de um PMR, dos que ocorrem no processo de geração de colunas, do problema relativo a um nodo da árvore de pesquisa pode ter a variável associada a um circuito com valor positivo, mas na solução óptima desse mesmo nodo, essa variável é não-básica ou básica com o valor nulo;
- a própria solução óptima do problema relativo ao nodo da árvore de pesquisa tem uma variável associada a um circuito que é básica, com valor positivo.

No primeiro caso, o circuito assume temporariamente um valor positivo, mas nunca mais é um circuito atractivo no subproblema. Tal decorre, claramente, do facto de as variáveis básicas terem custos reduzidos nulos. Existe uma interpretação interessante para a introdução de circuitos no PMR, que decorre de ver os processos de geração de colunas como algoritmos de planos de corte duais [20]: inserir uma variável no PMR corresponde a eliminar uma porção do espaço dual, evitando, desse modo, que as soluções pertencentes à porção cortada sejam regeneradas.

No segundo caso, como corolário da Proposição 1, o nodo pode ser abandonado:

**Corolário:** Um nodo da árvore de pesquisa cuja solução óptima contenha circuitos explicitamente (circuitos com fluxo positivo) ou implicitamente (arcos de diferentes caminhos com fluxo positivo a formarem circuitos) pode ser abandonado.

**Demonstração:** O abandono é feito com base num critério de dominância, por existir um nodo que domina a solução encontrada. Considere-se a solução óptima do nodo da árvore de

pesquisa. Claramente, existe sempre uma outra solução em que os fluxos em todos os caminhos são idênticos, mas os fluxos nos circuitos são nulos. Como a regra de partição explora todo o espaço válido de soluções, haverá um outro nodo em que essa solução será pesquisada. Dado que essa nova solução tem um custo inferior, por todos os circuitos serem de custo não-negativo, o nodo pode ser abandonado. ♦

#### 4. 4 Convergência do Método e Simetria

O número de caminhos e circuitos de uma rede é finito, embora exponencialmente grande com respeito ao número de nodos da rede. O mesmo pode ser dito em relação ao número de restrições de partição, pelo que existe a garantia de o método obter a solução ótima num número finito de passos, que pode ser exponencial, no pior caso.

Uma questão fundamental no desenvolvimento de algoritmos de partição e geração de colunas é a simetria do espaço de soluções. No processo de partição e geração de colunas, a mesma solução, ou o mesmo conjunto de soluções, não deve ser explorado em mais de um nodo da árvore de pesquisa. Claramente, a regra de partição usada, divide o conjunto de soluções válidas em conjuntos mutuamente exclusivos, que são explorados em nodos diferentes da árvore de pesquisa.

### 5. Implementação e Resultados de Testes Computacionais

#### 5. 1 Instâncias de Teste

Para a geração de instâncias do PFMCI foi utilizada o gerador MNETGEN, disponível no endereço <http://www.di.unipi.it/~frangio/> (acedido em Janeiro de 1999).

Os resultados aqui apresentados referem-se a dois grupos de instâncias. Para ambos os grupos (designados por 2 e 3) todos os arcos têm capacidade mútua. Em [2] são ainda apresentados testes computacionais para instâncias em que apenas uma percentagem dos arcos têm capacidade mútua (grupo de instâncias 1), o que torna essas instâncias mais fáceis. Para cada comodidade, metade dos nodos das instâncias do grupo 2 são origens e a outra metade são destinos. Para as instâncias do grupo 3, há apenas uma origem para cada comodidade e metade dos nodos são destinos, sendo os restantes nodos de transbordo puro.

Para cada grupo foram gerados seis subgrupos cuja designação, número de nodos e número de comodidades são apresentadas na Tabela 1.

Designação	a	b	c	d	e	f
n	64	64	64	128	128	128
h	4	8	16	4	8	16

Tabela 1 - Número de nodos e de comodidades das instâncias de teste

Dentro de cada subgrupo foram geradas doze instâncias. Dessas doze, as seis primeiras (de 01 a 06) correspondem a redes esparsas e as seis últimas (07 a 12) a redes densas. A medida de

densidade utilizada é  $m/n$ . Redes esparsas correspondem a  $m/n$  entre 2,5 e 3 (aproximadamente) e redes densas a  $m/n$  entre 5,5 e 6 (aproximadamente).

Embora seja considerado que as instâncias geradas aleatoriamente têm um menor grau de dificuldade que as correspondentes a problemas reais, em [12] é referido que o MNETGEN gera instâncias difíceis, por comparação com outros geradores para o PFMC. Um indicador da dificuldade de uma instância é o número de arcos saturados na solução óptima. Para algumas das instâncias geradas o número de arcos saturados atinge 25% do número de arcos da rede.

Uma caracterização detalhada do processo de geração de instâncias, bem como dos programas desenvolvidos e dos testes computacionais realizados, pode ser encontrada em [2].

## 5. 2 Implementação do Método Desenvolvido

O método apresentado neste artigo foi desenvolvido na linguagem de programação C (foi usado o ambiente de desenvolvimento "Microsoft Visual C++ Version 5.0") e serve-se da biblioteca de funções CPLEX [16] para resolver os problemas de programação linear. Esse programa é referido por PGC.

Nas Figuras 2 e 3 é apresentado o fluxograma do programa PGC. As suas principais funções são agora sucintamente descritas, conjuntamente com as diferentes alternativas que se colocaram no desenvolvimento do método e que foram implementadas. Na Subsecção seguinte são referidos os resultados de testes computacionais que permitiram seleccionar as melhores alternativas.

Em cada momento apenas há um problema mestre restrito no ambiente CPLEX. Na descrição das funções esse problema é denominado por problema actual.

A função LE\_INST lê a instância a resolver, a partir do ficheiro onde está caracterizada para a estrutura de dados que vai ser utilizada para a representar.

A função PPR1 cria o primeiro problema mestre restrito no ambiente CPLEX com base em um de dois algoritmos. Ambos os algoritmos se baseiam em soluções dos problemas de fluxo de custo mínimo independentes que se obtêm relaxando as restrições de capacidade mútua. O primeiro algoritmo determina a solução óptima de cada um desses problemas na formulação nodo-arco e transforma os fluxos em arcos em fluxos em caminhos, de acordo com o Teorema da Decomposição de fluxos e com um procedimento descrito em [1]. Um segundo algoritmo consiste numa heurística miópica ("greedy"), descrita em [2], baseada na determinação de caminhos mais curtos e, tendo em conta as ofertas e procuras dos nodos, na afectação de fluxos a esses caminhos. Não existindo a garantia de a solução obtida por um ou outro procedimento corresponder a um PMR válido, é incluído no PMR uma variável artificial.

A função OPTIMIZA otimiza o problema actual e retorna uma solução dual óptima.

A função PRICE avalia a atractividade das variáveis que não fazem parte do problema mestre restrito e, caso haja variáveis atractivas e dependendo da política de inserção de colunas adoptada, insere colunas atractivas. Caso não haja colunas atractivas, retorna essa informação. As políticas de inserção de colunas implementadas foram as seguintes: em cada iteração inserir

no PMR a coluna mais atractiva, a coluna mais atractiva de cada comodidade, a coluna mais atractiva de cada origem de cada comodidade e a coluna mais atractiva de cada par origem-destino de cada comodidade. A resolução do subproblema de cada comodidade é efectuada com base num algoritmo de correcção de rótulos que gere a lista de vértices por analisar através da regra "primeiro a entrar primeiro a sair". Esse algoritmo determina uma árvore de caminhos mais curtos, sendo invocado para cada origem de cada comodidade. A detecção de circuitos de comprimento negativo é efectuada com base no limite inferior para um rótulo numa rede sem circuitos de comprimento negativo que é a soma das distâncias de todos os arcos com distância negativa. Se, em qualquer iteração, o rótulo de um nodo tiver um valor inferior a esse limite, existe um circuito de comprimento negativo na rede.

Caso a solução óptima da relaxação linear do PFMCI não seja inteira, pode ou não ser efectuada a remoção de todas as colunas com custo reduzido positivo no final da resolução da relaxação linear.

A função *BRANCH* é responsável pela geração e inserção de uma restrição de partição no problema actual. Caso não exista nenhuma variável original com valor fraccionário, a função retorna a informação de que não foi gerada nenhuma restrição de partição. A selecção da variável original sobre a qual é efectuada a partição é feita percorrendo arcos de caminhos e calculando o seu fluxo até ser encontrada, se existir, uma variável original com valor fraccionário.

A função *ANALISA* retorna se o nodo deve ser abandonado ou se devem ser gerados descendentes. Um nodo é abandonado se a sua solução óptima é inteira (podendo ou não ser actualizada a solução incumbente), se a solução é fraccionária e tem um valor não inferior ao da solução incumbente, se tem variáveis associadas a circuitos positivas ou se o problema é impossível.

A função *BACKTRACK* é invocada depois de um nodo de pesquisa ser abandonado. Selecciona o próximo nodo de pesquisa (de entre os que podem conduzir a uma melhoria da solução incumbente), ou retorna a informação de que a árvore de pesquisa não tem mais nodos por pesquisar. A selecção do próximo nodo a pesquisar implica também a alteração do problema actual de tal forma que passe a corresponder ao problema do nodo a pesquisar.

Foi seguida uma estratégia de pesquisa da árvore "primeiro-em-profundidade", com as alternativas de analisar primeiro o nodo resultante da restrição de partição do tipo " $\geq$ " ou o resultante da restrição de partição do tipo " $\leq$ ".

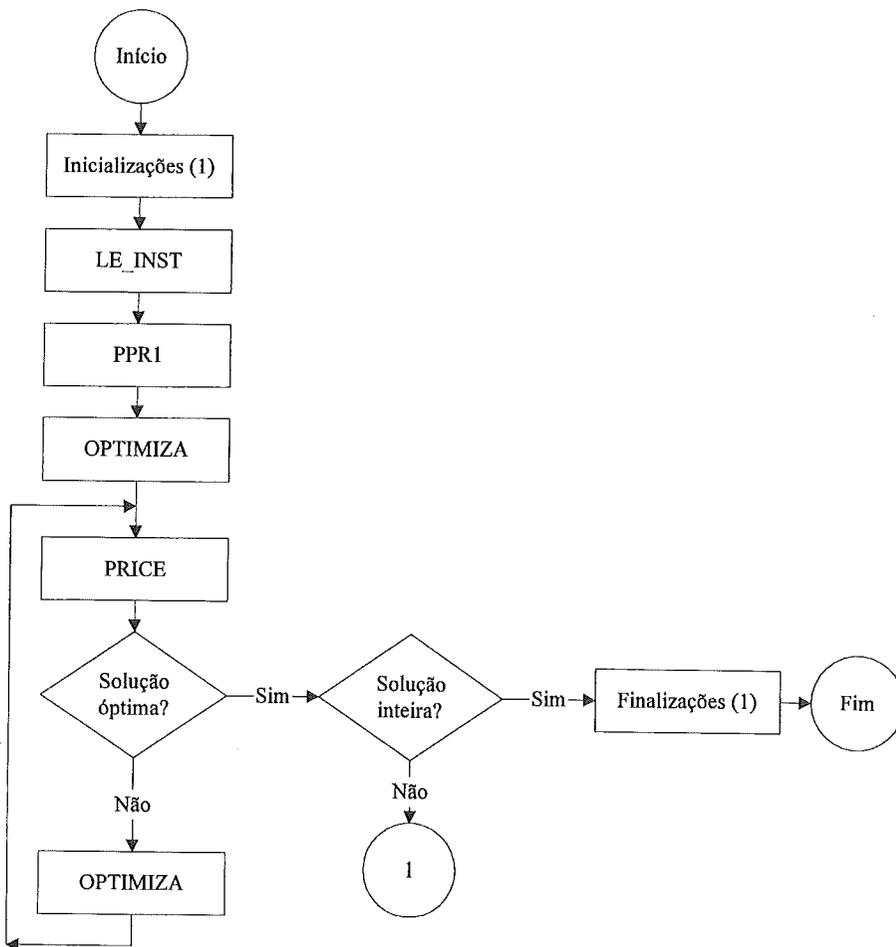


Figura 2 - Fluxograma do método de partição e geração de colunas implementado: resolução da relaxação linear

### 5. 3 Testes Computacionais Preliminares

Para a obtenção do primeiro PMR, os testes computacionais realizados para as instâncias do grupo 2 (para as instâncias do grupo 3, devido a terem apenas uma origem por comodidade, ambos os métodos fornecem a solução óptima dos problemas de fluxo de custo mínimo de cada comodidade [2]) indicaram que o algoritmo que determina a solução óptima dos problemas de fluxo de custo mínimo de cada comodidade permite uma maior eficiência do método, sendo entre 2 a 5 vezes (aproximadamente) mais rápida a obtenção da solução óptima da relaxação linear do PFMCI do que utilizando o algoritmo que se baseia na resolução heurística dos problemas de fluxo de custo mínimo de cada comodidade. A qualidade dos caminhos inicialmente considerados tem uma importância relevante na eficiência do método.

Quanto à política de inserção de colunas, verificou-se que quanto maior o número de colunas inseridas em cada iteração, melhor o desempenho do método. Assim, optou-se por inserir a coluna mais atractiva de cada origem-destino de cada comodidade.

Os testes computacionais realizados permitiram concluir que efectuar a remoção de todas as colunas com custo reduzido positivo no final da resolução da relaxação linear melhora o

desempenho do método. Nos nodos da árvore de pesquisa é gerada um número reduzido de colunas.

Os testes computacionais realizados para avaliar qual o primeiro nodo a analisar (o resultante da restrição de partição do tipo " $\geq$ " ou o resultante da restrição de partição do tipo " $\leq$ " foram inconclusivos: o número de vezes que uma alternativa tem um número de nodos pesquisados menor do que a outra é aproximadamente igual, tendo-se optado por analisar primeiro o nodo resultante da inserção de restrição do tipo " $\geq$ ".

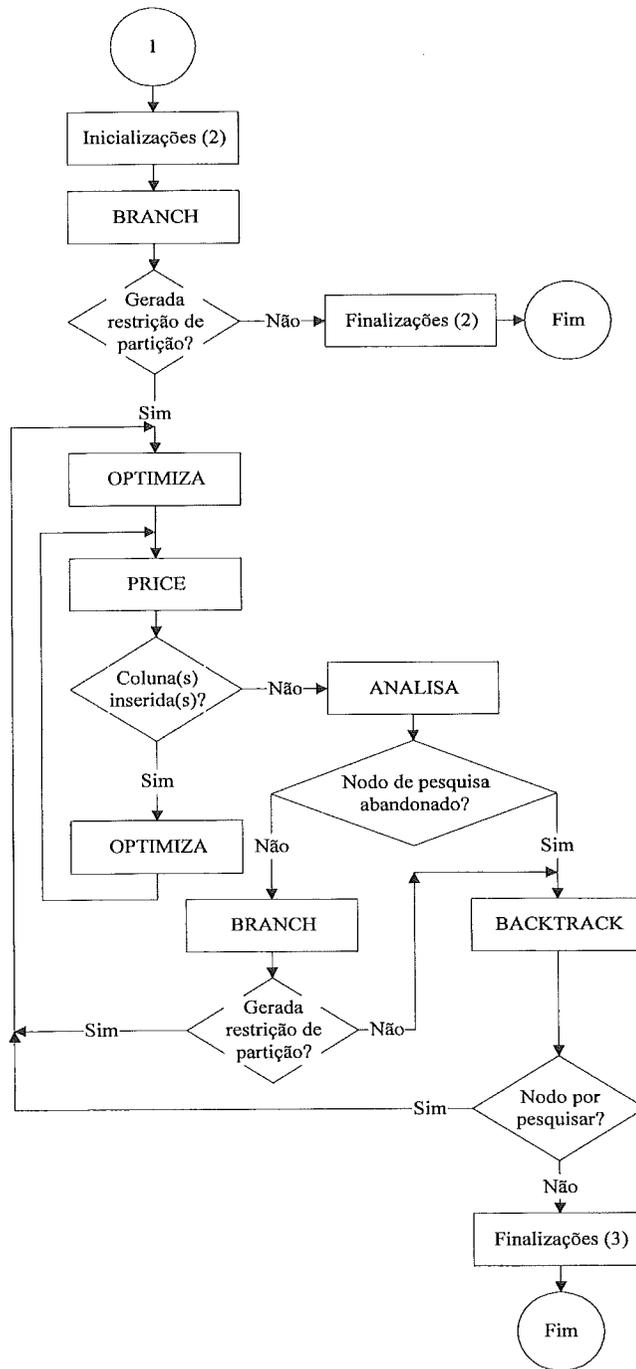


Figura 3 - Fluxograma do método de partição e geração de colunas implementado: resolução do problema inteiro

#### 5. 4 Resultados de Testes Computacionais

Para além do PGC, foi desenvolvido um programa que resolve o problema, na formulação nodo-arco, utilizando a biblioteca de funções CPLEX quer para a optimização da relaxação linear, quer para a obtenção da solução óptima inteira. Esse programa é referido por CPLEX.

Todos os testes computacionais foram realizados num computador pessoal equipado com um processador Pentium operando a uma frequência de relógio de 60 MHz e com 32 Mb de memória RAM.

Nas Figura 4 e 5 são apresentados graficamente os tempos de execução total médios para os dois programas em função do subgrupo de instâncias de teste.

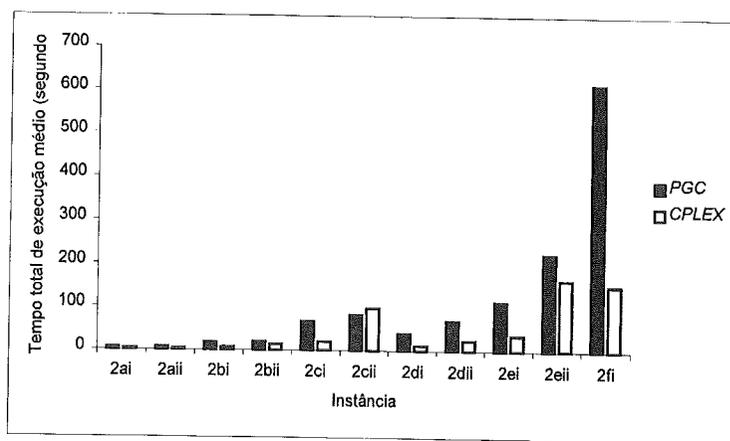


Figura 4 - Comparação entre o desempenho dos programas PGC e CPLEX para as instâncias do grupo 2 (i - subgrupo de 01 a 06, redes esparsas; ii - subgrupo de 07 a 12, redes densas)

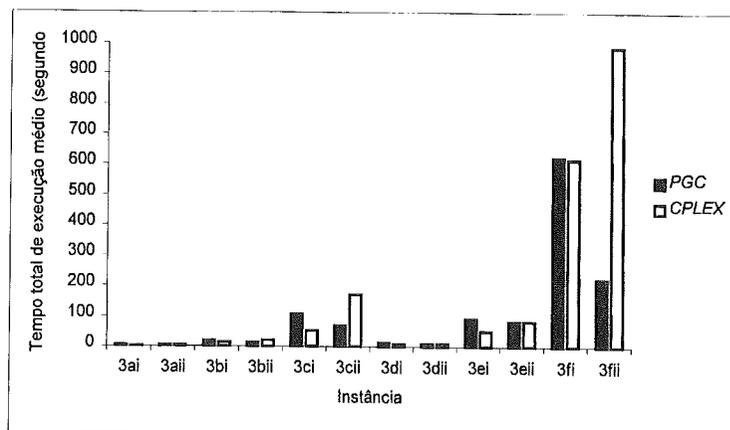


Figura 5 ( Comparação entre o desempenho dos programas PGC e CPLEX para as instâncias do grupo 3 (i - subgrupo de 01 a 06, redes esparsas; ii - subgrupo de 07 a 12, redes densas)

Para a instância 2f09, o PGC atingiu o número de nodos limite da árvore de pesquisa imposto (151 nodos). O subgrupo que inclui essa instância não é representado na Figura 4. Para esse subgrupo de instâncias o desempenho do PGC foi francamente inferior ao do CPLEX (no pior caso demorou 10 vezes mais tempo).

Para as instâncias do grupo 2 (metade dos nodos são origens e a outra metade destinos), o desempenho do PGC foi claramente inferior ao do CPLEX, embora nas instâncias mais

pequenas os tempos tenham sido da mesma ordem de grandeza. Apenas para as instâncias com o menor número de nodos, que correspondem a redes densas e que têm o maior número de comodidades testado, o PGC teve um melhor desempenho que o CPLEX.

Para instâncias do grupo 3 (uma origem e metade dos nodos são destinos), verificou-se que o PGC tem um melhor desempenho para redes densas que o CPLEX. Nas maiores instâncias, correspondentes a redes densas, a diferença entre desempenhos é muito significativa. Para redes esparsas, o desempenho do CPLEX é melhor, embora as diferenças não sejam tão pronunciadas.

Na Tabela 2, apresentam-se resultados relativos a instâncias cuja solução óptima da relaxação linear do PFMCI não é inteira.

Instância	Tempo na árvore de pesquisa (segs.)		Número de nodos pesquisados		Número de colunas do maior problema		Tempo de execução total (segs.)	
	PGC	CPLEX	PGC	CPLEX	PGC	CPLEX	PGC	CPLEX
2b04	7.3	0.6	8	4	864	1456	20.4	6.3
2b06	8.2	0.3	6	1	1612	1424	25.9	5.0
2d01	23.8	0.7	12	4	1502	1448	55.1	9.6
2d09	10.1	0.5	2	2	824	2884	60.9	24.2
2e05	59.0	2.9	15	29	2269	3000	188.3	47.2
2e11	32.7	0.8	1	1	2614	5992	269.7	153.1
2f05	125.7	0.9	18	2	4226	6096	453.5	168.4
2f06	1202.3	7.6	132	46	4546	6064	1624.5	170.0
2f07	9267.7	66.4	148	117	8692	11904	11173.0	1158.1
2f08	758.3	7.3	17	19	4420	12160	2094.0	1003.9
2f09	6829.3*	136.0	151*	172	7434*	12336	8449.0*	1682.3
2f10	311.1	67.0	9	183	3646	12128	1600.8	1051.9
2f11	4978.1	19.2	146	38	6737	12192	6795.4	1136.2
2f12	1997.7	210.6	54	671	4142	11904	3362.4	1183.7
3c01	3.6	0.6	2	1	949	3504	68.8	51.8
3c07	2.8	0.9	2	2	742	5280	57.7	153.4
3c08	4.5	0.8	3	2	1334	5296	67.2	166.7
3c10	24.6	2.6	46	21	850	5280	79.1	154.9
3e04	7.0	1.3	4	8	1633	3360	77.2	46.1
3e06	38.2	1.2	13	9	1367	3488	108.5	61.7
3f03	70.4	9.8	7	47	2446	7104	962.0	546.6
3f04	268.9	6.3	123	32	2189	7040	767.6	472.7
3f05	15.9	2.7	1	4	2586	7008	445.9	418.0
3f08	12.6	2.3	3	5	1612	10512	249.4	755.8
3f09	29.2	10.8	13	30	1345	10560	181.7	775.7
3f12	44.5	12.5	14	47	1868	10304	274.4	1372.5

\* Limite do número de nodos de pesquisa atingido: solução incumbente pode não ser óptima.

Tabela 2 - Resultados da obtenção de uma solução óptima inteira: PGC e CPLEX.

Embora os problemas de programação linear a resolver nos nodos da árvore de pesquisa tenham um menor número de colunas no caso do PGC, os tempos na árvore de pesquisa são sempre, e por vezes muito superiores aos tempos do CPLEX.

Mesmo para as instâncias em que o CPLEX pesquisa um número elevado de nodos, essa pesquisa é muito mais eficiente do que a efectuada pelo PGC. Para o CPLEX, o tempo na árvore de pesquisa é de uma ordem de grandeza muito inferior à do tempo total, o que não é verdade para o caso do PGC.

O bom desempenho comparativo do CPLEX encontra justificação nas técnicas computacionais utilizadas por um programa comercial que incorpora os últimos avanços na área da programação linear e inteira. Por outro lado, é de referir que o desempenho do programa PGC poderá ser melhorado, conforme analisado detalhadamente em [2]. Enunciam-se seguidamente alguns desses aspectos que poderiam trazer melhorias em termos de tempo computacional.

Em algumas instâncias a proporção de tempo em subproblemas em nodos da árvore de pesquisa em relação ao tempo total na árvore de pesquisa (sem a raiz) atinge valores da ordem dos 80%, o que indicia que a utilização de um algoritmo mais eficiente para a resolução dos subproblemas (problemas de caminho mais curto) poderia melhorar em muito o desempenho do PGC.

A alteração da estratégia de pesquisa, o refinamento da regra de partição, a implementação de um procedimento para o abandono prematuro do processo de geração de colunas e a utilização de heurísticas, quer para a melhoria da qualidade da solução do primeiro PMR, quer para a melhoria da qualidade de soluções incumbentes, poderiam ainda beneficiar o desempenho do método.

## **6. Conclusões e Trabalho Futuro**

Neste artigo, apresentou-se um algoritmo para a resolução do problema de fluxo multicomodidade inteiro. Foi mostrado que é possível derivar um algoritmo de partição e geração de colunas, com uma regra de partição que não induz simetria do espaço de soluções e que preserva a estrutura do subproblema.

O desenvolvimento deste trabalho constituiu uma primeira abordagem a este problema. Na versão abordada, o subproblema é um problema de caminho mais curto, logo tem a propriedade da integralidade. É sabido que, neste caso, a solução da relaxação linear do modelo desenvolvido fornece um limite inferior que é igual ao da solução da relaxação linear do modelo com as variáveis originais.

Assim sendo, não era de esperar que o número de nodos da árvore de pesquisa do algoritmo desenvolvido fosse claramente inferior ao obtido com o pacote comercial, que serviu para comparação de resultados. Em casos em que o subproblema não tem a propriedade da integralidade, e a reformulação é mais forte que a formulação com as variáveis originais,

espera-se que haja uma clara vantagem na utilização de modelos que usam o método de partição e geração de colunas.

O algoritmo desenvolvido usa uma regra de partição simples. As restrições que são inseridas nos nodos das árvores de pesquisa podem não dividir o espaço de soluções em sub-árvores que sejam balanceadas. Espera-se que a utilização de regras de partição mais balanceadas, baseadas em hiperplanos, possa conduzir a algoritmos mais robustos para este problema.

Para algumas das instâncias testadas, o método desenvolvido obteve melhores tempos computacionais do que o programa que utiliza exclusivamente a biblioteca de funções CPLEX. É, ainda, de esperar um aumento significativo do seu desempenho (i) introduzindo melhorias nos procedimentos de obtenção do primeiro problema mestre restrito, (ii) utilizando um algoritmo mais eficiente na resolução dos problemas de caminho mais curto, (iii) usando diferentes estratégias de pesquisa, (iv) implementando heurísticas para a obtenção de soluções nos nodos da árvore de pesquisa e (v) utilizando limites inferiores para o abandono prematuro do processo de geração de colunas.

## Referências

- [1] Ahuja, R., Magnanti, T. L. and Orlin, J. B., *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, New Jersey (1993).
- [2] Alvelos, F., *Aplicação do Método de Partição e Geração de Colunas ao Problema do Fluxo Multicomodidade*, Dissertação de Mestrado, Faculdade de Engenharia da Universidade do Porto (1999).
- [3] Bazaraa, M. S. and Jarvis, J. J., *Linear Programming and Network Flows*, John Wiley and Sons (1977).
- [4] Barnhart, C., Hane, C. A., Johnson, E. L. and Sigismondi, G., *A Column generation and Partitioning Approach for Multi-Commodity Flow Problems*, *Telecommunications Systems* 3 (1995) 329-258.
- [5] Barnhart, C., Hane, C. A. and Vance, P. H., *Integer Multicommodity Flow Problems*, In *Integer Programming and Combinatorial Optimization*, Proceedings of the 5th International IPCO Conference, Vancouver, British Columbia, Canada, (1996) 59-71, *Lecture Notes in Computer Science*, Springer.
- [6] Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. and P. H. Vance, *Branch-and-Price: Column Generation for Solving Huge Integer Programs*, *Operations Research* 46 (1998) 316-329.
- [7] Bertsekas, D., *Network Optimization: Continuous and Discrete Models*, Athena Scientific, Belmont, Massachusetts (1998).
- [8] Castro, J. and Nabona, N., *An Implementation of Linear and Nonlinear Multicommodity Network Flows*, *European Journal of Operational Research* 92 (1996) 37-53.
- [9] Dantzig, G. B. and Wolfe, P., *Decomposition Principle for Linear Programs*, *Operations Research* 8 (1960) 101-111.
- [10] Farvolden, J. M., Powell, W. B. and Lustig, I. J., *A Primal Partitioning Solution for the Arc-Chain Formulation of a Multicommodity Network Flow Problem*, *Operations Research* 41 (1993) 669-693.
- [11] Ford, L. R. and Fulkerson, D. R., *A Suggested Computation for Maximal Multicommodity Network Flows*, *Management Science* 5 (1958) 97-101.
- [12] Frangioni, A., *Dual Ascent Methods and Multicommodity Flow Problems*, Ph.D. Thesis, Università Degli Studi di Pisa (1997).
- [13] Gallo, G. and Pallottino, S., *Shortest Paths Algorithms*, *Annals of Operations Research* 13 (1988) 3-79.
- [14] Garey, M. R. and Johnson, D. S., *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company (1979).
- [15] Gouveia, L., *Multicommodity Flow Models for Spanning Tree with Hop Constraints*, *European Journal of Operational Research* 95 (1996) 178-190.
- [16] ILOG, Inc., *Version 5.0 of Using the CPLEX Callable Library* (1997).
- [17] Johnson, E. L., Nemhauser, G. L. and Savelsbergh, M. W. P., *Progress in Integer Programming: An Exposition*, <http://akula.isye.gatech.edu/~mwps> (acedido em Julho de 1999) (1997).
- [18] Jones, K. L., Lustig, I. J., Farvolden, J. M. and W. B. Powell, *Multicommodity Network Flows: The Impact of Formulation on Decomposition*, *Mathematical Programming* 62 (1993) 95-117.
- [19] Kennington, J. L., *A Survey of Linear Cost Multicommodity Network Flows*, *Operations Research* 26 (1978) 209-236.

- [20] Lasdon, L., Optimization Theory for Large Systems, Macmillan Series in Operations Research (1970).
- [21] McBride, R., Progress made in solving the Multicommodity Flow Problem, SIAM Journal on Optimization 8 (1998) 947-955.
- [22] Tomlin, J.A., A Minimum-cost Multicommodity Network Flows, Operations Research 14 (1966) 44-51.
- [23] Valério de Carvalho, J. M., Exact Solution of One-dimensional Cutting Stock Problems using Column Generation and Branch-and-bound, International Transactions in Operational Research 5, 1 (1998) 35-44.
- [24] Valério de Carvalho, J. M., Exact Solution of Bin-Packing Problems Using Column Generation and Branch and Bound, Annals of Operations Research J. M. 86 (1999) 629-659.
- [25] Vance, P. H., Branch-and-price Algorithms for the One-dimensional Cutting-stock Problem, Department of Industrial Engineering, Auburn University, Auburn, Alabama (in preparation) (1998).
- [26] Vanderbeck, F., On Integer Programming Decomposition and Ways to Enforce Integrality in the Master, University of Cambridge, Research Papers in Management Studies 29 (1996).
- [27] Vanderbeck, F., Computational Study of a Column Generation Algorithm for Binpacking and Cutting-stock problems, Mathematical Programming A-86 (1999) 565-594.
- [28] Vanderbeck, F. and Wolsey, L., An Exact Algorithm for IP Column Generation, Operations Research Letters 19 (1996) 151-159.
- [29] Zahorik, A., Thomas, L. J. and Trigeiro, W. W., Network Programming Models for Production Scheduling in Multi-Stage Multi-Item Capacitated Systems, Management Science 30 (1984) 308-325.

# RECONHECIMENTO DE VOGAIS UTILIZANDO UM MÉTODO DE DECISÃO DIFUSO

**Tiago M. L. M. Simas**

**Rita A. Ribeiro**

Departamento de Informática  
FCT - Universidade Nova Lisboa  
2825-114 Monte Caparica  
tms@fct.unl.pt  
rr@di.fct.unl.pt

## **Abstract**

In this work we use a multiple attribute fuzzy decision making method to recognize the five Portuguese vowels. The objective of this preliminary study is to show the potential of the fuzzy method in voice recognition.

## **Resumo**

Neste trabalho utiliza-se um método de decisão multi-atributo difuso (multiple attribute fuzzy decision making) para reconhecer as vogais da língua portuguesa. Trata-se de um estudo preliminar que serve para demonstrar as potencialidades de aplicação do método difuso no reconhecimento de voz.

## **Keywords**

Multiple Attribute Fuzzy Decision Making, Formants, Evolving Peaks.

## **1. Introdução**

Neste artigo apresenta-se uma aplicação de reconhecimento de vogais (voz) da língua Portuguesa utilizando um método de tomada de decisão multiatributo difusa [7]. Trata-se de um estudo preliminar, baseado no artigo de Sankar, e Dwijesh [8] mas tanto o método proposto para recolha e tratamento dos dados como o método de agregação são diferentes. Os resultados obtidos pela utilização do método multiatributo difuso escolhido [5] [6] [7] são discutidos para mostrar as potencialidades desta aproximação.

O trabalho desenvolveu-se em três fases distintas. Primeiro, a recolha de várias vozes pronunciando as vogais {a,e,i,o,u}. Segundo, o seu tratamento em termos de definição dos atributos (critérios) a ser utilizados no reconhecimento. Terceiro, a utilização do método de decisão multiatributo difuso para determinar qual o nível de reconhecimento das vogais. Este nível é dado em percentagem.

Para a recolha dos dados foram usadas tecnologias comerciais, vulgares na recolha dos dados, tais como um microfone unidireccional, impedância 600Ω e frequência de resposta 20Hz-20kHz.

Para o tratamento da informação, tanto a nível da elaboração dos atributos, como da sua agregação foram usados o: MATLAB v5.3 e SOUND FORGE 4.5.

A seguir apresentam-se breves descrições dos fundamentos teóricos do reconhecimento de voz e do método difuso, aplicado ao reconhecimento de vogais. Depois de introduzido o tema e o método de resolução são discutidos os resultados obtidos.

## 2. Reconhecimento de Voz

Do ponto de vista Físico uma vogal é uma propagação do som pelo nosso sistema Fisiológico (meio), Glote, Boca, Vias Nasais e Garganta. É gerado um impulso (através dos músculos) que provoca uma perturbação no ar e esta propaga-se da glote à boca, nariz e garganta. [1],[9] e [10].

Têm sido feitos estudos de forma a modular esta propagação a partir de conceitos físicos e verifica-se que se pode modelar o meio de propagação como sendo um modelo físico de pequenos tubos com determinada impedância [2] e [10]. Estas considerações levam-nos a propor um modelo físico linear como aproximação. Assim, poder-se-á considerar o sistema como sendo linear e invariante com relação ao tempo (LTI) [9] e [10].

Considerando o nosso sistema como sendo LTI, este sistema fica descrito pela convolução de duas funções:

$$y[k] = h[k]*u[k] \quad (1)$$

onde  $y[k]$  é o output (resposta do sistema à perturbação inicial  $u[k]$ , sendo  $h[k]$  a função de transferência, normalmente inferida de parâmetros físicos conhecidos. Esta função de transferência dá-nos a relação entre o sinal captado (output) e o sinal inicialmente gerado (impulso). O facto deste impulso passar por um meio físico provoca uma alteração e essa alteração é dada por  $h[k]$ .

No caso em estudo só é conhecido  $y[k]$  à saída da boca, pois só usamos um microfone à saída da boca, não conhecemos nem  $h[k]$  nem  $u[k]$ . Existem trabalhos em que se tenta medir o  $u[k]$  e a partir de conceitos físicos inferir  $h[k]$ , [10]. Como os sinais recolhidos (voz) são digitais as nossas funções são séries temporais.

Assim, podemos seguir vários caminhos para a determinação de  $h[k]$ . Um deles é obter a função de transferência a partir de propriedades físicas inerentes ao sistema, [10]. Outro método usado em muitos trabalhos é o método Linear Predictive Coding (LPC) onde se aproxima o sinal  $y[k]$  à saída da boca como uma combinação linear dos valores de  $y[k]$  passados, como podemos ver em (2).

$$y(k) = \sum_{i=1}^p a_i \cdot y(k-i) + G \cdot u(k) \quad (2)$$

Podemos assim determinar  $h[k]$  com base na equação (1) e (2) no domínio das frequências aplicando a transformada-Z (paea uma descrição mais detalhada ver [1] e [9]):

$$\begin{cases} y(k) = \sum_{i=1}^p a_i \cdot y(k-i) + G \cdot u(k) & \text{(LPC)} \\ y(k) = h(k) * u(k) & \text{(LTI)} \end{cases}$$

Obtemos para a equação (2):

$$Y(z) = Z(y(k)) = Y(z) \cdot \sum_{i=1}^p a_i \cdot z^{-i} + G \cdot U(z) \quad (3)$$

e atendendo que a transformada-Z da convolução é:

$$Z(h[k] * u[k]) = Z(h[k]) \cdot Z(u[k]) \quad (4)$$

podemos obter:

$$Y(z) = H(z)U(z) \quad (5)$$

onde  $Y(z) = Z(y[k])$ ,  $U(z) = Z(u[k])$  e  $H(z) = Z(h[k])$  e:

$$H(z) = H \frac{1}{1 - \sum_{i=1}^p a_i \cdot z^{-i}} \quad (6)$$

Os coeficientes  $a_i$  podem ser determinados por métodos Autoregressivos (AR) [9]. Podendo assim escrever  $H(z)$  como a divisão de um polinómio complexo;

$$H(z) = 1/A(z) \quad (7)$$

Para a determinação dos coeficientes de  $A(z)$  usou-se o método AR de Burg, com um grau igual a  $p = fs/1000$ , onde  $fs$  é a frequência de amostragem, que neste caso foi de 11025Hz, logo  $p = 12$ , e através do estudo de  $H(z)$  determinar os parâmetros que nos permitam tomar uma decisão sobre a classificação da vogal.

Sabemos que uma perturbação ao propagar-se num sistema tubular provoca frequências de ressonância [9], que correspondem aos pólos de  $H(z)$ , isto é, aos zeros de  $A(z)$ . Essas frequências de ressonância costumam denominar-se por formantes [9], e correspondem aos picos máximos da envolvente do espectro de frequências do sinal.

Assim, foram usadas as três primeiras formantes, designadas por,  $F_0$ ,  $F_1$  e  $F_2$ , sendo estas os atributos considerados no problema multiatributo difuso para o reconhecimento de vogais. Note-se que a primeira formante costuma estar próxima da frequência fundamental do sinal.

Foram então desenvolvidos vários algoritmos em MATLAB, que nos permitissem determinar as três primeiras formantes.

- 1 - Módulo de aquisição do sinal e sua normalização;
- 2 - Módulo para calculo das envolventes das formantes de acordo com o método AR de Burg (designado por método 1);
- 3 - Módulo para cálculo das envolventes das formantes de acordo com o método AR de Ghael-Sandgathe [11], explicado em seguida (designado por método 2);

- 4 - Módulo para determinação dos Picos da envolvente que correspondem às formantes, para ambos os métodos;
- 5 - Módulo para tomada de decisão difusa multi-atributo.

Com base nos resultados da implementação destes algoritmos, verificou-se uma certa incerteza quanto a essa determinação, pois existem pólos que estão muito próximos e o algoritmo que determina os picos (máximos) da envolvente das formantes não os detecta, ou seja detecta uma média dos dois. Assim foi utilizado o método 2, de Ghael-Sandgathe, para uma melhor discriminação desses pólos e sua detecção, como podemos ver na figura 1.

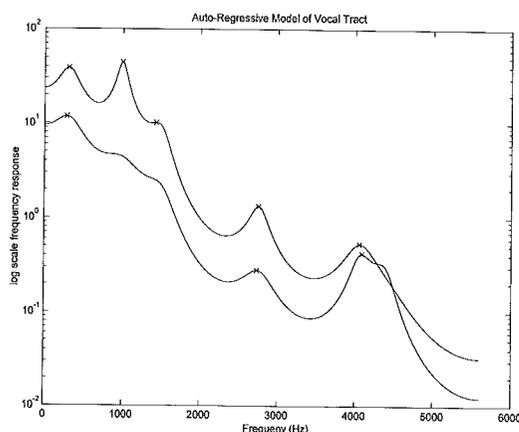


Figura 1- Dois picos detectados pelo método 2, de Ghael-Sandgathe (envolvente superior) que não são detectados pelo método 1 (envolvente inferior).

O método 2 aqui utilizado (Ghael-Sandgathe [11]) consiste em puxar pólos da zona exterior à do raio de convergência da transformada-Z, para o seu interior, recalculando assim os coeficientes do polinómio complexo  $A(z)$  (ver com melhor detalhe [[11]).

### 3. Decisão Multiatributo Difusa

Apresentaremos neste parágrafo uma breve introdução de lógica difusa (Fuzzy Logic) e de seguida a descrição do método de decisão multiatributo difuso (Multiple Attribute Fuzzy Decision Making), usado neste trabalho.

#### 3. 1 Breve descrição da lógica difusa

Em muitos casos em que temos um problema a modelar consideramos parâmetros que não podem ser definidos por um conjunto booleano (ou Crisp) bem delimitado, a tais conjuntos chamamos de difusos (ou Fuzzy) [3]. Consideremos um exemplo; quando pretendemos saber se um grupo de pessoas são altas ou baixas, podemos por exemplo dizer que uma pessoa é alta se a sua altura pertence ao conjunto  $C1 = [1,80; 2,00]$ m, e baixa se pertencer ao conjunto  $C2 = [1,00; 1,80]$ m. Quando fazemos tal afirmação estamos a considerar que em conjuntos booleanos (ou Crisp), uma pessoa que tenha altura de 1,79m é uma pessoa baixa e uma pessoa com 1,80m é alta. Neste tipo de abordagem uma pessoa com 1,00m é tão baixa como uma pessoa

com 1,79m, quando esta última se aproxima mais de uma pessoa alta do que baixa. Para resolver este tipo de problemas, a utilização de conjuntos difusos em vez de crisp é apropriada. Assim, podemos considerar o mesmo tipo de problema em lógica difusa da seguinte forma: consideremos um conjunto  $D1 = [1,60; 2,00]$  mas onde a cada altura corresponde uma determinada graduação ou grau de pertença  $\mu$ , em que  $\mu$  é uma função dos valores de  $D1$ , isto é, é a função  $\mu(x) : D1 \rightarrow [0,1]$  (normalizada). Podemos ver na figura 2 um exemplo, onde uma pessoa é 0.5 alta (mais ou menos alta) para uma altura de 1,70m e 0.25 alta se tiver uma altura de 1,65m (pouco alta).

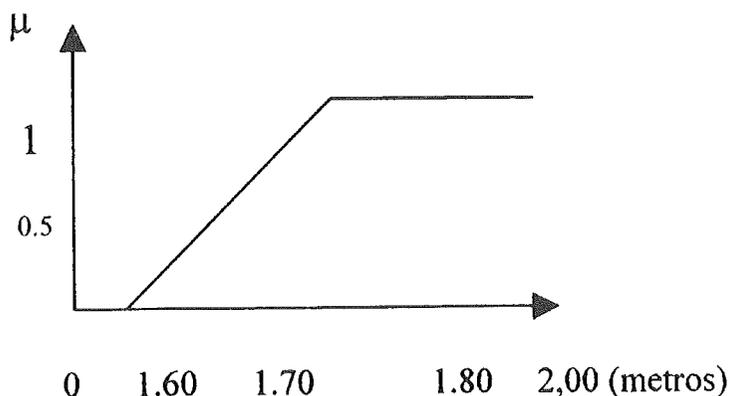


Figura 2 - Conjunto difuso "alto"

É de notar que podemos considerar um conjunto Crisp como um caso particular dum conjunto Fuzzy em que a função de pertinência toma valores em  $\mu(x) : C \rightarrow \{0,1\}$ .

Tal como na lógica booleana, na lógica difusa existem operadores de agregação entre proposições, tais como  $\wedge$ ,  $\vee$ , e  $\Rightarrow$ , para definirem as operações de intersecção, união, e implicação [3]. Os operadores de agregação mais comuns caem nas famílias das t-normas e t-conormas para, respectivamente, a intersecção e união (para mais tipos de operadores de agregação ver referência [3]). As t-normas e t-conormas são compostas por vários operadores sendo os mais conhecidos os seguintes: sejam A e B dois conjuntos difusos com funções de pertinência  $\mu_A$  e  $\mu_B$ , respectivamente,

t-normas

$$\mu_{A \wedge B} = \mu_A \cdot \mu_B$$

$$\mu_{A \wedge B} = \min(\mu_A, \mu_B)$$

t-conormas

$$\mu_{A \vee B} = \mu_A + \mu_B - (\mu_A * \mu_B)$$

$$\mu_{A \vee B} = \max(\mu_A, \mu_B)$$

(8)

Neste trabalho os operadores de agregação escolhidos foram o produto para a intersecção e o max para a união, pois empiricamente parecem ser mais apropriados para a classificação e reconhecimento das vogais. Note-se que mais estudos seriam necessários para garantir que outros operadores, como os paramétricos ou geométricos [3], não teriam melhor comportamento no reconhecimento das vogais.

### 3.2. Método Multiatributo Difuso

O método de decisão utilizado no reconhecimento de vogais, foi o multiatributo difuso (Multiple Attribute Fuzzy Decision - MAFD) [6]. Quando falamos de tomada de decisão multiatributo difusa [7], estamos a falar de problemas onde temos um conjunto de alternativas, conhecidas, as quais são classificadas com base em atributos/critérios pré-definidos. O processo de classificação é feito pela atribuição de valores a cada atributo, em relação a cada alternativa, que representam o nível de satisfação do atributo/critério para uma dada alternativa. No caso difuso, os atributos correspondem a conjuntos difusos, e portanto usam-se valores de pertença para definir o nível de satisfação dos atributos.

Depois agregam-se os atributos para cada alternativa, com um operador de agregação, para determinar a classificação final de cada alternativa. Este processo corresponde à intersecção dos atributos com o operador produto. Finalmente as alternativas são ordenadas e a que tiver melhor classificação é escolhida como óptimo. Este processo usa o operador max, das t-conormas. Neste trabalho as alternativas são as vogais (a,e,i,o,u), e os atributos considerados são as formantes fuzzificadas. Cada vogal tem um valor de pertença para os atributos que representa o grau de satisfação atingido na formante.

Usaram-se dois conjuntos de alternativas, as primeiras sendo as vogais e as segundas os dois métodos de cálculo das envolventes das formantes, o método 1 com os três primeiros picos da envolvente inicial e o método 2 com os três primeiros picos da envolvente determinada. A classificação das vogais é feita a partir das duas primeiras formantes, F1 e F2. Feita uma análise prévia dos dados, verificou-se uma certa dificuldade em determinar as duas primeiras formantes, pois os métodos usados por vezes não conseguem diferenciar bem as três primeiras formantes F0, F1 e F2, detectando um único pico que será um ponto que representa F0 e F1. Por forma a contornar este problema considera-se então para cada método os dois primeiros picos das envolventes como primeira e segunda formante respectivamente e um outro par que considera como primeira formante e segunda formante o segundo pico e o terceiro da envolvente, respectivamente, por forma a ultrapassar essa incerteza. Após a consideração anterior podemos ver na figura 3 a distribuição dos dados com todos os exemplos. Na figura 4 apresenta-se o mesmo estudo mas na banda 1.5kHzx3.5kHz onde se vê melhor as zonas de maior densidade onde se encontram as vogais. Comparando os resultados da figura 4 com outro trabalho realizado na área [4], figura 5, verifica-se que as zonas de maior densidade se aproximam das dos resultados obtidos no outro trabalho (figura 5).

Especificamente, os atributos considerados para cada uma das alternativas (a,e,i,o,u) foram a fuzzificação dos parâmetros F1 e F2. De acordo com a distribuição apresentada na Figura 4, que representam os dados obtidos experimentalmente e que representam o par ordenado (F1,F2) que caracteriza cada uma das vogais, e por comparação com a distribuição da Figura 5, de um outro trabalho já realizado (ver referência [4]), achou-se conveniente fazer como aproximação o uso de funções de pertença trapezoidais e triangulares.

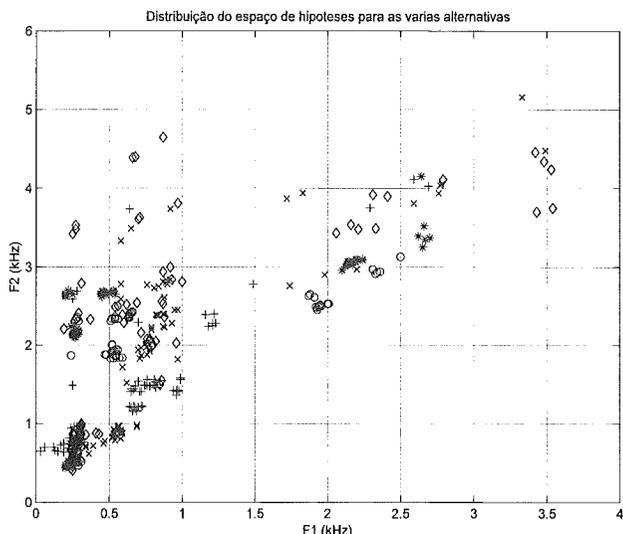


Figura 3 - Estudo experimental com todos os exemplos  
 Legenda: 'a' = '+', 'e' = 'o', 'i' = '\*', 'o' = 'x' e o 'u' = '◆'

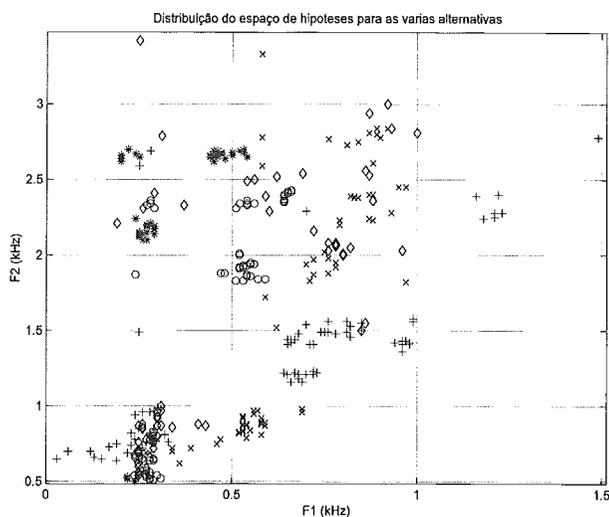


Figura 4 - Estudo experimental com exemplos na banda 1.5kHzx3.5kHz

Legenda: 'a' = '+', 'e' = 'o', 'i' = '\*', 'o' = 'x' e o 'u' = '◆'

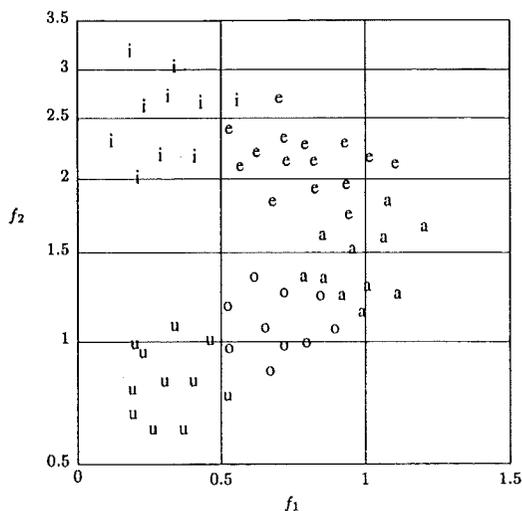


Figura 5 - Resultados de outro estudo[4]

Legenda: Frequências das duas primeiras formantes em kHz.

A tabela 1 apresenta os conjuntos difusos correspondentes às funções triangulares e trapezoidais considerados para as formantes F1 e F2. Esta tabela foi estabelecida por inspeção dos dados, de acordo com a análise das distribuições referenciadas anteriormente e apenas aponta os pontos extremos das funções i.e. os pontos onde os  $\mu$  são 0 ou 1. As funções foram criadas de forma a dar maior grau de pertença às zonas de maior densidade da vogal, tentando-se arranjar um compromisso entre os dados obtidos experimentalmente, figura 4, e os dados da figura 5 [4]. É de dar especial atenção à função que representa o 'u', pois esta é triangular pelo facto de existir nessa zona grande densidade de dados de outras vogais, conforme se pode ver

na figura 4. Este resultado deve-se ao facto de se ter incerteza em relação à consideração das duas primeiras formantes, conforme já explicado anteriormente.

A	$\mu_{F1}$	$\mu_{F2}$
a	Trap[0.5 0.7 0.9 1.4]	Trap[0.8 1.2 1.5 1.7]
e	Trap[0.3 0.6 0.7 1.0]	Trap[1.5 1.9 2.3 2.7]
i	Trap[0.15 0.3 0.4 0.6]	Trap[1.9 2.3 2.4 3.1]
o	Trap[0.4 0.55 0.65 0.8]	Trap[0.6 0.8 0.9 1.3]
u	Triag[0.1 0.3 0.6]	Triag[0.4 0.6 1.1]

Tabela 1- Funções de Pertença

Legenda: os valores apresentados são em kHz.

Por exemplo, Trap[0.5 0.7 0.9 1.4] representa a função trapezoidal apresentada na figura 6.

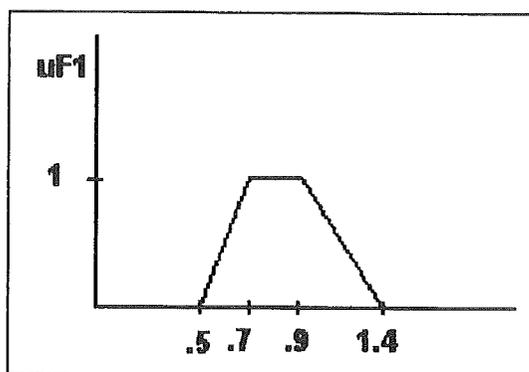


Figura 6 - Exemplo de uma função de pertinência

Como referido, neste trabalho são utilizados dois tipos de operadores de agregação  $\max()$  para a selecção das alternativas e  $\text{produto}()$  para combinação dos atributos. Muitos outros operadores poderiam ter sido usados como se viu em 3.1 e se poderá ver em [3]. De seguida passa-se a descrever o processo pelo qual foi desenvolvido o trabalho.

Assim, para cada um dos métodos, método 1 e 2 para determinação das formantes, o cálculo para a agregação foi feito seguindo os seguintes passos:

- (1) para cada envolvente das formantes escolhem-se dois vectores  $v_1 = [F1 \ F2]$ , tendo como coordenadas, os dois primeiros picos da envolvente e  $v_2 = [F1' \ F2']$ , tendo como coordenadas o segundo e terceiro pico da mesma envolvente.
- (2) para cada uma das alternativas calcula-se o grau de pertinência dos vectores de acordo com a tabela-1,  $\mu_{F1}$ ;  $\mu_{F2}$ ;  $\mu_{F1'}$ ;  $\mu_{F2'}$ .
- (3) depois faz-se a conjugação das coordenadas, usando o operador de agregação

$$\begin{cases} v_1 = \cap(\mu_{F_1}, \mu_{F_2}) = \mu_{F_1} \cdot \mu_{F_2} \\ v_2 = \cap(\mu_{F'_1}, \mu_{F'_2}) = \mu_{F'_1} \cdot \mu_{F'_2} \end{cases}$$

- (4) a seguir procede-se à sua disjunção, usando o operador de agregação max;  $\mu_{A_{i1}} = \max(v_1 \cdot v_2)$  e assim obtém-se o grau de pertença da primeira envolvente, método 1, para cada uma das alternativas (a,e,i,o,u). Repete-se o mesmo processo para o método 2, correspondente à segunda envolvente e obtemos,  $\mu_{A_{i2}} = \max(v_1 \cdot v_2)$
- (5) finalmente agregam-se os graus de pertença obtidos em (4) e (5) resultando num novo grau de pertença,  $\mu_{A_i} = \max[\mu_{A_{i1}} \mu_{A_{i2}}]$ . A decisão óptima sobre qual a vogal que foi reconhecida é então obtida de acordo com:

$$D = \max(\mu_{A_i})$$

A título de exemplo temos:

Obtemos para uma determinada pessoa os 3 primeiros picos das envolventes que poderão corresponder às 3 primeiras formantes, do sinal da vogal 'a':

Método 1	Método 2
F1 = 0.301 kHz	F1 = 0.280 kHz
F2 = 0.980 kHz	F2 = 2.692 kHz
F3 = 1.410 kHz	F3 = 4.027 kHz

Tabela 2 - Três primeiros picos das envolventes

Com estes valores são formados os vectores com as respectivas coordenadas, segundo o passo (1)

Método 1	Método 2
$v_1 = [0.301; 0.980]$	$v_1 = [0.280; 2.692]$
$v_2 = [0.980; 1.410]$	$v_2 = [2.692; 4.027]$

Tabela 3 - Aplicação do primeiro passo

Para cada alternativa são calculados os valores de pertença para cada formante considerada, segundo o passo (2), para cada um dos métodos usados.

Método 1				
Vogal	$\mu_{F_1}$	$\mu_{F'_1}$	$\mu_{F_2}$	$\mu_{F'_2}$
a	0	0.45	0.84	1
e	0.003	0	0.7	0
i	1	0	0	0
o	0	0.8	0	0
u	1	0.24	0	0

Tabela 4 - Aplicação do segundo passo ao método 1

Método 2				
Vogal	$\mu F_1$	$\mu F_1$	$\mu F_1'$	$\mu F_2'$
a	0	0	0	0
e	0	0.02	0	0
i	0.87	0.58	0	0
o	0	0	0	0
u	0.9	0	0	0

Tabela 5 - Aplicação do segundo passo ao método 2

Usando o operador de intersecção produto, os atributos são agregados de forma a resultar uma classificação por alternativa, segundo o passo (3).

Vogal	Método 1		Método 2	
	$v_1$	$v_2$	$v_1$	$v_2$
a	0	0.84	0	0
e	0	0	0	0
i	0	0	0.51	0
o	0	0	0	0
u	0.24	0	0	0

Tabela 6 - Aplicação do terceiro passo

Depois procede-se à selecção das alternativas, em cada método, usando o operador max, segundo os passo (4) e (5).

Método 1 ( $\mu A_1$ )				
a	e	i	o	u
0.84	0.00	0.00	0.00	0.24

Tabela 7 - Aplicação do quarto e quinto passo ao método 1

Método 2 ( $\mu A_2$ )				
a	e	i	o	u
0.00	0.00	0.51	0.00	0.00

Tabela 8 - Aplicação do quarto e quinto passo ao método 2

Finalmente procede-se à escolha do óptimo, ou seja do melhor valor obtido nos dois métodos, segundo o passo (6).

$\mu A_I$				
a	e	i	o	u
0.84	0.00	0.51	0.00	0.24

Tabela 9 - Aplicação do sexto passo

Então a vogal com melhor reconhecimento é:

**'a' com um valor de reconhecimento de 0.84**

Este método é bastante interessante pois permite ter uma medida de comparação entre as várias alternativas, que com alguma análise posterior poderá ser utilizado no reconhecimento do orador.

Fazendo uma breve análise comparativa entre o método 1 e 2, os resultados de sucesso parciais são apresentados na tabela 10.

Alternativa	Método 1	Método 2
a	52%	48%
e	43%	57%
i	50%	50%
o	54%	46%
u	80%	20%

Tabela 10 - Comparação entre métodos

Verificamos que os métodos se complementam de forma a atingir um grau de sucesso razoável. O resultado mais díspar, é a vogal *ëuí*, no entanto não se avançam de momento justificações pois pensamos fazer um estudo estatístico mais aprofundado no futuro.

#### 4. Resultados

Foram recolhidos dados de quatro pessoas, dois Homens e duas Mulheres, de idades compreendidas entre os 30 e 38 anos, para cada um dos indivíduos foram recolhidos 15 amostras de cada vogal, perfazendo um total de 300 dados que correspondem a 600 exemplos. A frequência de amostragem, foi de  $f_s = 11025\text{Hz}$ , com uma amplitude de 16bits. Para a aquisição dos dados (sons) foi utilizado o software "Sound Forge v4.5".

Note-se que neste tipo de modulação é importante o factor idade, pois uma criança tem uma fisiologia diferente de um adulto e o mesmo para grandes variações de idade. Estas foram as razões da escolha de população efectuada.

Os resultados obtidos, tanto para os resultados correctos como para os mal classificados, utilizando o método multi-atributo difuso são apresentados na tabela 11 e 12.

Alternativa	% Resultados Correctos	Maior Valor de Pertença	Menor Valor de Pertença
a	100%	1.00	0.57
e	58%	0.88	0.59
i	87%	0.67	0.25
o	62%	1.00	0.0
u	45%	0.81	0.0

Tabela 11 - Tabela de resultados correctos

Na tabela 12 apresentam-se os graus de pertença dos dois primeiros classificados, no caso do exemplo ser mal classificado, observando-se que se usou como elementos representativos as classificações em que o primeiro classificado tinha maior grau de pertença.

Alternativa	Resultado mal classificado com as duas melhores classificações com maior grau de pertença para o primeiro classificado
a	N/A
e	0.89/u+0.88/e; 0.94/i+0.66/e
i	0.52/u+0.42/i
o	0.74/u+0.63/o; 1.0/e+0.68/u; 0.75/e+0.43/o
u	1.0/e+0.73/u; 0.92i; 0.44/i+0.34/u

Tabela 12 - Tabela de resultados mal classificados

Na tabela 11, o resultado alto da vogal 'a', justifica-se pelo facto de se encontrar numa região onde F1 e F2 são altos, logo mais fácil a detecção dos picos das formantes e não se dispersa muito. Em relação ao 'u' era de esperar um valor baixo pois este situa-se em zonas de muito baixa frequências.

É curioso observar a dispersão dos dados na figura 3, e comparar com a tabela de resultados tabela 11, pois inicialmente os resultados pareciam bastante dispersos e a decisão parecia ser difícil. No entanto, através do método aplicado, a decisão mostrada estatisticamente na tabela 11, revela um bom poder de decisão, o que mostra que o método poderá ser eficaz para este género de problema.

Como se pode ainda ver na tabela 11, os valores de pertença obtidos também apresentam alguma disparidade. Enquanto para a vogal "a" foi de 1.00 o valor máximo de pertença, para a vogal "e" foi de 0.88, vogal "i" de 0.67, vogal "o" de 1.00 e para a vogal "u" de 0.81. Comparando os valores de pertença máximos com os valores estatísticos verifica-se que para que todas as vogais à excepção do "a" se encontram um pouco dispersas como se pode ver nas figuras 3 e 4. Para melhorar estes resultados deveriam ser considerados os seguintes aspectos: utilizar novas medidas de decisão; melhorar o algoritmo de detecção de picos da envolvente; ou melhorar a qualidade da aquisição dos dados. No entanto, dadas as limitações de hardware, o método de decisão apresentado parece ser bastante viável face ao ruído do sinal. A solução para melhores resultados depende mais do método utilizado para a detecção dos picos da envolvente e será necessária uma análise estatística mais aprofundada, considerando uma amostra maior, médias de reconhecimento, desvios padries etc., para se poderem tirar mais ilações sobre os valores de pertença obtidos e qual o seu significado neste contexto. Isto fará parte de trabalho futuro.

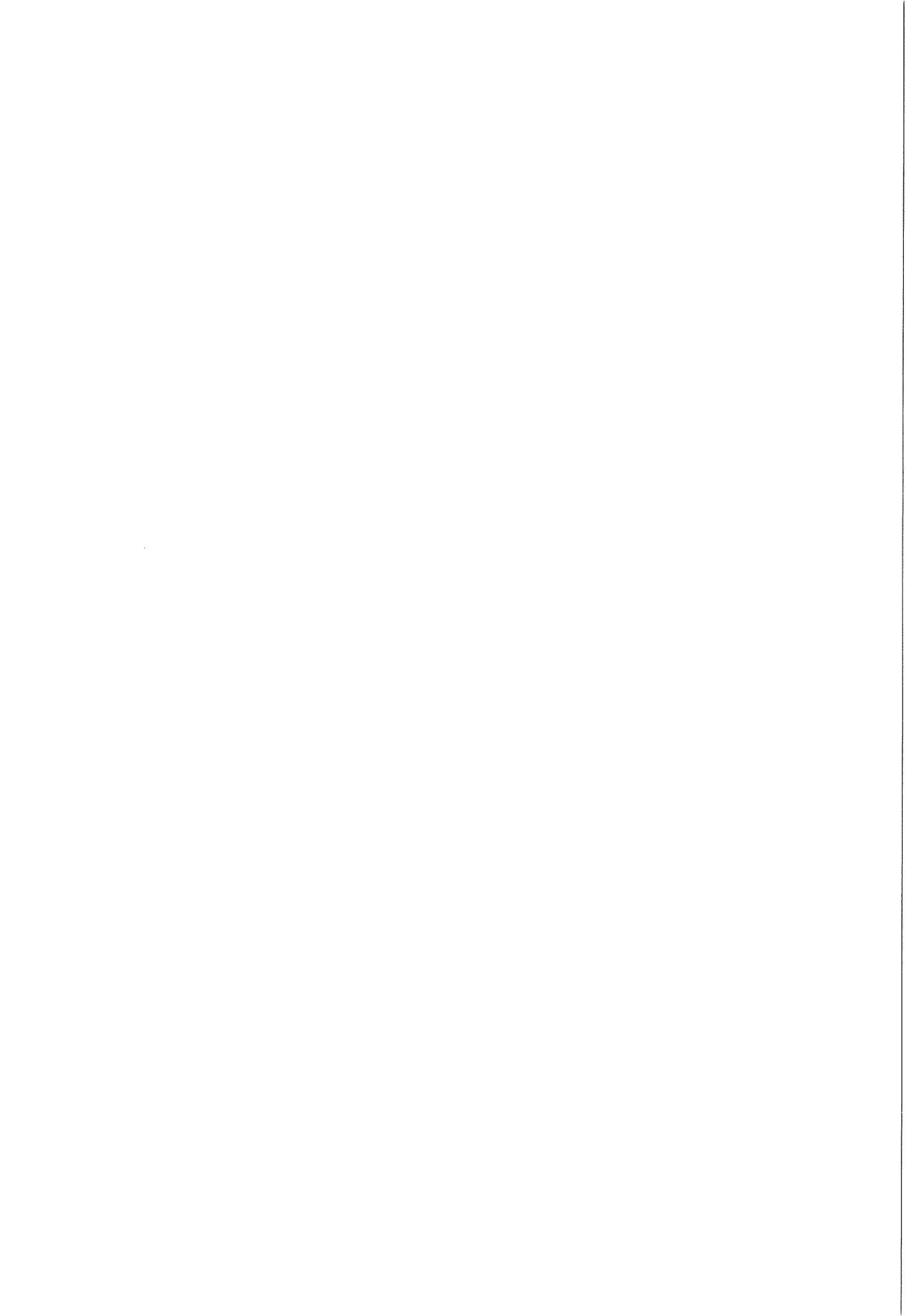
## 5. Conclusões

Este trabalho representa uma primeira experiência do uso de um método de decisão difuso em problemas de reconhecimento de vogais Portuguesas. Para ser feito um estudo mais rigoroso, esta aproximação terá de aumentar significativamente o número de dados, e considerar no mínimo 30 pessoas, de varias idades, e uma maior quantidade de dados por pessoa, o que não foi possível neste primeiro trabalho.

Contudo, pode-se concluir que o método multi-atributo de decisão difuso parece ter grande potencial para o reconhecimento de voz, uma vez que é capaz de lidar, de uma forma simples, com a incerteza intrínseca deste tipo de problemas. O exemplo apresentado assim o demonstra.

## Referências

- [1] Chen, C. T., System and signal analysis (2<sup>TM</sup>ED), Saunders Colledge Publishing, New York (1994).
- [2] Halliday, D., Resnick, R., Fundamentals of physics (3<sup>TM</sup>ED), John Wiley & Sons, New York (1988).
- [3] Klir, G.J. e T.A. Folger, Fuzzy sets, uncertainty, and Information, Prentice-Hall, New York (1988).
- [4] Marques, J.S., Reconhecimento de padrões, IST Press, Portugal (1999).
- [5] Ribeiro, R. A., Fuzzy Evaluation of Thermal Quality Of Buildings, Computer-Aided Civil and Infrastructure Engineering 14 (1999) 155-162.
- [6] Ribeiro, R. A., Baldwin, J F., A Multiple Attribute Fuzzy Decision Support System: Two Applications, Fuzzy logic and Soft computing (1995) 452-461.
- [7] Ribeiro, R. A., Fuzzy multiple attribute decision making: A review and new preference elicitation techniques, Fuzzy Sets and Systems 78 (1996) 155-181.
- [8] Sankar, K. P., Dwijesh, D M., Fuzzy sets and decision making approaches in vowel and speaker recognition, IEEE Trans. Systems, Man and Cybernetics (1977) 625-629.
- [9] Shaughnessy, D. O., Speech Communication Human and Machine, Addison-Wesley, New York (1990).
- [10] <http://speech.llnl.gov/thesis/index.htm>
- [11] <http://umamitoba.ca/linguistics/russell/138/sec4/formants.htm>



## INSTRUÇÕES AOS AUTORES

Os autores que desejam submeter um artigo à Investigação Operacional devem enviar três cópias desse trabalho para:

Prof. Joaquim J. Júdice  
Departamento de Matemática  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra  
Apartado 3008  
3001-454 Coimbra, Portugal

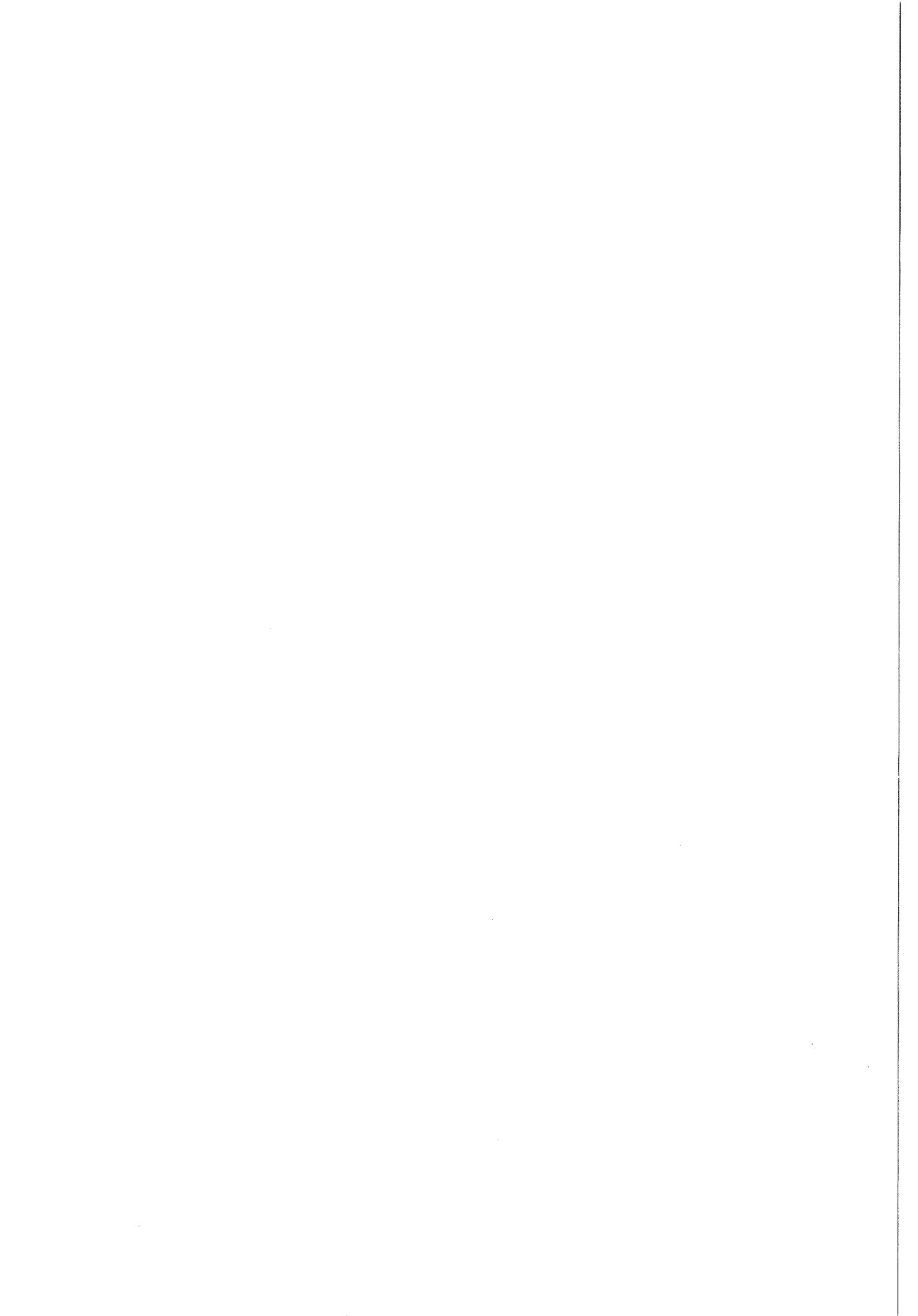
Os artigos devem ser escritos em Português ou Inglês. A primeira página deve conter a seguinte informação:

- Título do artigo
- Autor(es) e instituição(ões) a que pertence(em)
- Abstract (em inglês)
- Resumo
- Keywords (em inglês)
- Título abreviado

As figuras devem aparecer em separado de modo a poderem ser reduzidas e fotocopiadas. As referências devem ser numeradas consecutivamente e aparecer por ordem alfabética de acordo com os seguintes formatos:

Artigos: autor(es), título, título e número da revista (livro com indicação dos editores), ano, páginas.

Livros: autor(es), título, editorial, local de edição, ano.



## ÍNDICE

C. Oliveira, C. Henggeler Antunes, Um modelo multiobjectivo para planeamento energético baseado na Análise Input-Output - Aplicação ao caso português .....	3
C. Gomes da Silva, J. Figueira, J. Lisboa, Análise multiobjectivo no planeamento agregado da produção .....	21
A. I. Vaz, E. M. Fernandes, M. P. Gomes, Discretization methods for semi-infinite programming .....	37
E. Q. Martins, M. M. Pascoal, J. L. Santos, A new improvement for a K shortest paths algorithm .....	47
T. Gomes, L. Martins, J. Craveirinha, An algorithm for calculating K shortest paths with a maximum number of arcs .....	61
F. Alvelos, J. Valério de Carvalho, Aplicação do método de partição e geração de colunas ao problema do fluxo multicomodidade .....	71
T. M. Simas, R. A. Ribeiro, Reconhecimento de vogais utilizando um método de decisão difuso .....	93

