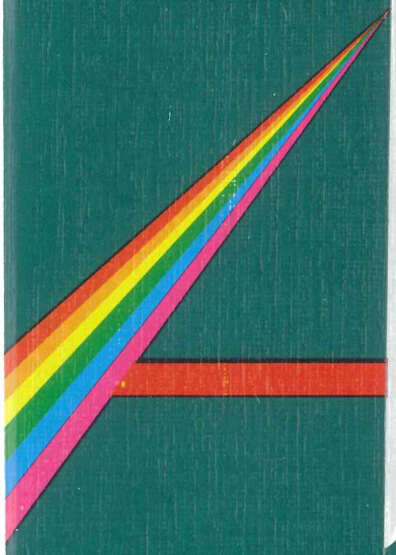
INVESTIGAÇÃO OPERACIONAL

Dezembro 1996
Número 2
Volume 16



Publicação Científica da

Associação Portuguesa para o Desenvolvimento
da Investigação Operacional

INVESTIGAÇÃO OPERACIONAL

Propriedade:
APDIO — Associação Portuguesa para o Desenvolvimento da Investigação Operacional

ESTATUTO EDITORIAL

«Investigação Operacional», órgão oficial da APDIO cobre uma larga gama de assuntos reflectindo assim a grande diversidade de profissões e interesses dos sócios da Associação, bem como as muitas áreas de aplicação da I. O. O seu objectivo primordial é promover a aplicação do método e técnicas da I.O. aos problemas da Sociedade Portuguesa.
A publicação acolhe contribuições nos campos da metodologia, técnicas, e áreas de aplicação e software de I. O. sendo no entanto dada prioridade a bons casos de estudo de carácter eminentemente prático.

Distribuição gratuita aos sócios da APDIO

INVESTIGAÇÃO OPERACIONAL

Volume 16 - nº 2 - Dezembro 1996 Publicação semestral

Editor Principal: Joaquim J.Júdice
Universidade de Coimbra

Comissão Editorial

A Revista "INVESTIGAÇÃO OPERACIONAL" está registada na Secretaria de Estado da Comunicação Social sob o nº 108335.
Esta Revista é distribuida gratuitamente aos sócios da APDIO. As informações sobre inscrições na Associação, assim como a correspondência para a Revista devem ser enviadas para a sede da APDIO - Associação Portuguesa para o Desenvolvimento da Investigação Operacional - CESUR, Instituto Superior Técnico, Av. Rovisco Pais, 1000 Lisboa.

Este Volume foi subsidiado por :

Junta Nacional de Investigação Científica e Tecnológica (JNICT)

Fundação Calouste Gulbenkian

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

Assinatura: 5.000$00

INVESTIGAÇÃO OPERACIONAL, vol. 16 (Dezembro 1996) 89-114

UMA ANÁLISE COMPARATIVA DE FORMULAÇÕES PARA O PROBLEMA DO CAIXEIRO VIAJANTE

Luís Gouveia

DEIO
Faculdade de Ciências
Universidade de Lisboa
Bloco C2-Piso 2, Campo Grande
1700 Lisboa - Portugal

José Manuel Pires

ISCAL
Av. Miguel Bombarda, 20
1100 Lisboa - Portugal

Abstract

Several formulations for the Traveling Salesman Problem are compared with respect to their Linear Programming (LP) relaxations. In particular, we present a survey of several techniques used for tightening the LP relaxation of a given model.

Resumo

Neste trabalho discutem-se várias formulações em Programação Linear Inteira para o problema do Caixeiro Viajante. O objectivo principal deste trabalho é comparar as relaxações em Programação Linear das formulações apresentadas e mostrar diferentes maneiras de analisar e melhorar tais relaxações.

Keywords

Travelling Salesman Problem, Linear Programming Relaxations, Reformulations.

1. Introdução

Seja $G = (V, A)$ um grafo orientado onde $V = \{1, 2, \ldots, n\}$ denota o conjunto dos nodos e A o conjunto dos arcos. A cada arco (i, j) está associado um custo c_{ij}. O problema do caixeiro viajante consiste na determinação, em G, do circuito hamiltoniano (circuito que passa por cada nodo uma e uma só vez) de custo mínimo (o custo de um circuito corresponde à soma dos custos dos arcos que o constituem). Se o custo associado a cada arco (i, j) é igual ao custo associado ao arco inverso (j, i), o problema é denominado de caixeiro viajante simétrico (PCV). Caso contrário, o problema toma a designação de caixeiro viajante assimétrico (PCVA).

O problema do caixeiro viajante é, à semelhança de outros problema de optimização combinatória, um problema NP-difícil (ver, por exemplo, Johnson e Papadimitriou (1985)).

Uma maneira de produzir um limite inferior para o custo da solução óptima consiste em resolver a relaxação em programação linear de uma formulação em programação linear inteira (PLI) para este problema. A qualidade desse limite depende da relaxação em programação linear da formulação utilizada. Assim, diversas formulações têm sido propostas para a resolução do problema do caixeiro viajante e, sempre que possível, tem-se tentado caracterizar a qualidade da correspondente relaxação em programação linear.

Dependendo das variáveis envolvidas, as formulações em programação linear inteira para o PCVA, e, de um modo geral, para os problemas de optimização combinatória, podem ser agrupadas em duas classes: a das formulações "naturais" e a das formulações "estendidas". Uma definição formal do que se entende por formulação natural ou, alternativamente, por formulação estendida de um problema de optimização combinatória pode ser encontrada em Pulleyblank (1989). Informalmente, uma formulação para o PCVA é chamada formulação natural se contém uma variável e apenas uma para cada arco incluído no grafo subjacente. Por outro lado, é denominada formulação estendida se usa outras variáveis que podem estar ou não associadas aos arcos. Por exemplo, uma conhecida formulação estendida para o PCVA (ver Miller-Tucker-Zemlim (1960)) inclui um conjunto de variáveis associadas aos nodos. Estas variáveis adicionais podem ser consideradas como variáveis supérfluas no sentido de que não são necessárias para obter uma formulação válida para o problema. Contudo, a informação adicional associada às novas variáveis pode reduzir consideravelmente o número de restrições envolvidas. Geralmente, o uso de variáveis adicionais permite derivar formulações compactas (isto é, formulações que envolvem um número polinomial de restrições e variáveis). Por projecção, no subespaço das variáveis naturais, do conjunto das soluções admissíveis da relaxação em programação linear de uma formulação estendida, é possível obter uma formulação equivalente que utiliza apenas variáveis naturais. Contudo, tais formulações naturais envolvem usualmente um número exponencial de restrições.

Neste trabalho apresentam-se formulações naturais e formulações estendidas para o PCVA. Simultaneamente, é feita uma comparação entre a qualidade das respectivas relaxações em programação linear e, em particular, dar-se-à ênfase ao conceito de reformulação, isto é, a técnicas que permitem transformar um modelo para o PCVA em programação linear inteira noutro cuja relaxação linear seja mais forte do que a do modelo anterior, ou que seja equivalente mas envolvendo um conjunto diferente de variáveis que permita mais facilmente a derivação de novas desigualdades válidas para o PCVA. Assim, na secção 2 apresentam-se algumas formulações naturais para o PCVA. Na secção 3 apresentam-se formulações estendidas baseadas em modelos de fluxos em rede, onde se discutem os modelos de fluxo agregado e os modelos de fluxo desagregado. Usando técnicas de projecção, caracterizam-se as relaxações em programação linear dos modelos de fluxo no subespaço das variáveis naturais. Na secção 4 apresentam-se formulações estendidas que incluem variáveis adequadas a uma generalização do PCVA com dependências temporais. Mostra-se também que os modelos de fluxo agregado

podem ser vistos como modelos adequados a esta generalização do PCVA. Na secção 5 faz-se uma breve referência a um problema muito semelhante ao PCVA, o problema do caixeiro viajante com custos cumulativos (PCVC), e focam-se algumas questões em aberto relativamente a este último. Alguns resultados computacionais são apresentados na secção 6. Por último, na secção 7 apresentam-se as principais conclusões.

No que se segue, dada uma formulação P, P_L designa a correspondenre relaxação em programação linear de P e $v(P)$ o custo da solução óptima de P.

2. Formulações naturais

Muitas das formulações em programação linear inteira existentes para o PCVA usam variáveis binárias x_{ij} para indicar se o arco (i, j) está ou não na solução óptima. Tais modelos são baseados no seguinte esquema (ver, por exemplo, Langevin, Soumis e Desrosiers (1990)):

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \qquad (1)$$

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

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,\ldots, n \qquad (2b)$$

$$\{(i,j) : x_{ij} = 1;\ i,j = 2,\ldots, n\} \qquad \text{não contém suncircuitos} \qquad (3)$$

$$x_{ij} \in \{0,1\} \qquad i,j = 1,\ldots, n \qquad (4)$$

Para facilitar a indexação, as variáveis x_{ii} $(i = 1,\ldots, n)$ não são consideradas neste modelo. Com o mesmo objectivo, assume-se que o problema está definido num grafo completo, isto é, para cada par i e j $(i \neq j)$ existem os arcos (i,j) e (j,i). As restrições (2a), (2b) e (4) definem um problema de afectação. A inclusão das restrições (3) deve-se à necessidade de evitar a existência de soluções admissíveis que incluam vários subcircuitos. Note-se que não é necessário incluir explicitamente restrições que eliminam subcircuitos envolvendo o nodo 1. As restrições (2a) e (2b) garantem que se existe um subcircuito envolvendo o nodo 1, então tem que existir pelo menos outro subcircuito que não inclui esse nodo. Mas estes subcircuitos são eliminados pelas restrições (3). Estas restrições podem ser apresentadas de várias maneiras, permitindo assim a obtenção de diversas formulações para o PCVA. Com o objectivo de facilitar a notação sejam

$$X(S) = \sum_{i \in S} \sum_{j \in S} x_{ij} \quad \text{e} \quad X(S_1,S_2) = \sum_{i \in S_1} \sum_{j \in S_2} x_{ij}\ .$$

Possivelmente, a representação mais conhecida das restrições (3) é dada pelas desigualdades:

$$X(S) \leq |S| - 1 \quad \forall S \subseteq \{2,\ldots, n\} \text{ e } |S| \geq 2 \qquad (5)$$

Para se verificar que (5) impede a formação de subcircuitos, basta considerar um subcircuito que envolva um subconjunto de nodos $S \subseteq \{2,\ldots, n\}$ e concluir que a soma dos x_{ij} para todas as ligações desse subcircuito é igual a $|S|$. Uma vez que se pretende eliminar este tipo de solução, impõe-se que a referida soma inferior a $|S|$. Estas desigualdades são usualmente conhecidas por restrições de eliminação de subcircuitos, precisamente pelo que garantem quando incluídas em formulações de problema de optimização combinatória bem conhecidos como, por exemplo, o PCVA aqui tratado.

A formulação (1), (2a), (2b), (5) e (4) é bem conhecida na literatura de Optimização Combinatória e é devida a Dantzig, Fulkerson e Johnson (1954). Ao longo deste trabalho, esta formulação será designada por *SUB*, precisamente por ser caracetrizada pelas restrições de eliminação de subcircuitos (5). Trata-se de uma formulação natural uma vez que envolve apenas as variáveis x_{ij}.

As restrições (3) podem também ser modeladas através do seguinte conjunto de restrições:

$$X(S^C,S) \geq 1 \qquad \forall S \subseteq \{2,\ldots, n\} \text{ e } |S| \geq 2 \qquad (6)$$

em que $S^C = V \setminus S$. É fácil de verificar que uma solução admissível para (2a), (2b) e (4) contém subcircuitos se e só se é desconexa. As restrições (6) garantem que tal solução tem que ser conexa. Estas restrições são usualmente designadas por restrições de corte, por garantirem a inclusão de pelo menos um arco do corte $[S^C,S\,]$ em qualquer solução admissível do PCVA. No que se segue denotar-se-á por *CORTE* o modelo designado por (1), (2a), (2b), (6) e (4).

Na presença das restrições (2a) ou (2b) é fácil verificar que (5) e (6) são equivalentes. Com efeito, adicionando (2a) para $j \in S$ e $S \subseteq \{2,\ldots, n\}$ tem-se que $X(V,S) = |S|$, o que é equivalente a ter $X(S) + X(S^C,S\,) = |S|$. A igualdade anterior garante que se tem $X(S) \leq |S| - 1$ se e só se $X(S^C,S\,) \geq 1$ para todo o $S \subseteq \{2,\ldots, n\}$, ou seja a restrição de eliminação de subcircuitos (5) para um determinado subconjunto S é válida se e só se o mesmo se verifica para a restrição de corte (6) correspondente ao mesmo subconjunto S.

Note-se que na derivação anterior não se fez uso do facto de as variáveis x_{ij} terem que ser inteiras. Assim, a equivalência referida verifica-se também quando se consideram as relaxações lineares associadas aos dois modelos e portanto tem-se que $\nu(SUB_L) = \nu(CORTE_L)$.

Como se mencionou anteriormente, apenas se modelam as restrições (3) para subconjuntos $S \subseteq \{2,\ldots, n\}$, já que não é necessário considerar explicitamente tais restrições para subconjuntos que contenham o nodo 1. Apresentadas que estão duas representações (equivalentes) para as restrições (3), a questão que se coloca é de saber exactamente qual a representação para as restrições que eliminam subcircuitos em subconjuntos S tais que $\{1\} \subseteq S$ e que estão implícitas em SUB_L. Seja então S um subconjunto tal que $S \supseteq \{1\}$. Adicionando as restrições (2b) para $i \in S$ obtém-se $X(S) + X(S,S^C) = |S|$. Como $S^C \subseteq \{2,\ldots, n\}$ e SUB_L satisfaz as restrições de corte (6), tem-se que $X(S,S^C) \geq 1$, o que implica que $X(S) \leq |S| - 1$.

Por outras palavras, neste modelo as restrições de eliminação de subcircuitos para subconjuntos S tais que $S \supseteq \{1\}$ e que estão implícitas em SUB_L apresentam a mesma forma

que as restrições (5). É fácil de verificar que o mesmo se passa em relação às restrições de corte para os subconjuntos S que contêm o nodo 1. Tanto quanto se sabe da literatura, o primeiro trabalho que refere tal deve-se a Barros e Bárcia (1991).

O exposto permite afirmar que tanto o modelo SUB_L como o modelo $CORTE_L$ são independentes do nodo que se selecciona para o nodo 1. A razão de se focar este assunto aqui deve-se ao facto de que para outras formulações para o PCVA, as restrições de eliminação de subcircuitos para $S \subseteq \{2,\ldots, n\}$ serem diferentes, em termos da relaxação linear correspondente, das restrições de eliminação de subcircuitos associadas a subconjuntos que contêm o nodo 1. Nestes casos, a correspondente relaxação linear será dependente do nodo que for considerado como nodo 1. Tais formulações serão discutidas mais à frente.

A grande desvantagem associada à escolha de qualquer destes dois tipos de restrições, (5) ou (6), para eliminar subcircuitos deve-se ao número exagerado das mesmas. No entanto, existem actualmente métodos de geração implícita de restrições que permitem a resolução de SUB_L para instâncias com mais de 1000 nodos (ver, por exemplo, Padberg e Rinaldi (1991)).

3. Formulações estendidas baseadas em modelos de fluxos

Uma forma de obter formulações estendidas para o PCVA é defini-lo como um problema de fluxos em rede. Cada nodo do conjunto $\{2,\ldots, n\}$ recebe uma unidade de fluxo do nodo 1 (que pode ser considerado como um depósito).

3.1 Modelos de fluxo agregado

Um modelo de fluxo agregado (o termo "agregado" é utilizado para distinguir este modelo de outros apresentados mais à frente e que podem ser considerados como versões desagregadas deste) para o PCVA e aqui designado por *MFA*, foi apresentado por Gavish e Graves (1979). Trata-se de uma formulação estendida já que, para além das variáveis necessárias x_{ij}, é também usado um conjunto de variáveis contínuas não negativas y_{ij} $(i,j = 1,\ldots, n;\ i \neq j)$ que denotam a quantidade de fluxo enviada pelo nodo 1 e que percorre o arco (i,j). Como se mostra em Gavish e Graves (1979), é possível usar conjuntamente estes dois tipos de variáveis para escrever um conjunto de $O(n^2)$ restrições que também eliminam subcircuitos da relaxação de afectação. Tal é conseguido adicionando a (2a), (2b) e (4) as restrições:

$$\sum_{j=1}^{n} y_{1j} = n-1 \qquad (7a)$$

$$\sum_{i=1}^{n} y_{ij} - \sum_{i=1}^{n} y_{ji} = 1 \qquad j = 2,\ldots, n \qquad (7b)$$

$$y_{ij} \leq (n-1)x_{ij} \qquad i,j = 1,\ldots, n \qquad (8)$$

$$y_{ij} \geq 0 \qquad i,j = 1,\ldots, n \qquad (9)$$

Com o objectivo de novamente facilitar a indexação, as variáveis y_{ii} $(i = 1,\ldots, n)$ não são consideradas no sistema (7a) - (9). As restrições (7a) e (7b) são restrições de conservação de fluxo: (7a) indica que o nodo 1 envia n-1 unidades de fluxo e (7b) indicam que cada um dos restantes nodos recebe uma unidade de fluxo. As restrições (8) garantem que a existência de fluxo num arco (i,j) $(y_{ij} \geq 0)$ implica que tal arco esteja incluído no circuito óptimo $(x_{ij} = 1)$ e que o fluxo máximo em qualquer arco é igual a n-1.

É fácil de verificar que a quantidade n-y_{ij} indica a posição do arco (i,j) no circuito óptimo. Tal facto garante que qualquer solução admissível do modelo *MFA* não contém subcircuitos. De salientar que (7a), (7b), (8) e (9) é uma forma muito mais compacta de modelar as restrições de eliminação de subcircuitos (3). Isto indica uma vantagem de usar MFA_L em vez de SUB_L quando se pretende utilizar o respectivo valor óptimo para obter um limite inferior para o custo da solução óptima inteira correspondente. Contudo, como se verá a seguir, os valores obtidos por MFA_L são, em geral, muito mais fracos do que os obtidos por SUB_L.

Note-se que se n-1 unidades de fluxo são enviadas pelo depósito e cada uma dessas unidades é recebida por cada um dos restantes nodos, então o arco convergente no nodo 1 (depósito) deve ter fluxo igual a zero. De facto, pode mostrar-se que o mesmo se passa na relaxação linear deste modelo (isto é, $y_{i1} = 0$ para $i = 2,\ldots, n$). Subtraindo a (7a) a soma das n-1 restrições (7b) obtém-se

$$\sum_{i=2}^{n} y_{i1} = 0 \qquad (10)$$

Combinando (10) com (9) obtém-se $y_{i1} = 0$, para $i = 2,\ldots, n$. Note-se também que a restrição (8) para $i = 1$ é sempre verificada na igualdade na relaxação linear de *MFA*. Para verificar tal facto assuma-se que para um determinado par $(1,k)$ a restrição (8) é satisfeita como uma desigualdade estrita. Adicionando agora todas as desigualdades (8) para os pares $(1,j)$ $(j = 2,\ldots, n)$ obter-se-ia

$$\sum_{j=2}^{n} y_{1j} < (n\text{-}1) \sum_{j=2}^{n} x_{1j} .$$

Devido a (2b) para $i = 1$ esta desigualdade é equivalente a

$$\sum_{j=2}^{n} y_{1j} < n - 1$$

o que está em contradição com a restrição (7a).

As restrições (10) e o facto de as restrições (8) para $i = 1$ serem satisfeitas na igualdade sugerem que o modelo MFA_L é dependente do nodo 1. De facto, considere-se a seguinte solução admissível para MFA_L:

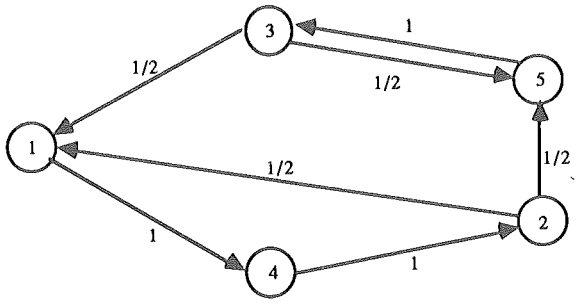


Figura 1 - Exemplo de uma solução admissível para MFA_L

Na figura 1 a etiqueta "α" associada a um arco (i,j) indica que $x_{ij} = \alpha$. Um valor de fluxo admissível para essa solução é dado por $y_{14} = 4$, $y_{42} = 3$, $y_{25} = 2$, $y_{53} = 1$ e $y_{31} = 0$ e restantes variáveis iguais a zero. Comece-se por renumerar os nodos do problema de modo a que na nova ordenação o nodo 1 corresponda ao nodo 5 da ordenação original. O valor do fluxo acima indicado não pode ser admissível neste caso porque não se têm 4 unidades de fluxo a serem enviadas pelo nodo 5 e, além disso, o valor y_{25} teria de ser igual a zero devido a (10). Isto indica que o conjunto das soluções admissíveis de MFA_L depende do nodo seleccionado para nodo 1. No entanto, como apenas as variáveis x_{ij} estão envolvidas na função objectivo, é possível que exista um outro fluxo (com origem no nodo 5) que seja admissível conjuntamente com os valores indicados para as variáveis x_{ij}.

A única maneira admissível do nodo 5 conseguir enviar 4 unidades de fluxo é fazendo $y_{53} = 4$. Mas a restrição de conservação de fluxo no nodo 3 indica que três unidades de fluxo teriam de ser enviadas pelo nodo 3. Como y_{35} tem que ser nulo (note-se que o nodo 5 é agora considerado como depósito), essa quantidade de fluxo teria que ser enviada através do arco (3,1). Comparando o valor de x_{31} (=0.5) com o de y_{31} (=3) verifica-se que a restrição (8) é violada para $(i,j) = (3,1)$. Tem-se assim que, para os valores indicados das variáveis x_{ij}, não é possível encontrar um fluxo admissível com origem no nodo 5 e, portanto, o custo da solução óptima de MFA_L é dependente do nodo escolhido para nodo 1.

3.2 Modelos de fluxo desagregado

Uma forma bem conhecida (possivelmente sugerida em primeiro lugar por Rardin e Choe (1979) para um problema mais geral) de melhorar a relaxação linear do modelo de fluxo agregado consiste em desagregar, por origem ou destino, a informação associada às variáveis de fluxo. Para definir este modelo usa-se um conjunto de variáveis de fluxo desagregado f_{ij}^k $(i,j = 1,\ldots, n;\ k = 2,\ldots, n;\ i \neq j)$ que indicam a quantidade de fluxo enviada pelo nodo 1, percorrendo o arco (i,j) e com destino ao nodo k. Existem várias formas de derivar tais modelos para o PCVA. Neste trabalho apresenta-se o modelo proposto por Claus (1984) no qual o conjunto (3) é modelado por um conjunto de restrições de fluxo mais um conjunto adequado de

$O(n^3)$ restrições que relacionam as variáveis de fluxo com as variáveis x_{ij}. Este sistema é mostrado a seguir de uma forma ligeiramente diferente:

$$\sum_{j=2}^{n} f^k_{1j} = 1 \qquad k = 2,\ldots, n \qquad (11a)$$

$$\sum_{i=1}^{n} f^k_{ij} - \sum_{i=1}^{n} f^k_{ji} = 0 \qquad j,k = 2,\ldots, n; j \neq k \qquad (11b)$$

$$\sum_{i=1}^{n} f^k_{ik} = 1 \qquad k = 2,\ldots, n \qquad (11c)$$

$$f^k_{ij} \leq x_{ij} \qquad i,j = 1,\ldots, n; k = 2,\ldots, n \qquad (12)$$

$$f^k_{ij} \geq 0 \qquad i,j = 1,\ldots, n; k = 2,\ldots, n \qquad (13)$$

De um modo análogo ao usado para o modelo *MFA* é possível provar que as variáveis f^j_{ji} ($j = 2,\ldots, n$; $i = 1,\ldots, n$) e f^k_{i1} ($i,k = 2,\ldots, n$) têm valor zero na relaxação linear deste modelo. O primeiro caso indica que um arco divergente de um nodo j ($j = 2,\ldots, n$) não é utilizado para enviar fluxo cujo destino é precisamente o mesmo nodo. O segundo caso indica que o nodo 1 não recebe qualquer quantidade de fluxo. Para cada $k = 2,\ldots, n$, o sistema (11a), (11b) e (11c) garante a existência de um caminho do nodo 1 para o nodo k. As restrições de ligação (12) garantem simplesmente que se o arco (i,j) está no caminho para um nodo k ($f^k_{ij} = 1$) então está necessariamente no circuito óptimo ($x_{ij} = 1$). Assim, uma solução admissível para este modelo é necessariamente conexa. Denote-se por *MFD* (modelo de fluxo desagregado) o modelo definido por (1), (2a), (2b), (11a), (11b), (11c), (4), (12), e (13). Note-se que as variáveis de fluxo agregado e as variáveis de fluxo desagregado estão relaciondas do seguinte modo:

$$\sum_{k=2}^{n} f^k_{ij} = y_{ij} \qquad i,j = 1,\ldots, n \qquad (14)$$

Se para um par fixo (i,j) adicionarmos as restrições (12) para $k = 2,\ldots, n$ e utilizarmos (14) obtém-se (8). Analogamente se obtêm as restrições de conservação de fluxo agregado (7a) e (7b) a partir das restrições de fluxo desagregado (11a), (11b) e (11c). Isto mostra que $v(MFD_L) \geq v(MFA_L)$. Em geral, os valores da relaxação linear de *MFD* são bastantes melhores que os da relaxação linear de *MFA*. O preço de tal melhoramente é um considerável aumento do número de variáveis e restrições.

Tanto quanto se sabe da literatura, o primeiro trabalho sobre modelos de fluxo desagregado para o PCVA foi proposto por Wong (1980). Este modelo inclui o dobro de variáveis de fluxo incluídas no modelo de Claus. Em Langevin, Soumis e Desrosiers (1990) são apresentadas duas variantes do modelo de Wong que utilizam o mesmo número de variáveis, mas um número menor de restrições. Tendo em conta que o valor da solução óptima da relaxação linear

produzida pelos quatro modelos é o mesmo, como se mostra em Langevin, Soumis e Desrosiers (1990), optou-se pelo modelo de Claus por ser mais compacto que qualquer um dos outros.

Convém salientar que para obter um modelo mais forte não basta criar o novo conjunto de variáveis f_{ij}^k. É necessário saber usá-las na derivação de restrições que podem levar à obtenção de modelos com relaxação linear mais forte. De facto, se no modelo de fluxo desagregado se utilizassem as restrições

$$\sum_{k=2}^{n} f_{ij}^k \leq (n-1)x_{ij} \qquad i,j = 1,\ldots, n \qquad (8')$$

$$\sum_{k=2}^{n} f_{ij}^k \geq 0 \qquad i = 1,\ldots, n; j = 2,\ldots, n \qquad (9')$$

em vez das restrições (12), obter-se-ia um modelo cuja relaxação linear seria equivalente à relaxação linear de *MFA*. Tal modelo seria apenas uma desagregação do modelo *MFA*. Por outras palavras, a introdução de novas variáveis conduz apenas à criação de um modelo menos compacto. É precisamente devido à introdução das restrições (12) que é possível derivar um modelo mais forte, em termos de relaxação linear associada. Note-se que as restrições (12) indicam que o valor de x_{ij} tem que ser não inferior ao máximo dos valores dos vários fluxos que passam pelo arco (i,j). Por outro lado, as restrições (8') indicam que o valor de x_{ij} tem que ser não inferior à média dos valores dos vários fluxos que passam pelo arco (i,j).

Para exemplificar esta diferença considere-se novamente a solução admissível para MFA_L, ilustrada na figura 1, reescrita agora com as variáveis de fluxo desagregado f_{ij}^k de modo a satisfazer as restrições (11a), (11b), (11c), (8'), (9') e (14):

$f_{14}^2 = f_{42}^2 = 1$; $f_{14}^3 = f_{42}^3 = f_{25}^3 = f_{53}^3 = 1$; $f_{14}^4 = 1$; $f_{14}^5 = f_{42}^5 = f_{25}^5 = 1$;

restantes variáveis de fluxo iguais a zero;

o valor das variáveis x_{ij} encontra-se indicado na figura 1.

Examinando o arco (2,5) é fácil de verificar que, como indica a restrição (8') para (2,5), o valor de x_{25} (=1/2) é maior ou igual que a média dos valor de $f_{25}^3, f_{25}^4, f_{25}^5$. Isto é, $f_{25}^3+f_{25}^4+f_{25}^5 \leq 4x_{25}$ $(1+0+1 \leq 4 \times 1/2)$. No entanto, as restrições (12)

$f_{25}^3 \leq x_{25}$ e $f_{25}^5 \leq x_{25}$

são violadas por essa solução. No modelo MFD_L, x_{25} tem que ser maior ou igual ao máximo de $\{f_{25}^3, f_{25}^4, f_{25}^5\}$ que é igual a 1.

3.3 Comparação dos modelos de fluxo no subespaço definido pelas variáveis x_{ij}

Será interessante agora comparar os modelos de fluxo referidos com o modelo SUB_L. Considerando o valor da variável x_{ij} como sendo a capacidade do arco (i,j), o teorema do fluxo máximo-corte mínimo (ver, por exemplo, Ahuja, Magnanti e Orlin, (1993)) permite afirmar que para $k = 2,\ldots, n$ o sistema (11a), (11b), (11c) e (12) é admissível se e só se a restrição de

corte (6) é válida para um subconjunto S tal que $\{k\} \subseteq S$ e $1 \notin S$. Considerando que o sistema é admissível para todos os $k = 2,\ldots, n$ sai imediatamente que a projecção, no subespaço definido pelas variáveis x_{ij}, do conjunto das soluções admissíveis definido pela relaxação linear de *MFD* é completamente descrita por (2a), (2b), (4') e (6). Isto implica que o modelo MFD_L é uma versão estendida do modelo $CORTE_L$ (e também do modelo SUB_L). Esta equivalência indica também que, ao contrário do modelo MFA_L, o modelo MFD_L é independente do nodo escolhido para nodo 1. Tanto quanto se sabe da literatura, Wong (1980) foi o primeiro a utilizar o teorema do fluxo máximo-corte mínimo para estabelecer uma equivalência formal entre a relaxação linear da formulação *CORTE* (ou *SUB*) e da formulação de fluxos desagregada, *MFD*.

O resultado anterior implica que os limites dados pelo modelo natural $CORTE_L$ (ou SUB_L) e pelo modelo estendido MFD_L são sempre iguais. Em primeira análise, tal resultado indica que o modelo MFD_L será preferido ao modelo $CORTE_L$ (ou SUB_L). Contudo, é sabido que para instâncias com pelo menos 20 nodos é necessário utilizar enormes quantidades de tempo de CPU para resolver o modelo MFD_L. Como já foi referido, a desvantagem associada ao modelo $CORTE_L$ (ou SUB_L) é o número exponencial de restrições que eliminam subcircuitos. Contudo, apenas algumas dessas restrições são satisfeitas na igualdade pela solução óptima da relaxação linear. Este facto sugere que um procedimento de geração de restrições pode constituir uma alternativa razoável para a resolução do problema linear associado a qualquer um dos dois modelos *CORTE* (ou *SUB*). Assim, comece-se por resolver a relaxação linear de um submodelo do modelo *CORTE* (ou *SUB*) que envolve apenas um pequeno subconjunto de restrições que eliminam subcircuitos. De seguida, um procedimento de geração implícita de restrições é usado para identificar as desigualdades que são violadas pela solução corrente. Estas desigualdades são adicionadas ao modelo linear corrente e um novo modelo é resolvido.

Isto parece sugerir que o resultado da equivalência entre MFD_L e $CORTE_L$ (ou SUB_L) apenas se revela interessante do ponto de vista teórico. Contudo, tal não é verdade. Uma importante consequência, quer do ponto de vista teórico, quer do ponto de vista prático, é o facto de o resultado mostrar que o problema de separação associado à identificação das restrições (6) ou (5) pode ser eficientemente resolvido. Recorde-se que o problema de separação associado a um conjunto de restrições R consiste em determinar se uma solução linear sarisfaz todas as restrições R ou, no caso contrário, identificar algumas restrições em R violadas por essa solução.

Para encontrar a relação entre a equivalência dos dois modelos com o problema de separação associado, considere-se uma solução admissível para a relaxação linear de uma formulação natural válida para o PCVA. Pretende-se verificar se essa solução satisfaz todas as restrições de corte (6). Pense-se no valor dessas variáveis como capacidades máximas dos arcos existentes num grafo $G = (V,A)$ em que V é o conjunto dos nodos do problema inicial e $(i,j) \in A$ se e só se $x_{ij} \geq 0$. Tende-se agora enviar uma unidade de fluxo do nodo 1 para todos os

restantes nodos. Se tal for possível, então a capacidade de qualquer corte $[S^C,S]$ com $1 \in S^C$ é maior ou igual a 1, ou seja $X(S^C,S) \geq 1$ e todas as restrições (6) são satisfeitas por essa solução. Caso contrário, a quantidade máxima de fluxo que é possível enviar com destino a um nodo k é igual a ν ($\nu < 1$). Então, pelo teorema do fluxo máximo, existe um corte $[S^C,S]$ tal que $1 \in S^C$ e $k \in S$ e cuja capacidade é igual a ν e portanto inferior a 1. Assim, tem-se $X(S^C,S) = \nu$ e a correspondente restrição de corte (6) é violada para esse corte. Tal restrição pode ser adicionada ao modelo linear num processo de geração implícita.

Deste modo, o problema de separação associado às restrições (6) pode ser resolvido através de n-1 problemas de fluxo máximo. Convém salientar que actualmente existem algoritmos mais eficientes e que permitem identificar a restrição de corte (6) violada (caso exista) que maximiza a diferença entre o termo independente (de valor 1) e o valor do membro esquerdo na solução linear corrente (ver Padberg e Grotschel (1985)).

Como exemplo do procedimento acima indicado, utilize-se a solução linear ilustrada na figura 1. O valor do fluxo máximo de 1 para 5 é igual a 1/2. Muitos algoritmos que permitem indicar tal fluxo indicam também o corte de capacidade mínima $[S^C,S]$ (ver, por exemplo, o algoritmo de Ford Fulkerson (1962)), neste caso constituído por $S^C = \{1,2,4\}$ e $S = \{3,5\}$. Isto indica que para esta solução se tem

$$X(S^C,S) = 1/2 < 1$$

Logo a restrição (6) associada a esse corte pode ser adicionada ao modelo que produziu a solução original.

Para uma melhor compreensão da diferença entre os modelos MFA_L e MFD_L apresenta-se de seguida um resultado que dá uma caracterização, no subespaço definido pelas variáveis x_{ij}, do conjunto das soluções admissíveis de MFA_L. Este resultado indica que a projecção no subespaço definido pelas variáveis x_{ij}, do conjunto das soluções admissíveis definido por MFA_L, é completamente descrito por (2a), (2b), (4') e o seguinte conjunto de restrições

$$X(S^C,S) \geq \frac{|S|}{n\text{-}1} \qquad \forall S \subseteq \{2,\ldots, n\} \qquad (15)$$

que é uma versão mais fraca das restrições de corte (6).

Pata simplificar a notação utilizada na demonstração que se segue, seja $Y(A,B) = \sum_{i \in A} \sum_{j \in B} y_{ij}$. Considere-se então uma solução admissível $\{x_{ij},y_{ij}\}$ para MFA_L e um conjunto $S \subseteq \{2,\ldots, n\}$. Adicionando as restrições (7b) para $j \in S$ e removendo os termos iguais obtém-se $Y(S^C,S) - Y(S,S^C) = |S|$ que é equivalente a

$$Y(S^C,S) = |S| + Y(S,S^C) \qquad (16)$$

Atendendo a (8) tem-se

$$(n\text{-}1)\, X(S^C,S) \geq |S| \qquad (17)$$

Finalmente, dividindo cada membro de (17) por n-1 obtém-se (15). Deste modo tem-se que a solução $\{x_{ij},y_{ij}\}$, admissível para MFA_L, satisfaz as restrições (15).

Reciprocamente, considere-se uma solução $\{x_{ij}\}$ admissível para (2a), (2b) e (15). Para mostrar que existe um vector de fluxos $\{y_{ij}\}$ de tal modo que $\{x_{ij},y_{ij}\}$ é admissível para MFA_L basta notar que a existência de (7b), (8), (9) e (10) (note-se que se mostrou que (10) e (7a) são equivalentes na presença de (7b)) induz uma rede de fluxos com capacidades, em que as restrições (8) e (9) podem ser vistas como capacidades máxima e mínima nos arcos de fluxos. Uma consequência directa do teorema do fluxo máximo indica que esse sistema tem um fluxo admissível se e só se

$$\sum_{i\in S^C}\sum_{j\in S} M_{ij} - \sum_{j\in S}\sum_{i\in S^C} m_{ij} \geq |S| \qquad \forall S\subseteq\{2,\ldots,n\} \qquad (18)$$

em que M_{ij} e m_{ij} são capacidades máxima e mínima do arco (i,j). Fazendo $M_{ij} = (n-1)x_{ij}$ e $m_{ij} = 0$, vem que (18) é equivalente a

$$(n-1)\,X(S^C,S) \geq |S| \qquad (19)$$

Como $\{x_{ij}\}$ satisfaz (15) sai que (19) é válida e, portanto, existe um fluxo admissível satisfazendo (7b), (8), (9) e (10).

Note-se que $\lceil |S|/n-1 = 1 \rceil$, $\forall S\subseteq\{2,\ldots,n\}$. Comparando o segundo membro das restrições de corte (6) com o segundo membro da versão mais fraca destas (15) verifica-se que reformular o modelo de fluxo agregado como um modelo de fluxo desagregado corresponde precisamente a apertar o termo independente de (15). Já se tinha visto antes que $\nu(MFD_L) \geq \nu(MFA_L)$. A caracterização destes modelos no subespaço definido pelas variáveis x_{ij} seria uma maneira indirecta de provar tal resultado.

Do mesmo modo que se mostrou que (5) e (6) eram equivalentes, também se pode mostrar que, na presença de (2a), as restrições de corte mais fracas (15) são equivalentes às restrições de subcircuitos mais fracas

$$X(S) \leq |S| - \frac{|S|}{n-1} \qquad \forall S\subseteq\{2,\ldots,n\} \qquad (20)$$

Note-se também que (15) ou (20) dão origem a outras formulações naturais para o PCVA. Do ponto de vista das respectivas relaxações lineares, tais formulações poder-se-iam considerar desinteressantes por duas razões: em primeiro lugar, porque o mesmo efeito poderia ser obtido com a utilização de um modelo de fluxos mais compacto; em segundo lugar, e caso preferíssemos utilizar a versão natural do modelo, seria muito mais óbvio utilizar a versão forte das mesmas restrições já que o número de restrições não se modificaria. O interesse em derivar (15) ou (20) é de que permite avaliar a diferença entre MFD_L e MFA_L no subespaço definido apenas pelas variáveis x_{ij}.

Como se mencionou antes, a formulação MFA_L depende do nodo que for seleccionado para nodo 1. Esta dependência torna-se ainda mais evidente no subespaço das variáveis x_{ij}. Com um raciocínio análogo ao utilizado no fim da secção 2 é possível mostrar que (20) se verifica se e só se

$$X(S) \leq |S| - \frac{|S^C|}{n-1} \qquad \forall S\subseteq\{2,\ldots,n\};\ \{1\}\subseteq S;\ |S| < n \qquad (21)$$

Isto é, o termo independente da versão fraca das restrições de subcircuitos verificadas por MFA_L depende do facto de S conter ou não o nodo 1.

Considere-se a solução ilustrada na figura 1 e o conjunto $S = \{3,5\}$. Esta solução satisfaz as restrições (20) para esse subconjunto de nodos. No entanto, se o nodo 5 fosse o nodo considerado como depósito ter-se-ia que utilizar a desigualdade (21), obtendo-se

$$X(\{3,5\}) \leq |\{3,5\}| - \frac{|\{1,2,4\}|}{4} = 2 - \frac{3}{4} = 1.25$$

e tal restrição é violada pela solução ilustrada, já que $X(\{3,5\}) = x_{35} + x_{53} = 1.5$.

4. Formulações estendidas com dependências de tempo

Como já se referiu, o modelo *MFA* pode ser considerado desinteressante no sentido em que nada traz de novo para a descrição do politopo associado ao PCVA. No entanto, e como se viu na secção anterior, com uma desagregação adequada das variáveis de fluxo é possível obter um modelo mais forte que o modelo de fluxo agregado. Nesta secção mostra-se outra maneira de fortalecer o modelo *MFA*.

Comece-se por notar no modelo *MFA* que apenas no arco do circuito óptimo que diverge da raíz o valor do fluxo é igual a n-1. Relativamente aos outros arcos esse valor é, no máximo, igual a n-2, porque uma unidade de fluxo é necessariamente absorvida pelo nodo imediatamente a seguir ao depósito. De um modo análogo se verifica que, com excepção do arco convergente no depósito, todos os restantes arcos do circuito óptimo têm valor de fluxo não inferior a 1. Em consequência tem-se que as restrições (8) para $i > 1$ e (9) para $j > 1$ podem ser apertadas. Deste modo obtém-se o seguinte modelo, aqui designado por MFA^+:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \tag{1}$$

$$\text{s.a.} \sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,\ldots, n \tag{2a}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,\ldots, n \tag{2b}$$

$$\sum_{j=1}^{n} y_{1j} = n\text{-}1 \qquad j = 2,\ldots, n \tag{7a}$$

$$\sum_{i=1}^{n} y_{ij} - \sum_{i=2}^{n} y_{ji} = 1 \qquad j = 2,\ldots, n \tag{7b}$$

$$\begin{aligned} y_{1j} &= (n\text{-}1)x_{ij} && j = 2,\ldots, n \\ y_{ij} &\leq (n\text{-}2)x_{ij} && i,j = 2,\ldots, n \end{aligned} \tag{8"}$$

$$y_{i1} \geq 0 \qquad i = 2,\ldots, n$$

$$y_{ij} \geq x_{ij} \qquad i = 1,\ldots, n\ ; j = 2;\ldots, n \qquad (9'')$$

$$x_{ij} \in \{0,1\} \qquad i,j = 1,\ldots, n \qquad (4)$$

A relaxação linear deste modelo é mais forte que a do modelo *MFA*. Como exemplo, considere-se novamente a solução ilustrada na figura 1. Note-se que para essa solução se tem $y_{25} = 4x_{25}$, o que implica que a restrição (8"), para o par (2,5), é violada. Por outro lado, sendo $y_{35} = 0$ e como $x_{35} > 0$ tem-se que a restrição (9") também é violada para o par (3,5). Como se referiu antes, isso não implica a impossibilidade de existência de outro fluxo com origem no nodo 1 e que satisfaça as restrições mais apertadas (8") e (9"). No entanto, é fácil ver que é necessário ter $y_{14} = 4$ e $y_{42} = 3$, pelo que 2 unidades de fluxo têm que ser enviadas pelo nodo 2. Como por (10) se tem que ter $y_{21} = 0$ sai imediatamente que $y_{25} = 2$ e a restrição (8"), para o par (2,5), é violada. Como se pode observar mais à frente, o valor óptimo da correspondente relaxação linear do modelo MFA^+ ainda se encontra, em geral, afastado do obtido por SUB_L. Obviamente, o valor $\nu(MFA_L^+)$ também é dependente do nodo que se considerar para o nodo 1. Aliás, esse facto é ainda mais evidente no caso do modelo MFA_L^+ porque, como as restrições (8") e (9") indicam, os limites, máximo e mínimo, do valor do fluxo nos arcos adjacentes ao nodo 1 são mais fracos do que os limites, máximo e mínimo, para os outros arcos do circuito óptimo.

Do mesmo modo que se mostrou que $\nu(MFD_L) \geq \nu(MFA_L)$, também se pode mostrar que $\nu(MFD_L) \geq \nu(MFA_L^+)$. Basta apenas mostrar que (8") e (9") podem ser derivadas a partir das desigualdades do modelo MFD_L e (14). A partir das restrições (12) e (14) e utilizando o facto de que $f_{ij}^i = 0$ para todo o par (i,j) obtém-se (8") para $i > 1$. Relativamente a (8") para $i = 1$, note-se que as restrições (2b) para $i = 1$ combinadas com as restrições (11a) para qualquer k garantem que as restrições (12) para $i = 1$ e $j,k = 2,\ldots, n$ são satisfeitas na igualdade, isto é, $f_{1j}^k = x_{1j}$. Adicionando, para $k = 2,\ldots, n$, estas restrições para um par fixo $(1,j)$ obtém-se (8") para $i = 1$. Note-se também que os dois conjuntos de restrições (2a) para $j = 2,\ldots, n$ e (11c) implicam que as restrições (12) são sempre satisfeitas na igualdade quando $k = j$, isto é, $f_{ij}^j = x_{ij}$.. Isto mostra que MFD_L satisfaz também as restrições (9").

O principal objectivo em introduzir neste trabalho a formulação MFA^+ é de que esta pode ser vista como uma "ponte" para outra classe de formulações estendidas para o PCVA. Na realidade, existe uma forma alternativa de usar a informação associada às variáveis de fluxo em formulações estendidas para o PCVA. Em vez de se utilizar um conjunto adicional de variáveis, introduz-se um terceiro índice nas variáveis x_{ij}, que indica o fluxo do correspondente arco.

De facto, a informação associada às variáveis x_{ij} e y_{ij} do modelo pode MFA^+ ser duplicada através da introdução de variáveis z_{ij}^t $(i,j,t = 1,\ldots, n)$ tais que $z_{ij}^t = 1$ se o arco (i,j) tem fluxo t e $z_{ij}^t = 0$ no caso contrário. Tal duplicação de informação é conseguida através das seguintes relações:

$$x_{ij} = \sum_{t=1}^{n-2} z_{ij}^t \qquad i,j = 2,\ldots, n \qquad (23a) \qquad y_{ij} = \sum_{t=1}^{n-2} t\, z_{ij}^t \qquad i,j = 2,\ldots, n \qquad (23b)$$

$$x_{1j} = z_{1j}^{n-1} \qquad j = 2,\ldots, n \qquad (24a) \qquad y_{1j} = (n\text{-}1) z_{1j}^{n-1} \qquad j = 2,\ldots, n \qquad (24b)$$

$$x_{i1} = z_{i1}^0 \qquad i = 2,\ldots, n \qquad (25a) \qquad x_{i1} = 0.z_{i1}^0 \qquad i = 2,\ldots, n \qquad (25b)$$

Com o objectivo de simplificar a indexação nas formulações que a seguir se apresentam, considera-se o índice t a variar de 0 até n-1 em qualquer variável z_{ij}^t. As relações dadas anteriormente indicam então que

$$z_{1j}^t = 0 \qquad t < n\text{-}1 \qquad (26) \qquad z_{i1}^t = 0 \qquad t > 0 \qquad (27)$$

$$z_{ij}^0 = 0 \qquad j > 1 \qquad (28) \qquad z_{ij}^{n-1} = 0 \qquad i > 1 \qquad (29)$$

Substituindo as variáveis x_{ij} e y_{ij} no modelo MFA^+ pelos correspondentes segundos membros das igualdades (23a)-(25b) obtém-se a seguinte formulação para o PCVA, designada por *FGG*3 (a seguir será indicada uma explicação para esta notação).

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=0}^{n-1} c_{ij}\, z_{ij}^t \qquad (30)$$

$$\text{s.a.} \sum_{i=1}^{n} \sum_{t=0}^{n-1} z_{ij}^t = 1 \qquad j = 1,\ldots, n \qquad (31a)$$

$$\sum_{j=1}^{n} \sum_{t=0}^{n-1} z_{ij}^t = 1 \qquad i = 1,\ldots, n \qquad (31b)$$

$$\sum_{j=1}^{n} \sum_{t=1}^{n-1} t\, z_{ij}^t - \sum_{j=1}^{n} \sum_{t=0}^{n-2} t\, z_{ji}^t = 1 \qquad i = 2,\ldots, n \qquad (32)$$

(26), (27), (28) e (29)

$$y_{ij}^t \in \{0,1\} \qquad i,j = 1,\ldots, n\,;\, t = 0,\ldots, n-1 \qquad (33)$$

Por razões já anteriormente evocadas, as variáveis z_{ii}^t ($i,t = 1,\ldots, n$) não são consideradas. O principal motivo pelo qual se procedeu à substituição, na formulação MFA^+, das variáveis x_{ij} e y_{ij} pelas variáveis z_{ij}^t deve-se ao facto de as restrições (8") e (9") serem sempre verificadas quando reescritas com as novas variáveis, podendo assim serem omitidas na formulação *FGG*3 e consequentemente permitir uma redução considerável no número de restrições do modelo reformulado. De facto, as restrições (8") para $i \neq 1$ reescritas nas novas variáveis, dão origem às desigualdades.

$$\sum_{t=1}^{n-2} t\, z_{ij}^t \leq (n\text{-}2) \sum_{t=1}^{n-2} z_{ij}^t \qquad i,j = 2,\ldots, n$$

que são sempre verdadeiras visto que

$$\sum_{t=1}^{n-2} t\, z_{ij}^{t} \le \sum_{t=1}^{n-2} (n\text{-}2)\, z_{ij}^{t} = (n\text{-}2) \sum_{t=1}^{n-2} z_{ij}^{t} \qquad i,j = 2,\ldots, n$$

A restrição (8") para $i = 1$ é uma igualdade do tipo $A = A$.

Relativamente a (9") a restrição $0.z_{i1}^{0} \ge 0$ é obviamente verdadeira. Para provar a validade da restrição

$$\sum_{t=1}^{n-2} t\, z_{ij}^{t} \ge \sum_{t=1}^{n-2} z_{ij}^{t} \qquad i,j = 2,\ldots, n$$

que é a segunda restrição em (9") para $i \neq 1$ reescrita com as novas variáveis, basta ter em conta que $t \ge 1$. O caso $i = 1$ sai de igual modo.

Note-se que o modelo $FGG3$ contém um número menor de restrições ($O(n)$) que o modelo MFA^{+} ($O(n^2)$). No novo modelo, o número de variáveis é de $O(n^3)$ enquanto no modelo anterior esse número é de $O(n^2)$. É possível também provar que as relaxações lineares das duas formulações em causa produzem soluções com o mesmo custo, isto é $\nu(FGG3_L) = \nu(MFA_L^{+})$.

Considere-se uma solução $\{z_{ij}^{t}\}$ admissível para $FGG3$ e seja $\{x_{ij}, y_{ij}\}$ a solução obtida de $\{z_{ij}^{t}\}$ através das relações (23a), (23b), (24a), (24b), (25a) e (25b). É fácil verificar que a solução $\{x_{ij}, y_{ij}\}$ é admissível para MFA^{+} e ambas as soluções têm o mesmo custo. Consequentemente tem-se $\nu(FGG3_L) \ge \nu(MFA_L^{+})$.

Reciprocamente, dada uma solução admissível $\{x_{ij}, y_{ij}\}$ de MFA^{+}, existem várias maneiras de construir uma solução admissível para $FGG3$. Uma dessas maneiras é descrita a seguir:

a) caso $i = 1$

De acordo com (24a) toma-se, $z_{1j}^{n-1} = x_{1j}$ para $j = 2,\ldots,$ n

b) caso $j = 1$

De acordo com (25a) toma-se, $z_{i1}^{0} = x_{i1}$ para $i = 2,\ldots,$ n

b) caso $i,j \neq 1$

Para cada arco (i,j) $(i,j = 2,\ldots, n)$ tem-se que o valor y_{ij} / x_{ij} ou é inteiro ou não. No primeiro caso toma-se $z_{ij}^{c} = x_{ij}$ e $z_{ij}^{t} = 0$ para $t = 1,\ldots, n\text{-}2$ $(t \neq c)$ onde $c = y_{ij} / x_{ij}$. Se y_{ij} / x_{ij} não é inteiro toma-se $z_{ij}^{t} = 0$ para $t = 1,\ldots, n\text{-}2$ $(t \neq a,b)$ onde $a = \lfloor y_{ij} / x_{ij} \rfloor$ e $b = \lceil y_{ij} / x_{ij} \rceil$. Os valores z_{ij}^{a} e z_{ij}^{b} podem ser obtidos resolvendo o sistema

$$z_{ij}^{a} + z_{ij}^{b} = x_{ij} \quad \text{e} \quad a z_{ij}^{a} + z_{ij}^{b} = y_{ij}$$

Deste modo tem-se $z_{ij}^{a} = \dfrac{b x_{ij} - y_{ij}}{b - a}$ e $z_{ij}^{b} = \dfrac{y_{ij} - a_{ij}}{b - a}$.

É um simples exercício verificar agora que a solução $\{z_{ij}^{t}\}$ satisfaz (31a), (31b) e (32). Relativamente a $0 \le z_{ij}^{t} \le 1$, note-se que de acordo com a definição de a e b se tem $a x_{ij} \le y_{ij} \le b x_{ij}$, o que implica que $z_{ij}^{a}, z_{ij}^{b} \ge 0$. Por outro lado, tendo em conta que $z_{ij}^{a} + z_{ij}^{b} = x_{ij}$ e $x_{ij} \le 1$ tem-se que $z_{ij}^{a}, z_{ij}^{b} \le 1$.

A equivalência dos valores da função objectivo dados por ambas as soluções pode ser estabelecida de acordo com (23a), (24a) e (25a). Isto mostra que $\nu(FGG3_L) \le \nu(MFA_L^{+})$. Consequentemente obtém-se o resultado pretendido.

A equivalência entre o modelo MFA_L^+ e $FGG3_L$ e, de um modo geral, a reformulação de modelos de fluxo agregado em modelos com variáveis de três índices foi proposta em Gouveia (1995) para um outro problema de desenho de redes (veja-se também Gouveia e Voβ (1995)).

Note-se que o modelo *FGG3* não segue o esquema apresentado na página 3. No entanto, um modelo equivalente em termos de relaxação linear associada poderia ser obtido através do remoção de (30), (31a) e (31b) de *FGG3* e inclusão de (1), (2a), (2b), (23a), (24a) e (25a). O novo modelo estaria de acordo com o tal esquema, mas incluiria $O(n^2)$ restrições e seria menos compacto que o modelo *FGG3* apresentado. Isto significa que, ao contrário dos modelos de fluxos discutidos na secção 3, a informação associada às variáveis supérfluas pode ser transformada na informação associada às variáveis naturais através das igualdades (23a), (24a) e (25a), o que permite a remoção das variáveis naturais do modelo *FGG3*.

Até agora, esta técnica de reformulação apenas permitiu a obtenção de um novo modelo cuja relaxação linear é equivalente à relaxação linear de um modelo conhecido de fluxos. No entanto, as novas variáveis sugerem novas restrições para o PCVA cuja relaxação linear é bastante mais forte do que a relaxação linear do modelo original. De facto, a relaxação linear do modelo *FGG3* pode ser fortalecida através da introdução das restrições:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij}^{t} = 1 \qquad t = 0,\ldots, n-1 \qquad (34)$$

que indicam que a cada unidade de fluxo está associado um e um só arco. Como exemplo, considere-se a solução ilustrada na figura 2, admissível para $FGG3_L$.

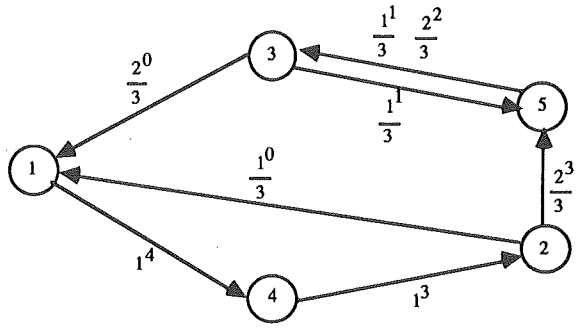


Figura 2 - Exemplo de uma solução admissível para $FGG3_L$

Na figura 2, uma etiqueta da forma a^b associada a um arco (i,j) indica que $z_{ij}^b = a$. Note-se também que, em alguns casos, pode existir mais do que uma etiqueta associada a um arco. Veja-se, por exemplo, o caso do arco (5,3). As etiquetas associadas a esse arco indicam que $z_{53}^1 = 1/3$, $z_{53}^2 = 1/3$ e $z_{53}^3 = 0$. Recorde-se que $z_{53} = z_{53} = 0$ é garantido por (28) e (29). A solução ilustrada não satisfaz as restrições (34) para $t = 2$ e $t = 3$. No que se segue designa-se por *FGG4* o modelo *FGG3* aumentado com as restrições (34). Obviamente tem-se $\nu(FGG4_L) \geq$

$v(FGG3_L)$. O exemplo anterior mostra que existem instâncias do problema para as quais $v(FGG4_L) > v(FGG3_L)$.

As igualdades (23a)-(25a) e/ou (23b)-(25b) indicam que, de um ponto de vista teórico, deve ser possível adicionar um conjunto de restrições ao modleo MFA^+ de modo a que o valor da relaxação linear do novo modelo seja igual a $v(FGG4_L)$. No entanto, não parece fácil obter tal conjunto de restrições e conjectura-se que tal conjunto contém um número exponencial (em n) de restrições. Esta conjectura, caso verdadeira, indica outra vantagem de se associar a um terceiro índice a informação correspondente ao fluxo de um arco.

Os modelos $FGG3$ e $FGG4$ foram propostas, de uma forma ligeiramente diferente, por Fox, Gavish e Graves (1980) para o problema do caixeiro viajante com dependências de tempo (PCVDT). Neste problema o índice "t " das variáveis envolvidas, u_{ij}^t, indica a posição do arco no circuito óptimo. Assim, $u_{ij}^t = 1$ se o aco (i,j) ocupa a posição t no circuito óptimo e $u_{ij}^t = 0$ no caso contrário. Estas variáveis relacionam-se com as variáveis z_{ij}^t, apresentadas antes, do seguinte modo:

$$u_{ij}^t = z^{n-t}{}_{ij} \qquad t = 1,\ldots, n$$

Neste problema o custo associado a um arco depende da posição em que é inserido no circuito óptimo e, portanto, é natural utilizar tais variáveis. Note-se que $FGG3$ e $FGG4$ contêm, respectivamente, $3n$ e $4n$ restrições. Este facto, juntamente com as iniciais dos autores, motivam a designação escolhida para estes dois modelos.

É possível mostrar que não existe qualquer relação entre os valores $v(FGG4_L)$ e $v(CORTE_L)$ (ou $v(SUB_L)$, ou $v(MFD_L)$) devido às equivalências anteriormente inidcadas. Considere-se a solução ilustrada na figura 3 que é admissível para $FGG4_L$. Usando as relações (23a), (24a) e (25a) verifica-se que $x_{12} = x_{53} = 1/3$ e, portanto, tal solução não satisfaz as restrições de corte (6) para o conjunto $S = \{2,3\}$.

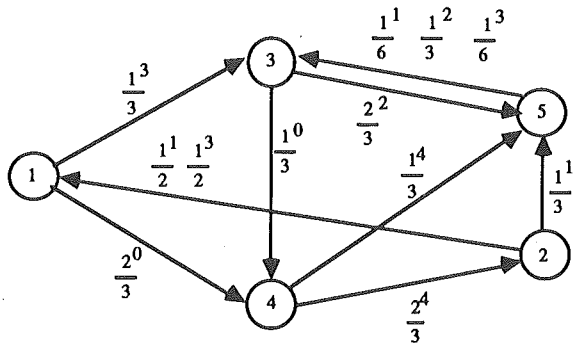


Figura 3 - Exemplo de uma solução admissível para $FGG4_L$ mas não admissível para $CORTE_L$

Considere-se agora a solução admissível para $CORTE_L$, apresentada na figura 4.

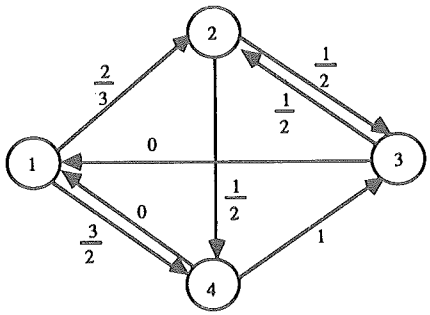


Figura 4 - Exemplo de uma solução admissível para $CORTE_L$ mas não admissível para $FGG4_L$

O valor das variáveis x_{ij} é dado pela seguinte regra: $x_{ij} = 1/2$ se o arco (i,j) está indicado na figura; $x_{ij} = 0$ no caso contrário. Devido à "equivalência" entre os modelos $CORTE_L$ e MFD_L, referida anteriormente, sabe-se que a solução acima indicada é também admissível para MFD_L. O correspondente valor das variáveis f_{ij}^k é o seguinte:

$f_{12}^2 = f_{14}^2 = f_{32}^2 = f_{43}^2 = 1/2$; $f_{12}^3 = f_{14}^3 = f_{23}^3 = f_{43}^3 = 1/2$; $f_{12}^4 = f_{14}^4 = f_{24}^3 = 1/2$.

Usando a igualdade $y_{ij} = \sum_{k=2}^{n} f_{ij}^k$, que relaciona as variáveis de fluxo agregado com as variáveis de fluxo desagregado, obtém-se uma solução equivalente reescrita nas variáveis $\{x_{ij}, y_{ij}\}$. O valor das variáveis assim obtido está também indicado na figura junto aos arcos correspondentes. Utilizando agora a transformação indica na prova do resultado $\nu(FGG3_L) = \nu(MFA_L^+)$, a solução $\{x_{ij}, y_{ij}\}$ pode ser transformada numa solução reescrita nas variáveis z_{ij}^t. Tal solução está ilustrada na figura 5:

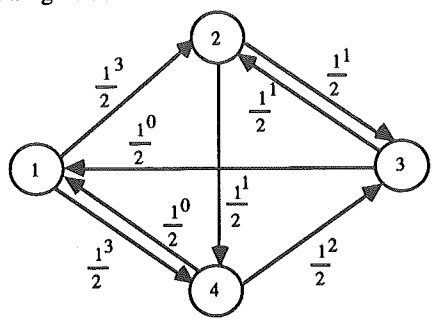


Figura 5 - Solução da figura 4 reescrita nas variáveis z_{ij}^t

Note-se que esta solução não é admissível para $FGG4_L$ porque não satisfaz as restrições (34) para $t = 1$ e $t = 2$. Um leitor mais atento poderá argumentar que pode existir outro conjunto de valores para as variáveis $\{z_{ij}^t\}$ que satisfaça todas as restrições (34) e que conduza à mesma solução $\{x_{ij}, y_{ij}\}$. Por exemplo, considere-se o arco (2,3). Devido a (28) e (29) tem-se

que $z_{23}^0 = z_{23}^3 = 0$. Assim, pelas restrições (23a) e (23b), o valor das variáveis $\{z_{ij}^t\}$ associado ao arco (2,3) é dado pelo sistema

$$z_{23}^1 + z_{23}^2 = x_{23} = 1/2 \quad \text{e} \quad 1.z_{23}^1 + 2.z_{23}^1 = y_{23} = 1/2$$

que tem a solução única $z_{23}^1 = 1/2$ e $z_{23}^2 = 0$. Um raciocínio análogo permite deduzir que para os valores $\{x_{ij}, y_{ij}\}$ da solução indicada na figura 4, o valor das variáveis $\{z_{ij}^t\}$ indicado na figura 5 é único. Conlui-se assim que não existe qualquer relação de dominância entre $FGG4_L$ e $CORTE_L$ (ou SUB_L).

É interessante notar que, utilizando as variáveis z_{ij}^t, existe uma maneira muito mais intuitiva de escrever restrições que eliminam subcircuitos. Tais restrições estão indicadas a seguir:

$$\sum_{i=1}^{n} z_{ij}^t = \sum_{i=1}^{n} z_{ji}^{t-1} \qquad j = 2,\ldots, n\,;\ t = 1,\ldots, n\text{-}1 \qquad (35)$$

Estas restrições obrigam a que se a um dado nodo, com excepção do nodo considerado como depósito, chega um arco com fluxo t então desse nodo sai um arco com fluxo t-1.

O exemplo apresentado na figura 3 que, como já se referiu, ilustra uma solução admissível para $FGG4_L$, permite verificar que as restrições (35) não são redundantes quando adicionadas ao modelo $FGG4_L$. De facto, constata-se facilmente que tais restrições são violadas por esta solução para qualquer nodo, excepto o nodo 1.

Uma formulação que envolve um conjunto de restrições semelhante a (35) foi também apresentada por Picard e Queyranne (1978) para o PCVDT. Assim, no que se segue, denota-se por PQ a formulação:

$$\text{min} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{t=0}^{n-1} c_{ij}\, z_{ij}^t \qquad (30)$$

$$\text{s.a.} \quad \sum_{i=1}^{n}\sum_{t=0}^{n-1} z_{ij}^t = 1 \qquad j = 1,\ldots, n \qquad (31a)$$

$$\sum_{j=1}^{n}\sum_{t=0}^{n-1} z_{ij}^t = 1 \qquad i = 1,\ldots, n \qquad (31b)$$

$$\sum_{i=1}^{n} z_{ij}^t = \sum_{i=1}^{n} z_{ji}^{t-1} \qquad j = 2,\ldots, n\,;\ t = 1,\ldots, n-1 \qquad (35)$$

(26), (27), (28) e (29)

$$y_{ij}^t \in \{0,1\} \qquad i,j = 1,\ldots, n\,;\ t = 0,\ldots, n-1 \qquad (33)$$

Em Gouveia e Voβ (1995) mostra-se que na presença das restrições (35), as restrições (32) e (34) são redundantes. Uma consequência deste facto é de que $v(PQ_L) \geq v(FGG4_L)$. O exemplo da figura 3 serve para mostrar que, em alguns casos, se tem $v(PQ_L) > v(FGG4_L)$. Como PQ_L domina $FGG4_L$ conclui-se que o novo modelo PQ_L não é dominado pelo modelo

$CORTE_L$. O recíproco também é verdadeiro, já que, utilizando novamente as relações (23a) e (24a) e (25a) facilmente se verifica que a seguinte solução admissível para PQ_L não verifica as restrições de corte (6) para $S = \{2,3\}$ e $S = \{4,5\}$ e, portanto, não é admissível para $CORTE_L$.

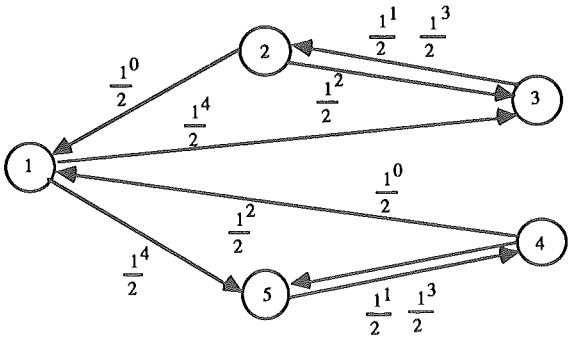


Figura 6 - Exemplo de uma solução admissível para PQ_L mas não admissível para $CORTE_L$

Estes resultados de não dominância entre os modelos PQ_L (ou $FGG4_L$) e $CORTE_L$ sugerem que seria de todo o interesse encontrar modelos equivalentes a PQ_L e $FGG4_L$ (em termos de relaxação linear) escritos apenas com as variáveis naturais x_{ij}. Até ao momento não se conhecem tais resultados. Os resultados expostos nesta secção ilustram a vantagem de se obter modelos equivalentes reescritos com conjuntos de variáveis diferentes. Até muito recentemente não se conhecia qualquer relação entre o modelo PQ_L e os modelos de fluxo agregado MFA_L e MFA_L^+. A equivalência entre o modelo MFA_L^+ e o modelo $FGG3_L$, conjuntamente com a dominância do modelo PQ_L sobre o modelo $FGG4_L$ permitiu construir uma "ponte" que permite mostrar que o modelo PQ_L é mais forte que o modelo MFA_L^+.

Note-se também que tais resultados podem ser interessantes mesmo para modelos para os quais já se conhecem resultados de não dominância. Em particular seria interessante conhecer descrições dos modelos $CORTE_L$ e PQ_L (e $FGG4_L$) no mesmo espaço de variáveis. A partir de tais modelos seria provavelmente mais fácil perceber-se quais as vantagens em utilizar um ou outro modelo.

De referir ainda que, devido às restrições (26), (27), (28) e (29), também as relaxações lineares das formulações apresentadas nesta secção dependem do nodo escolhido para nodo 1. De facto, a solução indicada acima é admissível para PQ_L se o nodo 1 for considerado como depósito. Tal solução não seria admissível para PQ_L se o nodo 5 fosse considerado como depósito, já que se teria necessariamente $z_{54}^4 = 1$ e $z_{45}^0 = 1/2$. Logo, por (28) ter-se-ia que a restrição (35) no nodo 4 seria violada.

Finalmente, note-se que a relação de não dominância entre os modelos PQ_L e MFD_L (ou $CORTE_L$, ou SUB_L) sugere uma maneira de fortalecer o modelo SUB_L ($CORTE_L$ ou MFD_L).

Basta adicionar a qualquer destes modelos as restrições (35) e as restrições de ligação (23a)-(25a).

Convém salientar que, de acordo com o nosso conhecimento, este trabalho é a primeira referência que indica que modelos compactos envolvendo variáveis com dependências temporais não são dominados pela conhecida formulação de fluxos desagregados *MFD*.

Na figura 7 apresenta-se uma relação entre as relaxações em programação linear das diversas formulações referidas neste trabalho. Recorde-se que não existe qualquer relação de dominância entre os valores fornecidos pela relaxação linear de *MFD* (*SUB* ou *CORTE*) e *PQ* ou *FGG4*.

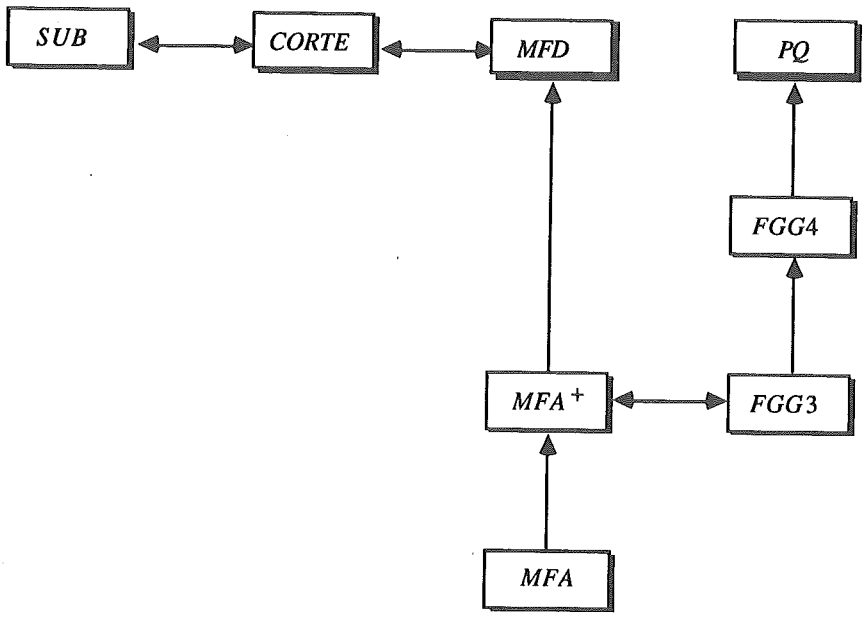


Figura 7 - Relação entre as relaxações lineares dos modelos apresentados. Uma seta de P1 para P2 significa que o valor da relaxação linear de P1 é inferior ao de P2. Uma seta em ambos os sentidos significa que as formulações são equivalentes em termos de relaxação linear. Relações de dominância por transitividade não são aqui assinaladas.

5. O Problema do Caixeiro Viajante com Custos Cumulativos

Os modelos de fluxo e os modelos com dependências temporais, discutidos neste trabalho, podem também servir para modelar um outro tipo de problema, que é uma variante do problema do caixeiro viajante. Este problema é bastante semelhante ao PCVA e difere deste apenas na função objectivo. Considere-se a formulação *MFA* com a seguinte função objectivo:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, y_{ij} \qquad (36)$$

Note-se que a única diferença entre este modelo para o novo problema e o modelo *MFA* apresentado para o PCVA reside no facto de que as variáveis x_{ij} foram substituídas pelas variáveis y_{ij} na correspondente função objectivo. Na nova função objectivo, o custo do primeiro arco é contabilizado n-1 vezes, o custo do segundo arco é contabilizado n-2 vezes, etc. Considere-se a situação em que um veículo sai do nodo 1 e tem que ir buscar clientes (ou produtos) aos nodos 2,..., n antes de voltar ao nodo 1. Suponha-se que cada custo c_{ij} representa o tempo que o veículo demora a atravessar o arco (i,j). A função objectivo assim definida corresponde a minimizar o tempo de espera dos clientes. Este problema é conhecido como o problema do Caixeiro Viajante com Cutos Cumulativos (PCVC) (Lucena (1990) e Fischetti, Laporte e Martello (1993)). Obviamente que a formulação MFA^+ pode ser igualmente adaptada para o problema com custos cumulativos.

É interessante notar que para este problema o papel das variáveis x_{ij} e y_{ij} é invertido, isto é, as variáveis y_{ij} tomam o papel das variáveis naturais enquanto que as variáveis x_{ij} são consideradas variáveis adicionais. Assim, as formulações *SUB* e *CORTE* não têm qualquer sentido para este novo problema. No entanto, os resultados da secção 3 indicam que as restrições de subcircuito (5) envolvidas em *SUB* (ou as restrições de corte (6) envolvidas em *CORTE*) podem ser utilizadas para fortalecer o modelo *MFA* (ou MFA^+) mesmo no âmbito do problema cumulativo.

O mesmo efeito, agora no âmbito de uma formulação compacta pode ser obtida através da inclusão da função objectivo

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=2}^{n} c_{ij} f_{ij}^{k} \qquad (37)$$

no modelo *MFD*. Alternativamente pode manter-se a função objectivo (36) do modelo *MFA* (ou MFA^+) desde que se introduzam as igualdades (14) que relacionam as variáveis de fluxo desagregado com as variáveis de fluxo agregado.

Os modelos com dependências temporais também podem facilmente ser adaptados para este problema. Basta adicionar o coeficiente "t" a cada uma das variáveis z_{ij}^{t} na correspondente função objectivo.

Note-se que todos os modelos estendidos que foram apresentados para o PCVA podem ser facilmente adaptados para a versão com custos cumulativos através de uma simples alteração na função objectivo. Isto significa que todas as relações estabelecidas nas secções anteriores entre diversos modelos para o PCVA se mantêm para o PCVC. No entanto, e como se verá na secção seguinte, existem alterações substanciais na relação empírica entre os valores óptimos das relaxações lineares de alguns dos modelos aqui discutidos quando se passa do PCVA para a versão com custos cumulativos.

De referir que, ao contrário do PCVA, o PCVC é dependente do nodo seleccionado para depósito. Por fim, convém salientar que para este problema não se conhecem ainda formulações naturais, isto é, formulações que envolvam apenas as variáveis y_{ij}.

6. Resultados computacionais

Até agora apresentaram-se resultados teóricos que indicam que a relação $\nu(F1) \geq \nu(F2)$ se verifica sempre para diversos pares de modelos, em programação linear, $F1$ e $F2$. No entanto, tais resultados não permitem avaliar as diferenças, porventura existentes, entre os valores óptimos de $F1$ e $F2$. Nesse sentido, apresentam-se alguns resultados computacionais relativos aos modelos discutidos neste trabalho. Consequentemente, mostra-se também que a dominância empírica entre as duas formulações se pode alterar drasticamente quando a função objectivo se altera (caso do PCVA e do PCVC). Apresentam-se testes apenas para instâncias de dimensão reduzida pelas razões já evocadas de que se pretende apenas analisar e examinar o que se pode ganhar por "navegar" no conjunto de formulações possíveis para um dado problema (neste caso para o PCVA e o PCVC). Não se aborda o facto de tentar determinar qual a melhor formulação para obter a solução óptima inteira (ou pelo menos a melhor formulação para um dado conjunto de instâncias) já que modelos que podem ser considerados não atractivos quando se utiliza a respectiva relaxação linear podem tornar-se bastante atractivos em face de outras técnicas, nomeadamente, técnicas de relaxação lagrangeana. Os valores óptimos da relaxação foram obtidos utilizando o pacote de programação linear CPLEX, versão 3.0, e os testes foram realizados num PC Pentium a 50MHz e com 16 MB de memória RAM.

Foram utilizados testes assimétricos TA e testes simétricos euclideanos e não euclideanos, respectivamente, TSE e TS, relativos a instâncias com 20 nodos. Não foram consideradas instâncias de maior dimensão devido à dificuldade de resolução do modelo MFD_L. Resultados relativos a instâncias de 40 nodos para as formulações comdependências temporais podem ser encontrados em Pires (1994). Os resultados que se apresentam nas tabelas 1 e 2, respectivamente para o PCVA e PCVC, reportam à média aritmética, sobre 5 testes, dos valores fornecidos por $(opt(F) - \nu(F))/\nu(F)$, onde $opt(F)$ representa o valor óptimo da instância e $\nu(F)$ o valor da respectiva relaxação em programação linear. Para obter os valores óptimos utilizou-se a versão de programação inteira do CPLEX. Esses valorem foram obtidos a partir do modelo MFD_L para o caso do PCVA e a partir do modelo PQ_L para o caso do PCVC. Nas tabelas 1a e 2a apresentam-se os tempos obtidos, em média, por cada um dos modelos. Os tempos relativos à obtenção da solução óptima foram, em média, para o PCVA, de 633.444, 0.286 e 129.32 segundos, respectivamente para os testes TA, TS e TSE. Para o caso do PCVC, esses tempos foram, pela ordem referida, de 12.03, 544.246 e 1414.106.

n	Tipo de Teste	MFA_L	MFA_L^+	$FGG4_L$	PQ_L	MFD_L
20	TA	0.018	0.017	0.017	0.012	0.008
	TS	0.080	0.076	0.075	0.072	0.000
	TSE	0.089	0.083	0.081	0.075	0.001

Tabela 1 - Resultados para o PCVA para instâncias de 20 nodos

n	Tipo de Teste	MFA_L	MFA_L^+	$FGG4_L$	PQ_L	MFD_L
20	TA	0.716	0.610	0.070	0.022	0.556
	TS	0.752	0.653	0.230	0.164	0.572
	TSE	0.670	0.618	0.134	0.096	0.520

Tabela 2 - Resultados para o PCVC para instâncias de 20 nodos

n	Tipo de Teste	MFA_L	MFA_L^+	$FGG4_L$	PQ_L	MFD_L
20	TA	0.406	0.802	5.138	9.108	131.658
	TS	0.484	0.912	4.556	9.546	206.804
	TSE	0.526	0.990	5.776	6.536	179.880

Tabela 1a - Tempos de execução (em segundos) para o PCVA para instâncias de 20 nodos

n	Tipo de Teste	MFA_L	MFA_L^+	$FGG4_L$	PQ_L	MFD_L
20	TA	0.330	1.704	9.414	11.030	27.526
	TS	0.320	1.700	7.548	10.206	31.482
	TSE	0.342	1.968	6.624	11.042	66.910

Tabela 2a - Tempos de execução (em segundos) para o PCVC para instâncias de 20 nodos

Relativamente ao PCVA, os resultados da tabela 1 indicam que a diferença entre os valores fornecidos pela relaxação linear dos modelos de fluxo agregado e do modelo de fluxo desagregado é consideravelmente maior quando se consideram testes euclidianos. Em geral, uilizando as relaxações dos modelos analisados, os problemas assimétricos são mais fáceis de resolver que os problemas simétricos. Por sua vez, os problemas simétricos aleatórios são mais fáceis que os auclideanos. Note-se também que o PCVA parece mais simples de resolver que o PCVC. É interessante notar que as formulações com dependências temporais se mostram de uma qualidade bastante superior à da formulação *MFD* no caso do PCVC. O contrário se passa no que diz respeito ao PCVA. Estes resultados mostram que uma modificação na função objectivo de várias formulações para um mesmo problema pode alterar consideravelmente a relação empírica entre as correspondentes relaxações lineares. Em particlar, é interessante notar a diferença nos valores das relaxações lineares das formulações MFA^+ (note-se que a relaxação linear desta é equivalente à relaxação linear de *FGG*3) e *FGG*4. No caso do PCVC, é natural esperar que nas soluções óptimas de $FGG3_L$, as restrições (34) para t grande sejam violadas por cima, enquanto que o contrário se deve esperar, relativamente às mesmas restrições, para t pequeno. Assim, espera-se um efeito substancial por inclusão de tais restrições no modelo $FGG3_L$. No caso do PCVA, a inclusão das restrições (34) em $FGG3_L$ pouco efeito produz.

7. Conclusões

Neste trabalho utilizou-se o conhecido problema do Caixeiro Viajante e uma sua variante com custos cumulativos para ilustrar algumas técnicas de reformulação de modelos em programação linear inteira. Note-se também que a maior parte das técnicas aqui representadas podem facilmente estender-se a outros tipos de problemas de desenho óptimo de redes.

Referências

[1] Ahuja, A.K., Magnanti, T.L. and Orlin, J.B., *Network Flows*, Prentice Hall, Englewood Cliffs, New Jersey (1993).

[2] Barros, A.I. e Barcia, P., *New Bounds for the ATSP*, apresentado na escola de verão da EURO na Universidade de Bilkent, Turquia, Julho 1991.

[3] Dantzig, G.B., Fulkerson, D.R. and Johnson, S.M., *Solution of a Large Scale Travelling Salesman Problem*, Operations Research 2 (1954) 393-410.

[4] Fischetti, M., Laport, G. and Martello, S., *The Delivery Man Problem and Cummulative Matroids*, Operations Research 41 (1993) 1055-1064.

[5] Ford, L.R. and Fulkerson, D.R., *Flows in Networks*, Princeton University Press, Princeton NJ (1962).

[6] Fox, K.R., Gavish, B. and Graves, S.C., *A N-Constrained Formulation of the (Time-dependent) Traveling Salesman Problem*, Operations Research 28 (1980) 1018-1021.

[7] Gavish, B. and Graves, S.C., *The Travelling Salesman Problem and Related Problem*, Working Paper, April 1979.

[8] Gouveia, L. and Voβ, S., *A Classification of Formulations for the (Time-Dependent) Traveling Salesman Problem*, European Journal of Operations Research 83 (1995) 69-82.

[9] Grotschel, M. and Padberg, M.W., *Polyhedral theory*, in The Traveling Salesman Problem - A Guided Tour of Combinatorial Optimization. Lawler, E.L., Lenstra, J.K., Rinnooy, A.H.G. and Shmoys, D.B. (eds). Joh Wiley and Sons Ltda. (1985).

[10] Lagevin, A., Soumis, F. and Desrosiers, J., *Classification of Traveling Salesman Problem Formulations*, Operations Research Letters 9 (1990) 127-132.

[11] Lucena, A., *Time-Dependent Traveling Salesman Problem - The Deliveryman Case*, Networks 20 (1990) 753-763.

[12] Miller, C.E., Tucker, A.W. and Zemlin, R.A., *Integer Programming Formulation of Traveling Salesman Problems*, Journal of the Association for Computing Machinery 7 (1960) 326-329.

[13] Padberg, M.W. and Grotschel, M., *Polyhedral Computations*, in the Traveling Salesman Problem - A Guided Tour of Combinatorial Optimization. Lawler, E.L., Lenstra, J.K., Rinnooy, A.H.G. and Shmoys, D.B. (eds). John Wiley and Sons Ltda. (1985).

[14] Padberg, M. and Rinaldi, G., *A Branch-and-Cut Algorithm for the resolution of Large-Scale Simmetric Traveling Salesman Problems*, Siam Review 33 (1991) 60-100.

[15] Picard, J.C. and Queyranne, M., *The Time-Dependent Traveling Salesman Problem and its Application to the Tardiness Problem in One-Maching Scheduling*, Operations Research 26 (1978) 86-110.

[16] Pires, J.M., *Uma Classificação de Fromulações para o Problema do Caixeiro Viajante com Dependências de Tempo - Experiência Computacional*, Dissertação de Mestrado, Universidade Técnica de Lisboa (1994).

[17] Pulleyblank, W., Polyhedral Combinatorics in Handbooks in OR & MS, Vol. 1 (1989), North-Holland.

[18] Rardin, R. and Choe, U., *Tighter Relaxations of Fixed Carge Network Flow Problems*, Report 3-79-18, Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta (1979).

[19] Wong, R.T., *Integer Programming Formulations of the Travelling Salesman Problem*, Proceedings of the IEEE International Conference of Circuits and Computers (1980) 149-152.

INVESTIGAÇÃO OPERACIONAL, vol. 16 (Dezembro 1996) 115-133

APLICAÇÃO DE META-HEURÍSTICAS A PROBLEMAS DE ESCALONAMENTO DE UMA ÚNICA MÁQUINA

Ana Maria Madureira
ISEP - Instituto Superior de Engenharia do Porto

Jorge Pinho de Sousa
INESC - Instituto de Engenharia de Sistemas e Computadores
FEUP - Faculdade de Engenharia da Universidade do Porto

Abstract

In this paper, we describe some general features of single-machine scheduling problems, and we use some of their structural properties to design local search procedures based on alternative definitions of neighbourhoods.

In particular, the traditional idea of "pairwise interchanges" is replaced by the idea of "exchanging jobs not apart more than a given number of positions (considered as a parameter of the algorithm)".

For generating initial solutions, we have tested some traditional priority rules, with some degree of randomization introducing, in general, a positive effect in the performance of the algorithms.

Through a set of computational tests, we have evaluated the importance of the different parameters, and we have tuned their values for different meta-heuristic procedures (simulated annealing, taboo search, and "randomized local search"). Though these tests have been exhaustive only for a given problem ("weighted tardiness"), the results already available show these approaches are robust and flexible, and that, in general, we obtain satisfactory solutions in an efficient way.

Resumo

Neste artigo, são inicialmente descritas algumas características gerais dos problemas de escalonamento de uma única máquina, e exploradas algumas das suas propriedades, com vista à definição de procedimentos de pesquisa local baseados em conceitos de "vizinhança" alternativos.

Em particular, o tradicional conceito de "troca de posição entre duas tarefas" é substituído pela ideia de "troca de tarefas não distantes mais do que um dado afastamento (parametrizável)".

Para a construção de soluções iniciais, foram testadas algumas regras de prioridade tradicionais, tendo adicionalmente sido introduzida uma aleatorização parcial dessas regras, com um efeito em geral positivo, no desempenho dos algoritmos.

Através de um conjunto de experiências, foi avaliado o impacto dos diferentes parâmetros e afinados os seus valores para utilização num conjunto de meta-heurísticas ("simulated annealing", "taboo search" e pesquisa local "aleatorizada"). Embora os testes computacionais realizados tenham sido exaustivos relativamente a um problema particular (minimização dos atrasos pesados - "weighted tardiness"), uma primeira avaliação mais global, feita com um conjunto de outros problemas, parece demonstrar a robustez e a flexibilidade destas abordagens, bem como a qualidade das soluções obtidas.

Keywords

Machine scheduling; local search; meta-heuristics.

1. Problemas de escalonamento de uma máquina

O planeamento de sistemas produtivos envolve frequentemente a resolução de problemas de escalonamento (*"scheduling"*) com um impacto significativo no desempenho das organizações. Tais problemas consistem basicamente em, dado um conjunto limitado de recursos, genericamente designados por máquinas ou processadores, determinar a sua utilização ao longo do tempo, de modo a processar um conjunto de tarefas independentes ou inter-relacionadas, com vista à satisfação de objectivos de natureza económica ou operacional.

O problema de escalonamento mais elementar verifica-se sempre que um conjunto de tarefas está disponível simultaneamente para processamento num só recurso ou máquina. Assume-se aqui um contexto determinístico, considerando-se que os tempos de processamento e as datas de entrega das tarefas estão definidos e são independentes da sequência de processamento escolhida. Neste caso, a ordenação das tarefas determina completamente o respectivo escalonamento (Baker [1974] e French [1982]).

O estudo dos problemas de escalonamento de uma máquina única (*"single machine"*) justifica-se por diversas razões, nomeadamente de natureza metodológica, já que uma boa compreensão do problema de uma máquina única ajudar-nos-á a compreender e a modelar problemas de escalonamento mais complexos. Tal poderá resultar não só da extensão de alguns conceitos, mas também do caso de uma única máquina constituir um elemento de construção e compreensão dos processos mais complexos de escalonamento.

Neste sentido, e tendo como finalidade modelar o comportamento de um sistema mais complexo, é importante compreender o funcionamento dos seus componentes que poderão eventualmente ser encarados como problemas de uma única máquina. Por vezes, é resolvido um problema mais elementar, independentemente, e só depois incorporado o resultado no problema principal. Por exemplo, em modelos com várias máquinas pode existir uma máquina crítica cuja capacidade de processamento é inferior à necessária (*"bottleneck"*). A sua análise e tratamento como problema de uma única máquina pode influenciar as decisões de escalonamento posteriores.

Note-se que as técnicas definidas e testadas para este tipo de problemas constituem frequentemente ferramentas eficientes para a resolução não só de problemas com múltiplas máquinas mas também de diferentes classes de problemas mais gerais de escalonamento.

No problema de escalonamento de uma única máquina (*"single machine"*) considera-se em geral que:

- as tarefas ($j = 1,\ldots, n$) são independentes e caracterizadas por tempos de processamento p_j previamente conhecidos;
- no instante zero a máquina está disponível para o processamento de todas as tarefas;
- os tempos de *"setup"* para as tarefas são independentes da sequência escolhida e podem ser considerados no tempo de processamento;

- a máquina está continuamente disponível e nunca será mantida parada com tarefas à espera de serem processadas;
- e, uma vez iniciado o processamento de uma tarefa, esta é processada até ao fim sem interrupções.

Sob estas condições, existe uma correspondência unívoca entre uma sequência de *n* tarefas e uma permutação nos índices das tarefas *1, 2,..., n*. O número total de soluções diferentes para o problema da máquina única é portanto *n!*.

Poderão ainda ser definidas para as tarefas algumas restrições adicionais, como por exemplo, considerar que a tarefa *j* tem uma data de lançamento (*"release date"*) r_j, o instante de tempo no qual esta se torna disponível para processamento, e uma data de entrega absoluta (*"deadline"*) d_j que é o instante de tempo no qual a tarefa deve estar terminada.

Critérios de optimalidade

Um calendário (*"schedule"*) de execução das tarefas (associado a uma dada sequência) é completamente descrito por um conjunto de *n* tempos de início t_j (se assumirmos que não é permitida interrupção) e consequentemente o valor ou custo de uma solução depende basicamente dos tempos de início de processamento da cada umas das tarefas ou dos seus **tempos de conclusão:** $\mathbf{C_j = t_j + p_j}$. Na prática, podem considerar-se diferentes critérios associados a esses tempos.

O atraso ($\mathbf{L_j}$ - *"lateness"*) mede a conformidade das tarefas de uma sequência com determinadas datas desejadas de entrega d_j (*"due date"*): $\mathbf{L_j = C_j - d_j}$. Note-se que o atraso toma valores negativos sempre que uma tarefa é completada mais cedo do que a respectiva data de entrega. Em muitas situações, associam-se penalizações distintas aos atrasos positivos (*"tardiness"*) e aos atrasos negativos (*"earliness"*).

O atraso "positivo" (*"tardiness"*) representa um atraso na conclusão de processamento de uma tarefa, sendo por isso referido ao longo deste trabalho apenas como *atraso* (atraso positivo): $\mathbf{T_j = max\ \{0, L_j\}}$.

A função objectivo (a minimizar) baseia-se frequentemente num destes critérios (ou noutros definidos de forma semelhante) que são uma função dos tempos de conclusão das tarefas, $f_j(C_j)$, e assume em geral um dos seguintes formatos:

- $\sum f_j = \sum f_j(C_j)$ – como, por exemplo, a minimização da soma pesada do número de tarefas em atraso $\sum w_jU_j$ e da soma pesada dos atrasos $\sum w_jT_j$ sendo w_j a penalização associada aos atrasos;
- $f_{max} = max_{j=1,2,...,n}\ f_j(C_j)$ - como, por exemplo, a minimização do "makespan" total, C_{max}, ou do atraso máximo L_{max}.

Os critérios de avaliação referidos são designados por critérios regulares, isto é, são funções reais não-decrescentes para qualquer tempo de conclusão. Sendo $f(C_1,..., C_n)$ o valor de uma determinada sequência de tarefas e $f(C_1',..., C_n')$ o custo de uma outra sequência, se $f(C_1,..., C_n) < f(C_1',..., C_n')$ então $C_j < C_j'$ para pelo menos uma tarefa *j*.

Tal leva a que para qualquer sequência, fica associado, de uma forma única, um dado calendário.

Minimização da soma pesada dos atrasos

Considere-se o problema de escalonamento de n tarefas numa única máquina, estando a cada tarefa j associado um tempo de processamento p_j, uma data de entrega pretendida d_j e uma penalização w_j por cada unidade de tempo em atraso. Pretende-se encontrar uma sequência de tarefas que minimize a soma pesada dos atrasos (WT-*"weighted tardiness"*) por w_j. O processamento da primeira tarefa da sequência começa no instante de tempo t = 0. O atraso de uma dada tarefa na sequência é dado por $T_j = \max\{0, t_j+p_j-d_j\}$, sendo t_j o tempo de início de processamento da tarefa j. Assim sendo, pretende-se encontrar a sequência que conduz a:

$$\min \sum w_j T_j, \quad \text{com} \quad T_j = \max\{t_j+p_j-d_j, 0\}$$

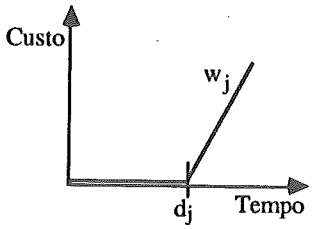


Figura 1 - Contribuição da tarefa j para o custo total da solução

Na prática, este problema pode ser consideravelmente simplificado pela aplicação sistemática de um conjunto de regras (de "dominância") que nos permitem fixar a posição de algumas tarefas ou estabelecer relações de precedência entre pares de tarefas, e que deverão ser aplicadas numa fase de pré-processamento que é, em geral, essencial para uma resolução eficiente dos problemas. No entanto, neste trabalho, não é feita qualquer utilização dessas regras, já que o objectivo essencial consiste antes em avaliar, em termos genéricos, os procedimentos de pesquisa em estudo.

2. Meta-Heurísticas

As meta-heurísticas caracterizam-se basicamente por constituírem algoritmos que iteram e aleatorizam os procedimentos de pesquisa local "pura" (baseados em conceitos de vizinhança e de exploração do espaço das soluções), com o objectivo de ultrapassar as limitações deste tipo de pesquisa (à custa de um maior esforço computacional). A eficiência destes algoritmos depende de inúmeros factores, como as características do problema a optimizar, a estrutura da vizinhança, os procedimentos ou regras heurísticas utilizadas na geração das soluções.

Nos últimos anos, os algoritmos heurísticos genéricos (do tipo meta-heurísticas) para a resolução de problemas de optimização combinatória têm sido alvo de grande atenção (ver, por exemplo, Pirlot [1992]).

A sequência de soluções de um problema, percorrida pelo algoritmo, constitui uma *trajectória* no espaço das soluções. O conjunto de soluções que se podem atingir a partir de cada uma das soluções do problema é habitualmente designado por *vizinhança* de uma solução. Considerando um problema caracterizado pelos argumentos (X, *F*), em que X é o conjunto, discreto, de soluções admissíveis e *F* a função custo associada (que se pretende minimizar), uma vizinhança *N* é uma função do tipo N: $X \rightarrow 2^X$. Isto implica a definição da estrutura da vizinhança em X, de tal modo que para cada $x \in X$ é associado um subconjunto $N(x) \subseteq X$ designado *vizinhança* de x. Por convenção, supomos que nenhuma solução é vizinha de si mesma, isto é, $x \notin N(x)$, $\forall x \in X$. Podemos dizer que os vizinhos de x são todas as soluções que podem ser obtidas a partir de x por movimentos elementares. A evolução da solução actual x_n ($n = 1, 2,\ldots$) define a trajectória em X.

Pesquisa local

Na estratégia de pesquisa local, parte-se de uma solução inicial arbitrária (ou eventualmente determinada por uma heurística construtiva) $x_1 \in X$ e em cada iteração *n*, é escolhida uma nova solução x_{n+1} na vizinhança $N(x_n)$ da solução actual x_n. Em geral, é escolhida uma solução que melhore o valor da função objectivo, isto é, a solução $x_{n+1} \in N(x_n)$ é tal que $F(x_{n+1}) \leq F(x)$, $\forall x \in N(x_n)$. É o movimento de escolha e deslocação para uma nova solução que obedece a diferentes critérios consoante o procedimento de pesquisa local utilizado. Assim sendo, num algoritmo de pesquisa local *"pura"*, designado aqui simplesmente por pesquisa local (*"local search"*), x_{n+1} torna-se, na iteração seguinte, solução actual se não for pior que a solução x_n, isto é, se $F(x_{n+1}) \leq F(x_n)$. Caso contrário, o procedimento de pesquisa é terminado.

Esta estratégia é também usualmente designada por *descendente*, e o algoritmo apresentado é do tipo *"melhor vizinho"*, no sentido em que é escolhida a melhor solução da vizinhança. É por vezes utilizada uma variante deste algoritmo que consiste na geração sucessiva de soluções vizinhas até ser encontrada uma melhor que a solução actual. Esta variante é conhecida por algoritmo de pesquisa local do tipo *"primeiro vizinho melhor"* e tem em vista produzir um procedimento mais eficiente (ainda que à custa da qualidade da solução obtida).

É de salientar que a escolha da estrutura de vizinhança é de extrema importância para a eficiência do processo. A principal desvantagem deste algoritmo e das suas variantes é, no entanto, a sua incapacidade em escapar aos *mínimos locais*, e que justifica o recurso a meta-heurísticas do tipo *"simulated annealing"* e *pesquisa tabu.*

Exemplo 1

Considere-se o problema de sequenciamento de 5 tarefas numa única máquina. Para cada tarefa *j (j = 1,..., 5)*, seja p_j o tempo de processamento, d_j a data de entrega e w_j a penalização no caso da tarefa j se atrasar. O objectivo é minimizar a soma pesada dos atrasos.

Tarefa j	p_j	d_j	w_j
1	2	5	1
2	4	7	6
3	1	11	2
4	3	9	3
5	3	8	2

O algoritmo é iniciado com a construção de uma solução inicial usando, por exemplo a regra EDD (*"earliest due dates"*), resultando a solução inicial x_1 e respectivo valor para a função objectivo $F(x_1)$:

$x_1 = [1\ 2\ 5\ 4\ 3]$ $\quad F(x_1) = 15$

A vizinhança escolhida é constituída por um conjunto de soluções obtidas trocando pares de tarefas adjacentes a partir da solução inicial. Temos assim que a vizinhança é constituída por *n-1* soluções, em que *n* é o número de tarefas. Nesta caso, a vizinhança é constituída pelas 4 soluções seguintes:

Vizinhança N(xn)	Valor de F(xn)
2 1 5 4 3	16
1 5 2 4 3	25
1 2 4 5 3	12
1 2 5 3 4	14

A melhor solução encontrada na 1ª iteração (associada à pesquisa de toda a vizinhança) é a sequência x* = [1 2 4 5 3] cujo valor é F* = 12. Esta sequência será a solução inicial da iteração seguinte. O procedimento sera repetido iterativamente até não ser encontrada nenhuma solução melhor que a actual.

♦

Pesquisa local aleatorizada (PLA)

Este procedimento é uma extensão da *pesquisa local*, e consiste em executar *n* vezes o algoritmo, a partir de soluções iniciais geradas aleatoriamente. O objectivo do algoritmo é ultrapassar as limitações referidas do algoritmo original de *pesquisa local*. Este tipo de procedimentos pode assumir várias formas, como, por exemplo, as heurísticas designadas por GRASP (ver, por exemplo, Feo et al [1989]).

"Simulated Annealing" (SA)

A motivação original destes algoritmos esteve na termodinâmica e na metalurgia, tendo por base o processo em que um metal em fusão é arrefecido lentamente, tendendo a solidificar numa estrutura de energia mínima. A mesma "lógica" é utilizada para os problemas de Optimização Combinatória no *"simulated annealing"*. No início do processo iterativo, quase todos os movimentos são aceites, isto é, são consideradas todas as actualizações da solução actual por uma solução x aleatoriamente escolhida na sua vizinhança, o que permite explorar, de uma forma mais "exaustiva", o espaço das soluções. Depois, e de uma forma gradual, a "temperatura" é diminuída, tornando o algoritmo progressivamente mais selectivo na aceitação

de uma nova solução. No fim, só os movimentos que melhoram o valor de *F* são aceites (Pirlot [1992]).

Os algoritmos de *"simulated annealing"* não procuram a melhor solução na vizinhança $N(x_n)$ da *solução actual* x_n, mas consideram uma solução aleatória x na vizinhança $N(x_n)$. Se $F(x) \leq F(x_n)$, x torna-se a nova solução actual. Se tal não se verificar, então é escolhida uma das duas alternativas seguintes: x torna-se a solução actual, com probabilidade *p(n)*, ou x_n permanece como solução actual, com probabilidade *1-p(n)*. Normalmente, *p(n)* depende da diferença de valor da função objectivo entre duas soluções e do valor da temperatura. Deve ser crescente com a temperatura e decrescente com o valor da deterioração $F(x)-F(x_n)$.

Com o objectivo de tornar o algoritmo operacional, devem ser tomadas algumas decisões genéricas, envolvendo nomeadamente a escolha da probabilidade de aceitação de soluções piores, o esquema de arrefecimento, a forma de calcular a temperatura inicial, a actualização da temperatura, o número de iterações que são realizadas à mesma temperatura e o critério de paragem do algoritmo.

Pesquisa tabu ("Taboo Search" - TS)

Os procedimentos de *pesquisa tabu* (*"taboo search"*) constituem uma outra estratégia de pesquisa local concebida com o objectivo de escapar aos mínimos locais. Este método de pesquisa "inspira-se" nalgumas regras simples de aprendizagem, simulando num certo sentido, alguns aspectos do comportamento humano. O que fundamentalmente distingue a pesquisa tabu de uma pesquisa local tradicional é a utilização do conceito de memória (Pirlot [1992]).

Na *pesquisa tabu* procura-se fugir a um mínimo local onde eventualmente se tenha caído, escolhendo para solução seguinte, a partir da solução actual x_n, a melhor solução possível x pertencente à sua vizinhança $N(x_n)$, ou a uma subvizinhança $N'(x_n) \subseteq N(x_n)$. Isto acontece mesmo que a nova solução seja pior que a solução actual, isto é, $F(x) > F(x_n)$.

Se a estrutura da vizinhança for simétrica, isto é, se x_n pertencer à vizinhança N(x) de x quando $x_n \in N(x)$, há o perigo de se criar um ciclo, quando no próximo passo se explorar N(x). A partir daqui, o algoritmo seria, em geral, incapaz de evoluir noutro sentido.

Para evitar esta e outras situações de ciclo, a ideia é guardar os últimos pares (x_n, x) de soluções visitadas em sequência, numa lista designada por *lista tabu*. Se o par (x_n, x) está na lista, o movimento $x \rightarrow x_n$ fica interdito durante um certo número de iterações subsequentes. Se o tamanho da *lista tabu* for limitado às últimas k soluções visitadas, consegue-se evitar ciclos de tamanho não superior a k: sempre que uma nova solução é visitada, entra para a *lista tabu* e sai a mais antiga. Finalmente, os critérios de paragem do algoritmo baseiam-se, em geral, no facto de o valor da função objectivo não melhorar nas últimas *N* iterações ou num número total pré-definido de iterações.

Estes princípios gerais comportam, na prática, alguns problemas técnicos, associados nomeadamente ao armazenamento da descrição completa das últimas soluções visitadas e à necessidade de testar para cada movimento candidato se é inverso do movimento guardado na

lista, o que corresponde a operações que podem ser computacionalmente demoradas. Uma alternativa é considerar na *lista tabu*, não os pares de soluções (x_n, x), mas apenas uma ou mais características dos movimentos. Assim, a *lista tabu* é usualmente uma lista de um ou mais atributos da solução recentemente visitada ou do movimento mais recente, devendo, por razões de eficiência computacional, estes atributos ser escolhidos com o maior cuidado.

Na prática, a "proibição" de soluções com um dado atributo é frequentemente demasiado restritiva. Por exemplo, se o movimento de uma solução para outra consiste em trocar as tarefas das posições 3 e 5, é inserida na *lista tabu* a transformação que consiste em trocar de posição os elementos 3 e 5, independentemente da sequência ou solução de origem, isto é, são excluídas muito mais soluções do que aquela que foi de facto visitada.

Para corrigir as consequências indesejáveis deste procedimento, a *pesquisa tabu* possui um mecanismo que permite contrariar nalguns casos a restrição imposta pelo facto de um movimento ser *tabu*, e aceitá-lo. Este mecanismo é implementado através do *nível de aspiração*, o qual define o que é, em cada momento, uma solução suficientemente boa. É adoptado normalmente um dos dois seguintes critérios de aspiração: se a solução gerada é melhor que a melhor solução encontrada até ao momento, ou se a solução gerada a partir de uma *transformação tabu* é suficientemente boa porque melhora o valor da função objectivo, então a solução é aceite.

As decisões que devem ser tomadas para implementar um algoritmo de *pesquisa tabu* são em maior número, menos normalizadas e menos suportadas por resultados teóricos do que as do *"simulated annealing"*. No entanto, na aplicação de algoritmos de *pesquisa tabu* em problemas de optimização combinatória há maior espaço para a criatividade por um lado, e para a experimentação por outro. As principais decisões a serem tomadas são: a especificação da estrutura da vizinhança, a escolha dos atributos dos movimentos a serem guardados na *lista tabu*, a definição do tamanho da *lista tabu*, a escolha do critério de aspiração e a selecção do critério de paragem.

3. Problemas teste e estudo computacional

Os resultados apresentados neste artigo têm em vista ilustrar e avaliar as ideias descritas, aplicando-as ao problema de minimização da soma pesada dos atrasos.

Em particular, foi estudada a influência de vários factores (regra de construção da solução inicial, mecanismo gerador de vizinhanças) no desempenho dos métodos de pesquisa local, para a resolução do problema referido. Depois de definidos e afinados alguns parâmetros, foi realizado um conjunto de testes computacionais com o objectivo de avaliar e comparar a qualidade das soluções obtidas por diferentes meta-heurísticas.

Finalmente, refira-se que os algoritmos foram implementados em C, sendo os testes computacionais executados num computador pessoal 386 a 25 Mhz.

Geração aleatória de problemas

Foi desenvolvida uma ferramenta informática ("gerador") para gerar aleatoriamente instâncias de problemas de *"minimização da soma pesada dos atrasos"*.

Os dados de entrada do gerador são: n - número de tarefas do problema, *Sd* - semente (*"seed"*) do processo de geração, *Tp* - limite superior da gama de valores para o tempo de processamento p_j, *Pe* - limite superior para os pesos w_j e o valor do parâmetro α (abaixo referido). Assume-se que os tempos de processamento, as datas de entrega e os pesos associados a um atraso são inteiros não negativos. Para cada uma das n tarefas, são gerados:

- um *tempo de processamento* p_j, segundo uma distribuição uniforme U[1,Tp];
- uma *data de entrega* d_j, distribuída uniformemente no intervalo $[1,\alpha\sum p_j]$, com $0 < \alpha < 1$;
- *um peso* w_j, distribuído uniformemente no intervalo [1,*Pe*].

4. Solução inicial

Com vista a avaliar o impacto ou uma possível relação entre algumas regras usadas tradicionalmente na construção de uma solução inicial e a obtenção de uma solução final de melhor qualidade, isto é, mais próxima da solução óptima global do problema, foram efectuados alguns testes computacionais utilizando o algoritmo de pesquisa local (*"local search"*) em instâncias de problemas do tipo WT (*"weighted tardiness"*), com 20 e 30 tarefas (aqui designadas por WT20A, WT20B, WT20C, WT30A e WT30B) e apresentadas em Sousa e Wolsey [1992].

Dado o carácter aleatório incorporado em algumas dessas regras e na geração de subvizinhanças dos testes computacionais, os resultados encontrados, para cada instância, são referentes a 12 corridas do algoritmo de pesquisa local. Para cada uma das instâncias são apresentados os valores máximos e mínimos, os valores médios (das soluções das 12 corridas e das 75% melhores soluções) e o desvio em relação ao valor óptimo de cada uma das instâncias.

O critério definido para seleccionar as 75% melhores tem em vista evitar eventuais comportamentos extremos do algoritmo, tendo em conta o importante grau de aleatoriedade introduzido.

Serão considerados ao longo deste trabalho dois tipos de regras ou procedimentos heurísticos para construção de uma solução inicial:

- ***determinísticos*** (a cada regra corresponde uma determinada solução ou sequência);
- ***aleatorizados*** (possibilitando a geração de soluções iniciais diferentes).

Nos procedimentos ***determinísticos*** procura-se aproveitar as caraterísticas específicas da função objectivo em causa, tendo sido objecto de estudo as seguintes regras de prioridades:

- SPT (*"Shortest Processing Time"*);
- EDD (*"Earliest Due Date"*);
- a regra D_j/W_j.

Nos procedimentos ***aleatorizados*** será considerada uma regra designada por RND (a ordem das tarefas na sequência inicial é seleccionada aleatoriamente), e uma versão aleatorizada da regra EDD que será referenciada por REDD (*"Randomized Earliest Due Date"*) e que se explica de seguida.

Aleatorização da solução EDD

A ideia da introdução de uma perturbação aleatória numa solução EDD surgiu da tentativa de simular uma situação intermédia entre uma solução à partida "boa", construída de uma forma determinística e soluções completamente aleatórias, tentando desta forma tirar vantagens da solução EDD e simultaneamente evitar os óptimos locais. A componente aleatória da regra REDD é introduzida pela troca de pares de tarefas seleccionadas à sorte.

Exemplo 2

No exemplo de sequenciamento de 5 tarefas numa única máquina que se tem vindo a considerar (com as datas de entrega: $d_1 = 5$, $d_2 = 7$, $d_3 = 11$, $d_4 = 9$ e $d_5 = 8$), a sequência EDD correspondente é [1 2 5 4 3]. A sequência REDD poderá ser obtida da aleatorização da solução EDD [1 2 5 4 3] pela troca de, por exemplo, 2 pares de tarefas seleccionadas à sorte, da seguinte forma: trocando as tarefas 5 e 3 obtem-se a solução [1 2 3 4 5], e finalmente se se trocar as tarefas 1 e 5, obtém-se a solução REDD pretendida [5 2 3 4 1].

♦

Foram realizados vários testes sobre os problemas de 20 e 30 tarefas anteriormente referidos, tentando-se avaliar a influência do factor de perturbação aleatória numa sequência EDD pela troca de 1 par, 4 pares e 7 pares de tarefas, seleccionados à sorte.

Os resultados computacionais, referentes a 12 corridas do algoritmo de pesquisa local partindo de diferentes soluções iniciais REDD, não são completamente conclusivos, julga-se que devido à componente aleatória que lhes está inerente. No entanto, esses resultados permitem mesmo assim, perceber a existência de um efeito positivo na introdução de alguma perturbação aleatória numa solução determinística EDD. O efeito positivo da aleatorização traduz-se numa redução dos desvios do valor médio em relação ao óptimo global, comparativamente com os desvios obtidos por sequenciação EDD (observada particularmente nos problemas WT20C, WT30A e WT30B).

Embora não resulte claro qual o nível de aleatorização a utilizar na prática, parece ser possível relacionar a dimensão do problema com o nível de aleatoriedade a introduzir, de modo a serem conseguidas melhorias significativas em termos de eficácia. Assim sendo, nos restantes testes computacionais, para os problemas mais pequenos (20 e 30 tarefas), a perturbação aleatória será introduzida pela troca de 1 a 4 pares, e para os problemas de maior dimensão pela troca de 4 até 7 pares de tarefas.

Estudo comparativo das regras de sequenciamento

O objectivo desta fase do trabalho foi efectuar um estudo comparativo das regras para construção de uma solução inicial já mencionadas (SPT, EDD, regra D_j/W_j, RND e REDD). Para tal foi, para cada uma destas regras, executado o algoritmo de pesquisa local 12 vezes em cada instância dos problemas anteriormente referidos.

Para o efeito, foi necessário fazer algumas outras opções (ver secção 5), tendo-se definido como mecanismo de geração de vizinhanças a troca de duas quaisquer tarefas com um afastamento máximo entre si que é definido em função do tamanho do problema (neste caso, 25% do número de tarefas). Para além disso, e dada a dimensão significativa dos problemas, optou-se por trabalhar com subvizinhanças constituídas por 25% das soluções, obtidas aleatoriamente na vizinhança do problema (ver também secção 5).

Da análise global dos resultados, verificou-se que, de um modo geral, a solução inicial não influencia significativamente a "qualidade" da solução final, isto é, não é claro que partindo de uma solução inicial cujo valor é mais próximo da solução óptima do problema, se obtenha uma solução melhor do que se partirmos de uma solução inicial pior.

Apesar de tudo, e na impossibilidade de ser encontrada uma regra ou procedimento heurístico de ordenação de tarefas garantidamente melhor, procurou-se identificar uma regra com um desempenho satisfatório na maioria dos casos.

Embora se evidencie uma certa vantagem no uso da regra EDD na maioria das instâncias, os resultados não são completamente conclusivos, devido possivelmente à forte componente aleatória introduzida no algoritmo (ver Tabela 1).

Como critério de comparação, foi usada a minimização dos desvios do valor médio relativamente ao óptimo global, para as 75% melhores soluções de cada uma das instâncias. Assim sendo, e da observação da Tabela 1, verifica-se que são obtidos menores desvios do valor médio em relação ao valor óptimo global nas instâncias WT20A, WT20B, WT20C com uma solução inicial gerada pela regra EDD, na instância WT30A com a geração aleatória da solução inicial RND e na WT30B com a regra REDD com a troca de 4 pares.

Como conclusão geral, pode-se assim afirmar que:

- com a solução inicial gerada por ordenação SPT, obteve-se na maioria das instâncias a pior das soluções;
- não se verificou particular vantagem na utilização da regra D_j/W_j;
- uma solução EDD apresenta vantagens em termos de eficácia na maioria dos problemas, podendo-se verificar em alguns casos, melhorias em qualidade com a introdução de alguma perturbação aleatória (regra REDD).

Problema	WT20A	WT20B	WT20C	WT30A	WT30B
Óptimo Global	137	329	425	1911	404
SPT					
melhor valor	171	387	467	1964	534
pior valor	403	707	636	2350	738
valor médio da soluções obtidas	238.5	468	573.1	2123.2	631
valor médio das 75% melhores a)	201.2	429.2	553.1	2072.3	596.8
desvio de a) em relação ao óptimo	46.88%	30.46%	30.14%	8.44%	47.72%
EDD					
melhor valor	147	329	451	1944	469
pior valor	189	656	746	2181	580
valor médio da soluções obtidas	162.3	420.3	532.2	2070.8	516.4
valor médio das 75% melhores a)	154.9	385	491.1	2040.1	497.3
desvio de a) em relação ao óptimo	13.06%	17.02%	15.56%	6.76%	23.1%
D_j/W_j					
melhor valor	147	405	478	1979	446
pior valor	230	559	653	2228	578
valor médio da soluções obtidas	282.3	428.7	527.3	2047.9	562.1
valor médio das 75% melhores a)	202.2	409.3	500.9	2040.1	552
desvio de a) em relação ao óptimo	47.61%	24.42%	17.86%	6.76%	36.63%
RND					
melhor valor	147	411	442	1924	460
pior valor	620	1017	641	2221	624
valor médio da soluções obtidas	262.5	509.75	527.6	2015.8	521.9
valor médio das 75% melhores a)	204.7	450.1	498.2	1984.8	499.3
desvio de a) em relação ao óptimo	49.39%	36.81%	17.23%	3.86%	23.6%
REDD de 4 pares					
melhor valor	149	366	445	1948	445
pior valor	259	606	570	2295	593
valor médio da soluções obtidas	195.2	466.8	510.2	2048.9	500.1
valor médio das 75% melhores a)	177.1	435.3	496.9	1996.3	482.1
desvio de a) em relação ao óptimo	29%	32.3%	16.9%	4.46%	19.3%

Tabela 1 - Influência da regra de construção de uma solução inicial

5. Vizinhanças

Mecanismos geradores de vizinhanças

Em problemas envolvendo permutações, têm sido utilizados vários mecanismos geradores de vizinhanças, nomeadamente a troca de duas tarefas adjacentes, a troca de duas quaisquer tarefas e a troca de três tarefas. De modo a tentar controlar o compromisso eficiência-eficácia, essencial para o bom desempenho prático dos algoritmos, foi neste trabalho considerada uma situação intermédia no caso da troca de duas quaisquer tarefas, e que consiste na troca de duas tarefas cujo afastamento máximo entre si é previamente definido (constituindo, portanto, um parâmetro do algoritmo).

Assim, e para efeitos de comparação, foram testados os dois procedimentos seguintes:

- *troca de pares de tarefas adjacentes*

Dada uma sequência inicial constituída por *n* tarefas, a primeira solução da vizinhança é encontrada trocando a primeira tarefa com a segunda, depois a segunda com a terceira, e assim consecutivamente até, finalmente, se obter a última solução trocando a tarefa *n-1* com a tarefa *n*. Para *n* tarefas, são geradas *N* (dimensão da vizinhança) sequências diferentes, com N = n-1.

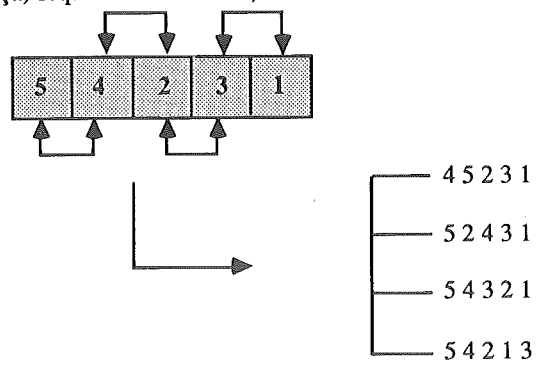


Figura 2 - Troca de pares de tarefas adjacentes

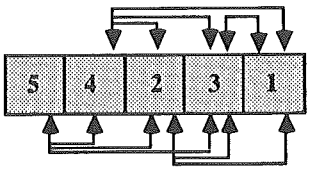
- *troca de pares de tarefas com um afastamento máximo af*

Dada uma sequência inicial constituída por *n* tarefas, começar por trocar a primeira tarefa com as *af* tarefas seguintes, a segunda tarefa com as *af* tarefas seguintes e assim consecutivamente, até se obter a última solução trocando a tarefa *n-1* com a tarefa *n*.

O afastamento é definido em função do tamanho do problema (número de tarefas *n*), fazendo-se *af* = %*n*. Este mecanismo gera *N* (dimensão da vizinhança) sequências diferentes, com:

$$N = (n\text{-}1) + \sum_{i=n-af}^{n-2} i$$

Por exemplo, se partirmos da solução inicial x_1 = [5 4 2 3 1], a vizinhança $N(x_1)$ da sequência x_1 gerada através da troca de duas tarefas quaisquer com um afastamento máximo de 3 (*af* = 60%**n*, com n = 5), é constituída pelas seguintes 9 soluções:



4 5 2 3 1; 5 2 4 3 1; 5 4 3 2 1;
2 4 5 3 1; 5 3 2 4 1; 5 4 1 3 2;
3 4 2 5 1; 5 1 2 3 4; 5 4 2 1 3.

Figura 3 - Troca de duas tarefas com um afastamento máximo 3

Com o objectivo de avaliar a influência do mecanismo de geração de vizinhanças na qualidade da solução final, foram executados alguns testes para as diferentes alternativas, nomeadamente a troca de duas tarefas adjacentes e a troca de duas quaisquer tarefas com um afastamento máximo de 25% da dimensão do problema, considerando subvizinhanças de igual tamanho.

Problema	**WT20A**	**WT20B**	**WT20C**	**WT30A**	**WT30B**
Óptimo Global	**137**	**329**	**425**	**1911**	**404**
Troca de duas adjacentes					
melhor valor	137	386	465	2113	599
pior valor	559	784	1300	2709	1174
valor médio	232.4	498.5	671.5	2259.05	752.3
desvio do valor médio	69.64%	51.52%	58.00%	18.21%	86.21%
Troca de duas quaisquer (afastamento = 25%n)					
melhor valor	147	341	431	1990	452
pior valor	297	660	1072	2207	607
valor médio	178.65	434.8	573.1	2090.85	532.45
desvio do valor médio	**30.40%**	**32.16%**	**34.85%**	**9.41%**	**31.79%**

Tabela 2 - Influência do mecanismo gerador

Através dos vários testes computacionais (ver Tabela 2), verificou-se que o mecanismo de troca de duas quaisquer tarefas (com um afastamento máximo entre si definido em função do tamanho do problema, isto é, do número de tarefas) parece ser o mais eficaz porque:

- tem como resultado uma vizinhança de dimensão superior;
- embora sejam definidas subvizinhanças de igual tamanho, a probabilidade de escapar aos óptimos locais é maior.

Subvizinhanças

A vizinhança total de uma solução é constituída por todas as sequências de tarefas, obtidas por troca de posição de duas tarefas, segundo um mecanismo determinado.

Dada, em geral, a impossibilidade de pesquisar a totalidade das soluções vizinhas, consideram-se, em alternativa, *subvizinhanças* (ou seja, subconjuntos da vizinhança). A dimensão (número de sequências) de uma subvizinhança N' é definida por uma percentagem $N' = \%\ N$, em que N é a dimensão da vizinhança de x_n. A subvizinhança $N'(x_n)$ de x_n é gerada aleatoriamente, escolhendo-se à sorte N' sequências da sua vizinhança $N(x_n)$.

Foram efectuados alguns testes computacionais, tendo-se concluído serem satisfatórias as subvizinhanças constituídas por apenas 25% das soluções vizinhas.

6. Pesquisa Local Aleatorizada

O algoritmo aqui designado por *pesquisa local aleatorizada* (PLA) consiste basicamente na execução repetida do algoritmo de pesquisa local (PL), partindo de soluções iniciais aleatórias. Inicialmente, procurou-se avaliar a influência da componente aleatória associada à regra de

construção da solução inicial no desempenho do algoritmo. Para cada uma das instâncias dos problemas a considerar, foi executado o algoritmo PLA com soluções iniciais geradas, ou duma forma completamente aleatória, ou pela regra REDD descrita anteriormente. Foram feitas as seguintes opções:

1. *Solução inicial* - nos problemas de 20 e 30 tarefas, a perturbação aleatória na regra REDD foi introduzida trocando de posição à sorte 4 pares de tarefas, e nos problemas de 50 e 60 tarefas, trocando 7 pares;
2. *Estrutura de vizinhança* - a vizinhança de uma solução é constituída por todas as sequências resultantes da troca de posição de duas tarefas, cujo afastamento máximo seja de 25% da dimensão do problema. Por exemplo, são trocadas de posição duas tarefas, sempre que o afastamento (número de tarefas que as separa na sequência inicial) seja igual ou inferior a 5, no caso de problemas com 20 tarefas, e de 7, nos problemas com 30 tarefas. A vizinhança resultante é constituída por 85 e 182 soluções, respectivamente. Na prática, foram pesquisadas apenas 25% das soluções, escolhidas aleatoriamente na vizinhança da solução actual;
3. Como critério de paragem do algoritmo, definiu-se o número total de 50 iterações para os problemas de 20 e 30 tarefas, e de 100 iterações para os de 50 e 60 tarefas.

Os resultados computacionais apresentados (Tabela 3) dizem respeito às instâncias já referidas (Sousa e Wolsey [1992]) e às instâncias WT50A e WT60A geradas aleatoriamente (ver secção 5.1). As datas de entrega d_j são uniformemente distribuídas no intervalo $[1, \alpha \sum p_j]$, com $0 \leq \alpha \leq 1$.

Problema	WT20A	WT20B	WT20C	WT30A	WT30B	WT50A	WT60B
Óptimo Global	137	329	425	1911	404	–	–
RND							
melhor valor	137	361	442	1924	431	242	904
pior valor	620	1017	641	2351	657	432	1110
valor médio	218.36	479.7	531.8	2059.6	528	316.1	983.2
desvio do melhor valor	**0%**	9.7%	4%	**0.7%**	6.7%		
desvio do valor médio	59.4%	45.8%	25.1%	7.8%	30.7%	30.6%*	8.8%*
REDD							
melhor valor	137	342	435	1936	415	218	948
pior valor	311	606	813	2295	649	402	1165
valor médio	206.1	431.36	542.1	2038	501.98	277	1064
desvio do melhor valor	**0%**	**4%**	**2.4%**	1.3%	**2.7%**		
desvio do valor médio	50.4%	31.1%	27.6%	6.6%	24.3%	27.1%*	12.2%*

* o desvio é calculado tendo como referência o melhor valor encontrado pelo algoritmo

Tabela 3 - Pesquisa local aleatorizada

Existe uma significativa variabilidade na qualidade dos resultados obtidos, que se poderá ficar a dever à forte componente aleatória introduzida e a alguns comportamentos extremos do algoritmo. No entanto, pela análise dos desvios do melhor valor relativamente ao óptimo, verifica-se que o algoritmo encontra soluções de boa qualidade, isto é, que apresentam valores muito próximos do valor óptimo.

Embora os resultados não sejam completamente conclusivos, a utilização de soluções iniciais não completamente aleatórias, geradas pela regra REDD, poderá reduzir este efeito, permitindo ao algoritmo convergir mais rapidamente para melhores soluções.

Fica a ideia de que, se o algoritmo de pesquisa local for repetido várias vezes, a probabilidade de se alcançar melhores soluções aumenta significativamente, conseguindo-se ainda melhorias em eficácia com a introdução de aleatoriedade na solução inicial (filosofia genérica do algoritmo PLA).

7. Pesquisa Tabu

Com o objectivo de definir os atributos que caracterizam um movimento tabu, e simultaneamente o tamanho da lista, foram realizados alguns testes computacionais em problemas WT (*"weighted tardiness"*). No entanto, verificou-se que dado o considerável grau de aleatoriedade dos algoritmos, se torna difícil tirar conclusões absolutas e definitivas quanto aos valores destes parâmetros, já que os resultados computacionais obtidos dependem da componente aleatória introduzida, quer na construção da solução inicial (solução gerada pela regra REDD descrita anteriormente), quer na definição de subvizinhanças (os elementos são escolhidos à sorte na vizinhança).

Atributos da lista tabu

Na *lista tabu* são armazenados temporariamente um ou mais atributos das soluções recentemente visitadas ou dos movimentos mais recentes, devendo estes atributos ser escolhidos cuidadosamente porque em geral deles depende fortemente o desempenho do algoritmo. São usadas tradicionalmente várias alternativas na definição de atributos; no entanto foram objecto de estudo neste trabalho apenas duas situações: uma em que a lista tabu armazena os pares de tarefas trocadas e outra na qual armazena as tarefas trocadas.

Comparando os resultados obtidos na execução do algoritmo de pesquisa tabu, para as diferentes instâncias dos problemas referidos, poder-se-á concluir que se obtêm geralmente melhores soluções com a primeira destas alternativas (pares de tarefas) do que com a segunda (conjunto das tarefas trocadas).

Tamanho da lista tabu

Procurou-se identificar uma relação de causa-efeito entre a dimensão da lista tabu e a qualidade da solução final. Assim sendo, foram executados testes computacionais para instâncias de problemas com 20, 30, 50, 60, 70, 80 e 100 tarefas, utilizando o algoritmo de pesquisa tabu com listas tabu de tamanho 0, 4, 7 e 10. A consideração da lista de tamanho 0 teve em vista comparar a situação em que o algoritmo considera todas as soluções melhores e piores sem qualquer limitação de movimentos na subvizinhança.

Para a execução do algoritmo de pesquisa tabu foram tomadas as seguintes decisões tendo em vista a respectiva parametrização:

1. *Solução inicial* - a solução inicial é gerada pela regra REDD, cuja perturbação aleatória é introduzida com a troca de 4 pares de tarefas escolhidas à sorte;
2. *Estrutura da vizinhança* - a vizinhança de uma solução é gerada pela troca de posição de duas quaisquer tarefas, com um afastamento máximo entre si de 25% do número de tarefas do problema; para efeitos de pesquisa, considerou-se uma subvizinhança constituída por 25% das soluções da vizinhança;
3. *Estado tabu* - a lista tabu guarda as tarefas ou par de tarefas envolvidas nos últimos movimentos de troca de posição, isto é, o par de tarefas envolvido na modificação que transforma uma solução na solução actual seguinte; por exemplo, se um movimento caracterizado pela troca de posição das tarefas 4 e 2 é armazenado na lista tabu, todos os movimentos seguintes que envolvam o par de tarefas 4 e 2 (ou a ordem inversa, 2 e 4) são considerados tabu;
4. *Tamanho da lista tabu* - o algoritmo foi executado com valores para o tamanho da lista tabu de 0, 4, 7 e 10;
5. *Critério de paragem* - como critério de paragem foi definido o número de 50 iterações (nas instâncias de 20 tarefas), 100 (nas instâncias de 30 tarefas), 200 (nas instâncias de 50 e 60 tarefas) e 300 (nas instâncias de 80 e 100 tarefas).

Partindo da mesma solução inicial (construída pela regra REDD) com a pesquisa realizada em diferentes subvizinhanças (isto é, com diferentes trajectórias, consoante a "corrida" do algoritmo), foram obtidos os resultados coligidos na Tabela 4.

Problema	Tamanho da Lista	0	4	7	10
WT30A	corrida 1	1911	1911	1911	1911
	corrida 2	1915	1915	1912	1915
	corrida 3	1915	1915	1915	1915
	corrida 4	1923	1923	1923	1923
WT50A	corrida 1	205	205	205	206
	corrida 2	208	208	208	206
	corrida 3	216	219	212	220
	corrida 4	206	205	205	210
WT70B	corrida 1	769	762	759	779
	corrida 2	781	759	761	760
WT80A	corrida 1	854	859	853	843
	corrida 2	854	841	844	841
WT100A	corrida 1	1640	1628	1626	1626
	corrida 2	1647	1647	1627	1627

Tabela 4 - Influência do tamanho da lista tabu no desempenho do algoritmo TS

Da análise dos resultados, verificou-se um efeito positivo da lista tabu no desempenho do algoritmo, embora não se tenha tornado claro qual o tamanho da lista a utilizar. Assim sendo, e pela observação da Tabela 4, poder-se-á perceber alguma vantagem na utilização de listas tabu

com tamanho 7 ou próximo. Foi, por isso, este o tamanho escolhido para a lista tabu, nos restantes testes computacionais.

Uma outra possível conclusão de carácter prático, refere-se ao interesse em executar repetidamente o algoritmo de pesquisa tabu (3 ou 4 vezes), com valores diferentes para o tamanho da lista tabu.

8. Avaliação comparativa dos resultados obtidos

Finalmente, foram comparados os desempenhos dos algoritmos de *pesquisa local aleatorizada* (PLA) e de *pesquisa tabu* (TS), na resolução das instâncias atrás referidas para o problema de minimização da soma pesada dos atrasos (os parâmetros utilizados nos algoritmos são os definidos nas secções 6 e 7). Para analisar os resultados obtidos, foram calculadas algumas medidas de desempenho: o melhor valor e o desvio do melhor valor relativamente ao valor óptimo.

Problema	WT20A	WT20B	WT20C	WT30A	WT30B	WT50A	WT60B
Óptimo Global	137	329	425	1911	404	–	–
PLA							
melhor valor	137	342	435	1936	415	218	948
desvio do melhor valor	0%	3.95%	2.35%	1.3%	2.27%		
TS							
melhor valor	137	329	425	1911	427	205	873
desvio do melhor valor	0%	0%	0%	0%	5.69%		

Tabela 5 - Estudo comparativo do desempenho das meta-heurísticas

A comparação realizada, em termos de eficácia (ou seja da qualidade das soluções), assenta numa parametrização dos dois algoritmos com a qual se procurou que o esforço computacional envolvido fosse idêntico.

Ambos os algoritmos obtiveram um bom desempenho, podendo-se, apesar de tudo, verificar vantagens na utilização do algoritmo de *pesquisa tabu* na resolução da maioria das instâncias, tendo mesmo sido encontrados os óptimos globais nalgumas das instâncias. O algoritmo PLA encontrou uma solução de melhor qualidade na instância WT30B. As diferenças encontradas são, contudo, pouco significativas (ver Tabela 5).

Apesar de tudo, poder-se-á considerar os seguintes aspectos como positivos na comparação do algoritmo PLA com o procedimento de *pesquisa tabu*:

- a possibilidade de obtenção de soluções de boa qualidade (próximas do valor óptimo e das soluções obtidas pelo algoritmo de pesquisa tabu), mais rapidamente e com um menor esforço computacional;
- é de mais fácil implementação;
- não necessita de um grande esforço de afinação de parâmetros.

Do conjunto de testes realizados, parece ficar claro o interesse mais global em construir "meta-algoritmos", baseados na repetição dos algoritmos originais (isto é, em corrê-los várias

vezes), possibilitando um melhoramento progressivo das soluções, na maioria das vezes com um esforço computacional pouco importante.

9. Conclusões

Mais do que desenvolver algoritmos de utilidade prática indiscutível, constituía um dos objectivos principais deste trabalho ilustrar, em problemas de sequenciamento simples, as potencialidades das abordagens meta-heurísticas. Os exemplos apresentados e os resultados computacionais obtidos, embora não sendo completamente conclusivos, mostram que neste tipo de problemas (onde existem estruturas de vizinhança bastante naturais - as trocas de duas tarefas adjacentes e de duas quaisquer com um afastamento máximo), estes procedimentos conduzem, em geral, a soluções satisfatórias, de uma forma eficiente. Parece também claro que se trata de procedimentos robustos, facilmente adaptáveis a diferentes problemas de escalonamento.

Por outro lado, um outro aspecto positivo deste trabalho foi o ter permitido estruturar uma abordagem geral, ainda que simples, para a concepção e avaliação de meta-heurísticas e que se julga poderá vir a ser usada em muitos outros problemas.

Os resultados obtidos, quando observados globalmente, mostram o interesse de, usando algoritmos de base com uma forte componente aleatória, construir "meta-procedimentos" que consistam em repetir esses algoritmos tantas vezes quantas o tempo de computação disponível o permita.

Finalmente, um possível desenvolvimento deste trabalho poderá consistir no tratamento de outras funções objectivo (que incluam por exemplo custos e tempos de setup) ou de outras restrições (como por exemplo, restrições de precedência entre tarefas). Trata-se de extensões em princípio de fácil implementação com as abordagens apresentadas neste artigo. Em particular, estas extensões permitirão construir procedimentos de análise multi-critério, a integrar possivelmente num Sistema de Apoio à Decisão, onde a interface com o utilizador poderá também desempenhar um papel importante.

Referências

[1] Abdul-Razaq, T.S., C.N. Potts e L.N. Van Wassenhove, A survey for the Single-Machine Scheduling Total Weighted Tardiness Scheduling Problem, Discrete Applied Mathematics 26 (1990) 235-253.
[2] Baker, K.R., Introduction to Sequencing and Scheduling, Wiley, New York (1974).
[3] French, S., Sequencing and Scheduling: An introduction to the Mathematics of the Job-Shop, Ellis Horwood, Chichester (1982).
[4] Lawler, E.L., Recent Results in the Theory of Machine Scheduling, Mathematical Programing: The State of the Art, edited by A.Bachem et al, Springer Verlag, (1983) 203-234.
[5] Feo, T. A., M. G. C. Resende e S. H. Smith, A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set, first draft (1989).
[6] Pirlot, M., General Local Search Heuristics in Combinatorial Optimization: a Tutorial, JORBEL 32 (1992) 7-68, Bruxelas.
[7] Sousa, J.P. e L.A. Wolsey, A time indexed formulation of non-preemptive single-machine scheduling problems, Mathematical Programming 54 (1992) 353-367.

INVESTIGAÇÃO OPERACIONAL, vol. 16 (Dezembro 1996) 135-154

UM SISTEMA DE APOIO À DECISÃO PARA O SEQUENCIAMENTO DA PRODUÇÃO NA INDÚSTRIA GRÁFICA

Manuel Luís Machado
Imprensa Nacional - Casa da Moeda

Rui Carvalho Oliveira
CESUR
Dep. Engª Civil - IST
Av. Rovisco Pais
1000 Lisboa - Portugal

Abstract

In this paper it is described a Decision Support System (DSS) designed to support the scheduling of operations in the printing industry modelled as a job-shop problem.

The paper is focused on the development of heuristics whose main objectives are, on one hand, to minimise the number of tardy jobs, their lateness in respect to due dates, as well as the mean job flow time and, on the other hand, they seek to maximise the utilisation of the work centres. The role of these heuristics in the DSS consists in the generation of initial schedules (that may then be improved in interactive sessions) and/or in the automatic completion of schedules partly defined by the planning agent. This DSS also presents the possibility of interaction with the planning agent, allowing him to impose different processing sequences in one or more centres identified as critical (bottlenecks) with the aim to increase the global performance of the final schedule having in mind several conflicting objectives. The heuristics determine the operations scheduling on the remaining work centres in an automatic way.

Experiences in an industrial environment (a graphic company) have been carried out and the results are considered good in respect to the most important objectives for the company.

Resumo

O trabalho apresentado nesta comunicação teve como objectivo o desenvolvimento de um Sistema de Apoio à Decisão (SAD) para o sequenciamento de operações em ambiente industrial de *job-shop*, na indústria gráfica.

O trabalho centrou-se essencialmente no desenvolvimento de heurísticas que procuram, por um lado, minimizar o número de ordens de fabrico atrasadas e os respectivos desvios relativamente às datas devidas de entrega bem como os tempos de permanência na oficina e, por outro, maximizar as taxas de utilização dos centros de trabalho. O papel destas heurísticas no SAD consiste na geração de soluções iniciais (que possam depois ser melhoradas em sessões interactivas) e/ou no completar, por via automática, de soluções parcialmente definidas pela agente de planeamento. O sistema permite aionda que este agente imponha diferentes sequências de processamento em um ou mais centros identificados como "críticos" (pontos de estrangulamento), com vista ao aumento da *performance* global da solução final, tendo em atenção os vários objectivos conflituosos. As heurísticas determinam a programação das operações nos restantes centros de trabalho de uma forma automática.

Foram efectuadas experiências em ambiente industrial, numa empresa gráfica, considerando-se bastante satisfatórios os resultados obtidos pela aplicação das heurísticas, tendo em atenção os objectivos com maior importância para a empresa.

Keywords

Decision Support Systems, Heuristics, Job-Shop, Printing Industry.

1. Introdução

O cumprimento das datas de entrega é cada vez mais um facto de competitividade para as empresas do sector gráfico com produção por encomenda, sendo crucial para assegurar vendas futuras. A par deste objectivo, as empresas devem apresentar bom desempenho em termos da resposta rápida às solicitações da procura (ou curtos prazos de entrega) e da flexibilidade para acomodar alterações resultantes tanto de avarias do equipamento como de solicitações dos clientes, que reflectem um objectivo global de maximização do nível de satisfação do cliente, aspecto vital na estratégia das empresas.

A não satisfação do cliente conduz à perda de confiança, em relação à qualidade do serviço da empresa e pode implicar a redução no volume das potenciais encomendas futuras do cliente não satisfeito e de outros que tomem conhecimento do sucedido. Este tipo de situações apresenta custos intangíveis de muito difícil avaliação visto não ser possível estimar os valores futuros da procura perdida como resultado da "má imagem" da empresa. Por outro lado e com certo grau de conflituosidade, apresentam-se objectivos relacionados com a estabilidade dos planos de fabrico, a utilização eficiente dos recursos disponíveis ou a minimização dos tempos de permanência das ordens de fabrico na oficina.

O planeamento da produção na empresa gráfica para a qual se desenvolveu este trabalho apresenta as características típicas de um problema de *job-shop*, como se descreve na secção 2. O problema do *job-shop* é de resolução reconhecidamente complexa dada a sua natureza combinatória, pelo que o recurso a algoritmos de optimização para a sua resolução é bastante limitado, levando a que seja praticamente inevitável a utilização dos métodos heurísticos na resolução de problemas de dimensão industrial [Baker, 1974; French, 1982; Lawler et al., 1982]. Mesmo os melhores algoritmos de optimização, como os de McMahon e Florian [1975] e de Carlier e Pinson [1988], tipicamente utilizando técnicas de *branch-and-bound*, apenas permitem resolver problemas de dimensão modesta, pelo que é habitual recorrer a heurísticas com graus de compexidade e sofisticação muito variadas.

As heurísticas "Shifting Bottleneck" de Adams et al [1988] e PRO (Partial Resource Optimization) de Marques [1993] são exemplos de métodos heurísticos relativamente sofisticados com bons resultados na resolução do problema em ambiente industrial. Outra das abordagens do problema, com menor complexidade, é a simulação interactiva [Hurrion, 1978], através da qual se pode ensaiar e avaliar toda a panóplia de regras heurísticas de sequenciamento disponíveis.

A aplicação das meta-heurísticas a problemas de sequenciamento da produção é muito recente, devendo-se o sucesso destes métodos a vários factores, nomeadamente, a aplicabilidade a problema diversos, a excelente qualidade das soluções, a fácil implementação e a flexibilidade na adaptação a casos reais [Barnes et al., 1995; Brown et al., 1995].

O presente trabalho descreve um Sistema de Apoio à Decisão (SAD) no sequenciamento da produção (ambiente do tipo *job-shop*), desenvolvido para aplicação numa empresa gráfica. Esta abordagem justifica-se na medida em que não está disponível um estratégia que permita descrever como construir uma solução global aceitável e os objectivos e restrições presentes no processo de tomada de decisão são conflituosos, de problemática formalização, e difíceis de avaliar. Os instrumentos de apoio para validação, criação ou melhoramento de soluções baseiam-se em heurísticas de afectação progressiva iterativa e de compactação das operações que se apresentam na secção 3.

2. Descrição do problema

2.1 Processo produtivo

A indústria de impressão apresenta três áreas principais, as publicações impressas, as embalagens e os trabalhos comerciais (onde estão englobados os restantes grupos de trabalhos gráficos). A empresa gráfica deste estudo insere-se no último grupo, predominado a produção do tipo por encomenda de uma ampla gama de produtos em que o papel é o suporte de impressão. De uma forma genérica, os centros de trabalho estão agrupados em três sectores distintos inerentes ao sistema produtivo (Figura 1), onde são executadas as operações relativas aos processos de pré-impressão, impressão e acabamento.

Embora o número de rotas diferentes na oficina seja elevado, o fluxo das operações segue um determinado padrão, visto que qualquer uma das rotas de fabrico tem operações (pelo menos uma) cujo processamento é efectuado em cada um dos sectores supra mencionados.A produção tem duas vertentes, para *stock* e por encomenda. Existe uma gama de produtos normalizados cujo consumo é efectuado por uma variedade de clientes. A empresa possui geralmente *stocks* destes produtos de forma a satisfazer de imediato as encomendas, pois existem compromissos estabelecidos a longo prazo com vista ao respectivo fornecimento. A ocorrência de rotura de *stocks* terá influências negativas na negociação para futuros fornecimentos, podendo mesmo motivar a perda de clientes a longo prazo.

Nos produtos cujo fabrico é por encomenda, em que a concorrência é extremamente forte, o prazo de entrega ou a rapidez de resposta às solicitações do cliente é, com alguma frequência, o factor responsável pela perda de encomendas e, nas situações em que não é respeitado, conduz à perda de confiança em relação à qualidade do serviço da empresa. Este tipo de produção representa cerca de 80% do volume total (em termos de ocupação dos centros).

2.2 Processo de planeamento uitlizado na empresa

Este caso de estudo envolve o sequenciamento da produção de um conjunto de ordens de fabrico, cujo número varia normalmente entre 100 e 150, abrangendo 32 centros de trabalho (todos com capacidade finita) e cobrindo um horizonte temporal de um mês. Os módulos de produção (ordens de fabrico firmadas) têm a duração semanal.

A revisão do plano de fabrico é efectuada diariamente (uma a duas vezes), coincidindo com a entrada de novas encomendas com carácter relativamente urgente ou motivada pela avaria de um equipamento cuja paragem apresente duração significativa.

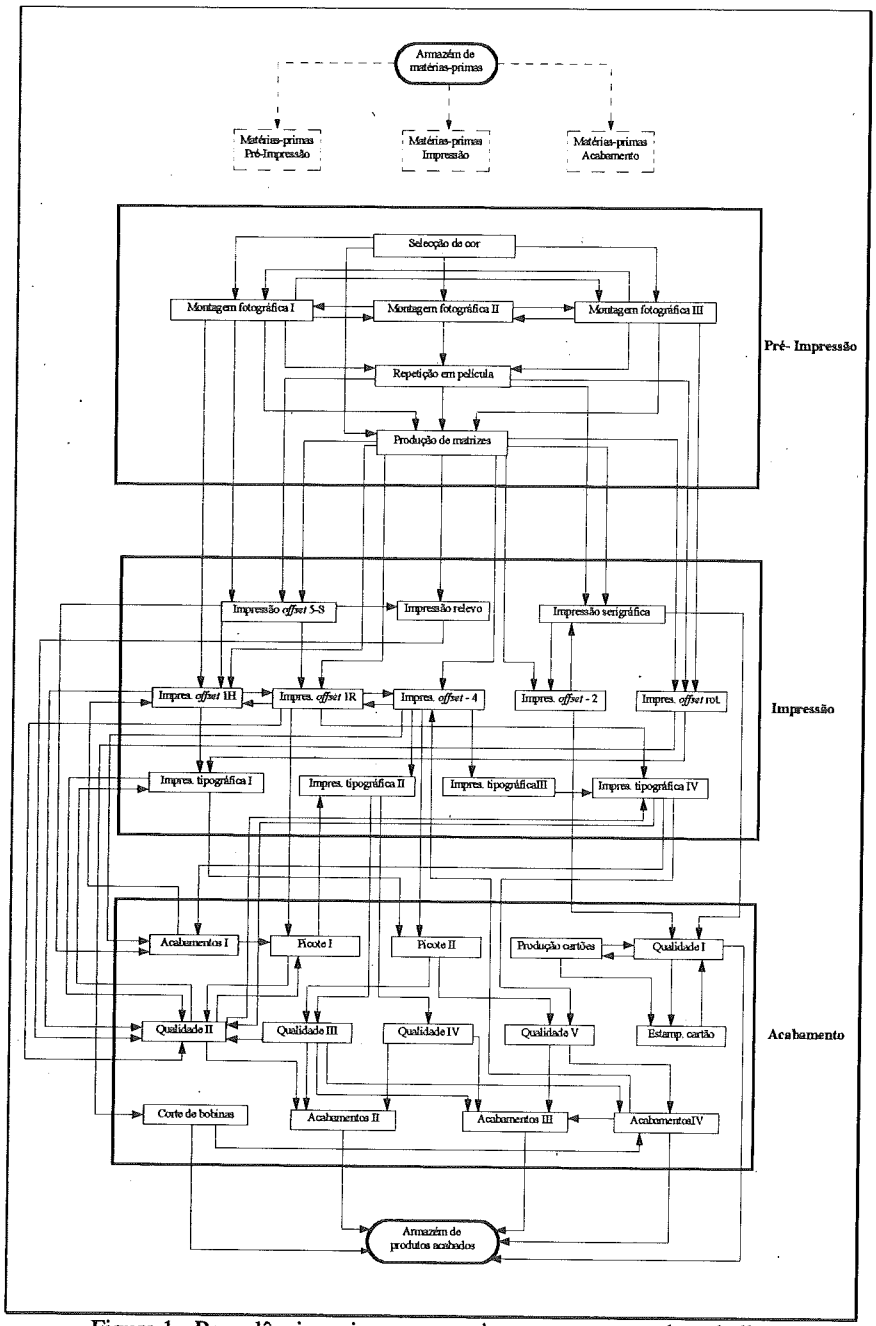


Figura 1 - Precedência mais representativas entre centros de trabalho

Actualmente, a elaboração do plano de fabrico é efectuada de forma manual, por alguém com bastante experiência e que procura essencialmente minimizar os atrasos (e de preferência evitá-los) relativamente aos prazos de entrega. Devido ao facto desta tarefa ser bastante morosa, a preocupação fundamental dos agentes de planeamento é a obtenção de um plano de fabrico exequível, sendo adoptado como definitivo o primeiro plano que for gerado. Por outro lado, estes agentes de decisão têm alguma dificuldade em definir de uma forma explícita os objectivos a atingir, dado apresentarem uma certa conflituosidade, como o cumprimento dos prazos de entrega ao cliente, a maximização da utilização dos centros de trabalho, a resposta rápida às solicitações dos clientes e a minimização dos *stocks* de produtos em vias-de-fabrico.

Devido à falta de um apoio informático adequado, os agentes de planeamento consideram que as soluções que se conseguem obter são em geral bastante insatisfatórias devido à percentagem de ordens de fabrico que são concluídas com atraso, ao elevado tempo de permanência na oficina dos produtos em vias-de-fabrico (consequentemente, com um significativo tempo de espera nos centros de trabalho) e à utilização de alguns centros de trabalho que funcionam por vezes como um estrangulamento das ordens de fabrico. Actualmente não é efectuada a avaliação do plano obtido (não são calculadas quaisquer medidas de desempenho), pois a sua utilidade seria muito reduzida visto que normalmente não são elaborados planos alternativos para comparação.

A cada ordem de fabrico é atribuído um índice de prioridade, pelo agente comercial, que reflecte o grau de importância associado à respectiva encomenda, com vista ao cumprimento do respectivo prazo de entrega. Esse índice é função do tipo de produção e está relacionado de uma forma muito significativa com o tipo de produto e mesmo com o respectivo cliente, nos produtos cujo fabrico é por encomenda. O índice é um valor de 1, 2 ou 3, consoante o grau (crescente) de importância associado à respectiva encomenda.

Na programação das operações fabris é elaborado um mapa de *Gantt* que é construído num quadro de parede representando o plano de fabrico, utilizando o seguinte procedimento:

(i) Seleccionar entre as ordens de fabrico a programar as que têm índice de prioridade mais elevado, sendo a ordenação (para um mesmo índice de prioridade) definida pelo valor crescente da folga (diferença entre o intervalo de tempo até à data devida e a duração das operações por realizar). Consideram-se todas as ordens de fabrico, incluindo aquelas em que já foi iniciado o processamento de uma ou mais operações.

(ii) Afectar as operações relativas a essas ordens de fabrico de uma forma progressiva, a partir da primeira operação, respeitando tanto os requisitos tecnológicos inerentes aos fabrico do produto como as restrições de capacidade limitada dos centros de trabalho.

(iii) Repetir o procedimento referido em (i) e (ii) para as restantes ordens de fabrico, segundo a ordem decrescente do respectivo índice de prioridade.

A programação das operações fabris é uma tarefa complexa, devido nomeadamente à existência de um número elevado de ordens de fabrico a programar, ao significativo número de operações por ordem de fabrico, às ordens de fabrico com rotas de fabrico bastante variadas, às rotas alternativas para alguns produtos, podendo certas operações ser processadas em mais do que um centro de trabalho, à possibilidade de voltar atrás na rota para correcção de anomalia detectada em fase posterior de fabricação, aos compassos de espera na cadeia produtiva provocados pela necessidade de aprovação de prova por parte do cliente, à necessidade de efectuar experiências de produção devido às características específicas do produto e à possibilidade de uma ordem de fabrico ter mais do que uma operação processada num determinado centro de trabalho.

Os agentes de planeamento e os responsáveis pela empresa têm a noção de que a utilização de uma "ferramenta" informática no processo de planeamento e controlo da produção poderá conduzir a um aumento considerável de produtividade e flexibilidade. Essa "ferramenta" deve permitir a utilização de técnicas adequadas de resolução dos vários problemas, a geração e comparação de planos alternativos, o que é actualmente impossível realizar em tempo útil, assim como uma maior coordenação das diferentes tarefas interactuantes no planeamento. Deste modo, foi adquirido um *package* informático com vista à gestão informatizada da produção, em que a arquitectura de desenvolvimento assenta na filosofia MRP II dos sistemas produtivos do tipo *flow-shop*. As principais dificuldades na implementação do sistema informático têm residido no sequenciamento da produção, visto que o sistema apresenta pouca flexibilidade para se adaptar aos requisitos da programação da produção.

A proposta deste trabalho é desenvolver um módulo de planeamento/sequenciamento da produção em ambiente *job-shop* de capacidade finita, adaptado à indústria gráfica com vista a ser integrado no *package* informático que está a ser implementado na empresa. Este trabalho foi motivado pela necessidade de desenvolver técnicas de sequenciamento com um elevado nível de aceitação pelo utilizador e que possam ser aplicadas num ambiente industrial com as características atrás descritas.

3. Procedimentos de sequenciamento na indústria gráfica

A inexistência de métodos de optimização eficientes para a resolução do problema do *job-shop* de dimensão industrial leva a que seja praticamente inevitável a utilização de métodos heurísticos na resolução deste tipo de problemas. No problema em estudo a complexidade é ainda maior devido à existência de vários objectivos parciais no processo de planeamento, que apresentam um certo grau de conflituosidade.

Neste contexto, uma abordagem relativamente recente aos problemas de programação de operações fabris passa pelo desenvolvimento e utilização de SAD's. Estes sistemas permitem a incorporação do conhecimento e experiência do agente de planeamento na elaboração do plano de fabrico, sendo permitida a definição da sequência de processamento de operações em determinadas máquinas, deixando ao sistema a definição da sequência de operações nas

restantes máquinas através da utilização de heurísticas que conduzem a soluções satisfatórias e são eficientes computacionalmente. Esta abordagem parece ser a mais adequada para a maioria dos problemas industriais.

3.1 Heurística de afectação progressiva

No que respeita às heurísticas e devido ao tipo de problema em estudo, será utilizado o método de sequenciamento com base na data devida de entrega, recorrendo a uma heurística designada de Afectação Progressiva Iterativa (API) para a geração de soluções iniciais (que possam depois ser melhoradas em sessões interactivas) e/ou no completar, por via automática, soluções parcialmente definidas pelo agente de planeamento. De uma forma genérica, a heurística utiliza o sequenciamento inicial exequível gerado pela heurística de Afectação Progressiva (AP), da primeira até à última operação a partir da data mínima de lançamento na produção, como uma solução inicial. De seguida, utilizando um processo iterativo, as ordens de fabrico que estão concluídas antes ou depois do respectivo prazo serão deslocadas no plano de forma que seja minimizado o desvio em relação à respectiva data devida de conclusão [White, 1985]. O ordem de afectação das ordens de fabrico aos centros baseia-se no índice de prioridade e no quociente "folga/número de operações".

O sequenciamento progressivo efectua a afectação de cada operação tão cedo quanto possível, estando normalmente concluída antes que seja requerida no centro subsequente da rota de fabrico, o que poderá promover a acumulação de *stocks* de produtos em vias-de-fabrico. No caso do sequenciamento regressivo (a operação deve apenas estar concluída quando for requerida no centro subsequente) são minimizados os inventários dos produtos em vias-de-fabrico, mas qualquer desrespeito no cumprimento da conclusão do processamento de uma operação na data devida poder-se-á traduzir no não cumprimento dos prazos devidos de entrega. O sequenciamento regressivo foi igualmente rejeitado por razões relacionadas com a possibilidade de geração de planos não exequíveis, o que não acontece quando se utilizam métodos progressivos, nomeadamente pode acontecer que o processamento das operações se inicie antes da data actual com vista ao cumprimento das datas devidas, o que poderia comprometer seriamente a aceitação do sistema pelo utilizador, visto que o obrigaria a efectuar diversas alterações apenas para que o plano de fabrico se tornasse exequível.

A principal limitação da heurística de Afectação Progressiva (AP) poderá ser o facto da conflituosidade entre a importância de uma ordem de fabrico (índice de prioridade) e a respectiva data devida de conclusão originar que o processamento de algumas ordens de fabrico termine inutilmente antes da data devida, podendo no entanto originar que outras ordens, com posicionamento relativamente próximo na ordenação de sequenciamento, sejam concluídas tardiamente. Com vista à melhoria da qualidade dos planos de fabrico resultantes apenas da heurística AP (redução dos custos associados aos produtos acabados e em vias-de-fabrico, bem como o cumprimento dos prazos de entrega) foi desenvolvida a heurística API.

3.2 Heurística de afectação progressiva iterativa

A heurística desenvolvida tem três fases, a primeira das quais é a construção de um plano de fabrico exequível utilizando os princípios já esboçados. A segunda fase compreende a selecção das ordens de fabrico cuja conclusão esteja estimada para antes da respectiva data devida, sendo consideradas as ordens de fabrico na ordem inversa em relação à sequência de afectação. O intuito desta fase é efectuar o reescalonamento destas ordens de fabrico abrindo, assim, intervalos de tempo no plano que poderão, então, ser utilizados pelas ordens de fabrico que estavam previamente escalonadas para terminarem para além da respectiva data devida.

Uma vez analisadas durante a segunda fase todas as ordens de fabrico que terminam adiantadas, a terceira fase envolve a pesquisa e ordens escalonadas para terminarem atrasadas. Nesta fase, as operações da ordem de fabrico atrasada são desescalonadas e reescalonadas utilizando a heurística AP. A segunda e terceira fases são repetidas até não ocorrer movimento de ordens de fabrico, sendo quatro ciclos destes considerado um limite razoável, em termos computacionais.

Prioridade	Ordem de Fabrico	Data mínima Dia	Data mínima Hora	Data devida Dia	Data devida Hora	Operação	Centro de trabalho	Duração (horas)
1ª	A	1	0:00	6	2:00	A1	δ	6
						A2	α	4
						A3	μ	4
						A4	β	2
						A5	φ	4
2ª	B	1	0:00	3	8:00	B1	δ	2
						B2	μ	4
						B3	β	6
						B4	δ	2
						B5	φ	4
3ª	C	1	0:00	4	2:00	C1	δ	4
						C2	α	6
						C3	β	4
						C4	α	2
4ª	D	1	0:00	4	6:00	D1	β	6
						D2	α	4
						D3	μ	4
						D4	φ	2
						D5	δ	2
5ª	E	1	0:00	6	4:00	E1	φ	6
						E2	μ	2
						E3	δ	4
						E4	α	4

Tabela 1 - Lista das ordens de fabrico a processar

As diferentes fases da heurística API são ilustradas com um pequeno exemplo, sendo complementada a descrição dos procedimentos de cada uma das fases com as alterações ocorridas no exemplo apresentado. A lista de ordens de fabrico a processar é apresentada na Tabela 1, com a definição da ordem de afectação às máquinas, que são cinco e estão disponíveis durante oito horas por dia, com um único processador. O plano inicial de fabrico resultante da aplicação da heurística AP é apresentado na Figura 2, verificando-se que as ordens de fabrico A e E são concluídas com adiantamento relativamente à data devida, enquanto que as restantes ordens de fabrico são concluídas com atraso.

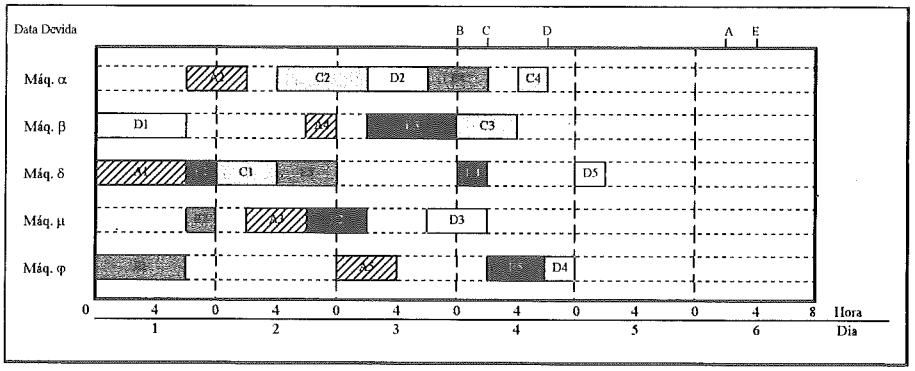


Figura 2 Plano de fabrico gerado pela aplicação da primeira fase da heurística API.

Na segunda fase, após o desescalonamento de uma ordem, não ocorre qualquer alteração em todas as restantes ordens, no que respeita aos instantes de início das operações. O escalonamento é efectuado de forma a utilizar os intervalos de tempo de inactividade dos centros de trabalho cujo cumprimento seja superior ou igual à duração total das respectiva operação, desde que respeitadas as restrições de precedência entre operações. Para cada ordem de fabrico adiantada devem ser efectuados os seguintes procedimentos:

Passo 1 - Deescalonar as operações da ordem de fabrico.

Passo 2 - Incrementar o início do processamento da ordem de fabrico do valor do intervalo de tempo definido entre os instantes previsto e devido de conclusão.

Passo 3 - Reescalonar a ordem de fabrico a partir da nova data de início de processamento.

Passo 4 - Se a data prevista de conclusão da ordem de fabrico não avançar, passar à ordem seguinte.

Passo 5 - Caso, agora, a data prevista de conclusão seja posterior à data devida, sendo o valor do intervalo de tempo do adiantamento da ordem superior a um dia, após a redução para metade do tempo de avanço, seguir para o passo (2). A pesquisa dos intervalos de tempo de inactividade dos centros será apenas efectuada nos que estão compreendidos entre a data inicial do início de processamento e a duração do primeiro incremento do valor do tempo de adiantamento.

A aplicação da segunda fase da heurística API, no exemplo apresentado, provoca o movimento das ordens de fabrico A e E, provocando a redução do desvio entre a data prevista e a data de conclusão do respectivo processamento (Figura 3).

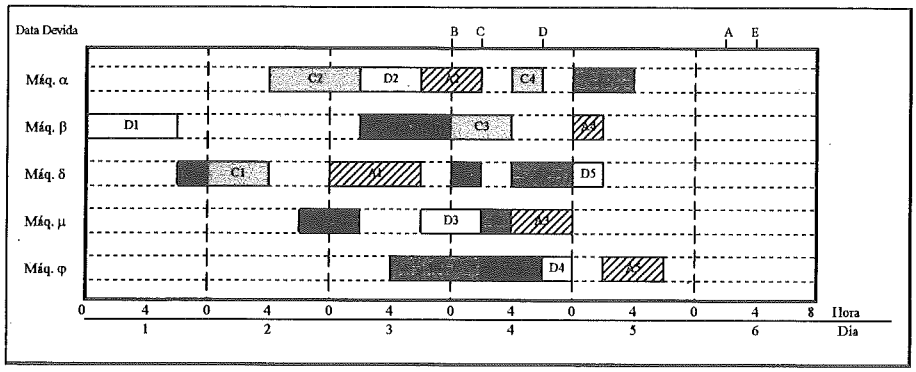


Figura 3 Plano de fabrico gerado pela aplicação da segunda fase da heurística API.

Na terceira fase, para cada ordem de fabrico atrasada, seguindo a ordem inicial de sequenciamento, devem ser efectuados os seguintes procedimentos:

Passo 1 - Desescalonar as operações da ordem de fabrico.

Passo 2 - Reescalonar a ordem de fabrico utilizando a heurística AP.

A aplicação da terceira fase da heurística API, no exemplo apresentado, provoca o movimento das ordens de fabrico atrasadas após a primeira fase (as ordens de fabrico B, C e D), originando que após esta fase apenas a ordem D é concluída com atraso (Figura 4).

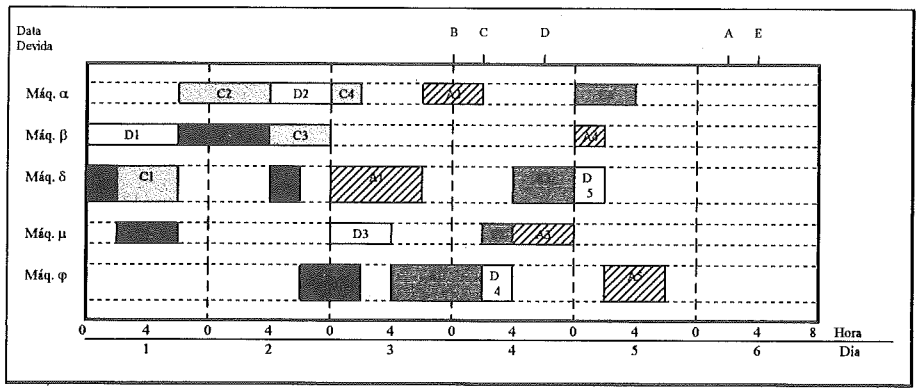


Figura 4 Plano de fabrico gerado pela aplicação da terceira fase da heurística API.

Com a aplicação dos restantes três ciclos iterativos (segunda e terceira fases da heurística API) verifica-se que todas as ordens de fabrico são concluídas sem atrasos (Figura 5).

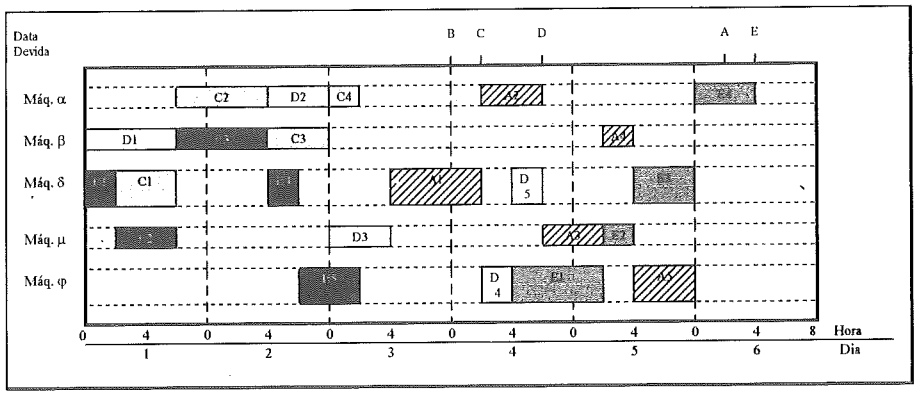


Figura 5 Plano de fabrico gerado após dois ciclos iterativos da heurística API.

3.3 Heurística de compactação

Após a aplicação das segunda e terceira fases da heurística API, verifica-se que normalmente são gerados vários intervalos de tempo (alguns mesmo relativamente pequenos), em que os centros de trabalho se encontram inactivos, reflectindo-se em valores inferiores das taxas de utilização de alguns centros de trabalho, quando comparados com os valores obtidos após a aplicação apenas da primeira fase da heurística. Esta alteração na *performance* do sistema produtivo é perfeitamente justificável, visto que o procedimento utilizado dá ênfase primordial ao cumprimento dos prazos de entrega, podendo eventualmente sacrificar outras medidas de desempenho.

Com o objectivo de atenuar este efeito menos desejável e aumentar a flexibilidade do plano para acomodar novas ordens, após a aplicação da heurística API poder-se-á aplicar a **heurística de compactação das operações** nos centros de trabalho (COMPACT), baseada no princípio de iniciar o processamento das operações tão cedo quanto possível (respeitando todas as restrições), sendo gerado um plano de fabrico semi-activo.

4. Sistema de apoio à decisão

Através dos chamados Sistemas de Apoio à Decisão (SAD) procura-se que o agente de decisão tenha uma intervenção activa na criação de soluções alternativas, incorporando na resolução dos problemas o seu conhecimento e experiência, fornecendo-lhe o sistema os elementos necessários à respectiva avaliação e instrumentos de apoio (nomeadamente, modelos analíticos) para a validação, criação ou melhoramento de soluções.

A tradução dos objectivos do sequenciamento da produção na forma operacional da actividade do planeamento é muito difícil, visto que alguns destes objectivos são contraditórios devendo ser mantida alguma forma de compromisso. Contudo, as relações de troca (*trade-off*) entre as medidas de desempenho associadas aos objectivos não podem ser explicitamente definidas, e este compromisso é um assunto de julgamento subjectivo que varia com o agente de

planeamento e o respectivo contexto. Esta actividade de planeamento pode ser classificada como um processo de decisão semi-estruturada, segundo a definição proposta por Keen e Scott Morton [1978].

Os agentes de planeamento participaram na construção do sistema, tendo-lhes sido solicitado pareceres ao longo de todo o processo. Fez-se sentir que as suas opiniões seriam tidas em consideração e foram-lhes explicadas as razões de certas opções tomadas, de forma a que se desencadeiem as mínimas resistências inerentes à mudança que poderiam favorecer uma atitude de boicote ao sistema. Contudo, as justificações baseadas numa racionalidade completa e irrefutável podem não agradar àqueles agentes de decisão cuja autoridade repousa fundamentalmente no carisma ou no poder formal. Têm sido criadas as condições para que a introdução do SAD constitua um momento particularmente motivante para os elementos envolvidos.

O sistema, implementado em ambiente MS-DOS e codificado em linguagem "C" (compilador "C", versão 7.0, Microsoft®), apresenta uma estrutura modular, integrando os componentes clássicos de um SAD [Sprague et al., 1993; Speranza et al., 1991]. O SAD é composto por uma base de dados, uma base de modelos e uma interface que permite a ligação do utilizador a cada um deles. A informação *input* ao sistema baseia-se fundamentalmente em ficheiros que contêm informação relativa à caracterização das ordens de fabrico a executar, das rotas de fabrico e da capacidade disponível nos centros de trabalho. A base de modelos é composta por vários módulos cujas funções permitem: a ordenação das ordens de fabrico segundo o critério definido; a elaboração do plano de fabrico, segundo a heurística seleccionada pelo utilizador; a avaliação do plano de fabrico através da determinação das medidas de desempenho seleccionadas e a edição do plano de fabrico com vista às alterações da posição relativa das operações processadas num centro ou da rota de fabrico.

Relativamente à possibilidade do utilizador poder alterar a posição relativa de duas operações, o sistema deverá apresentar as consequências dessas alterações, não só nesse centro, mas em todos os situados a jusante do processo das diferentes ordens de fabrico interessadas, nomeadamente através de alertas para situações que requeiram intervenções.

A interface é extremamente importante no sucesso do SAD, pois é através da estrutura de diálogos que se processa a comunicação entre o utilizador e o sistema, sendo o respectivo desenvolvimento uma das preocupações principais de qualquer SAD [Kee e Scott Morton 1978; Moreira e Oliveira, 1991].

A interface com o utilizador foi desenvolvida com vista a satisfazer diversos objectivos: tornar atractiva a utilização do sistema, apresentar a informação relevante em cada contexto de forma simples e perceptível, permitir ao utilizador comandar o processo impondo decisões no que respeita à definição das sequências de processamento em alguns centros de trabalho e apresentar sugestões ou alertas para situações que requeiram intervenções.

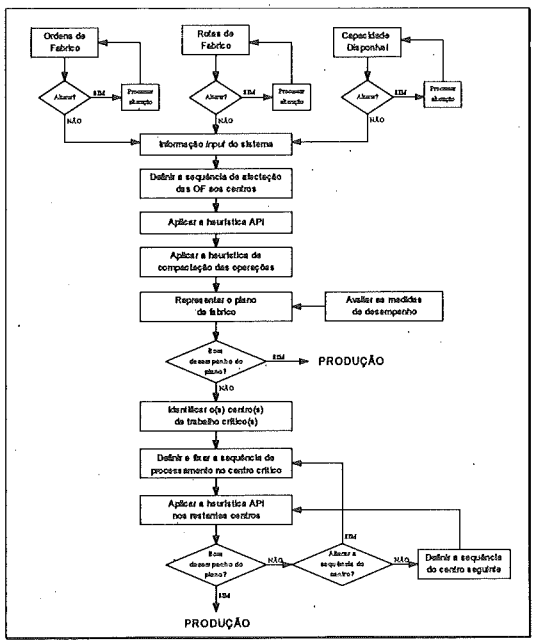


Figura 6 - Esquema de utilização do SAD no sequenciamento de operações

A forma de funcionamento do SAD na elaboração do plano de fabrico baseia-se em vários módulos sequenciais (Figura 6) relacionados com a definição do conjunto de ordens de fabrico a executar, a definição das gamas da fabrico, a definição da capacidade disponível dos centros de trabalho, a aplicação das heurísticas, a avaliação qualitativa dos planos de fabrico e a representação do plano de fabrico (formas numérica e gráfica).

O sistema apresenta ainda a possibilidade de interacção com o agente de planeamento, permitindo que se imponha (experimente) diferentes sequências de processamento em um ou mais centros de identificados como "críticos" (pontos de estrangulamento), com vista ao aumento da *performace* global da solução final. Nestes centros de trabalho, o sequenciamento abedece muitas vezes a regras muito específicas de cada processo de fabrico e de natureza pouco estruturada. A programação das operações nos restantes centros pode ser efectuada pela heurísticas de uma forma automática. Desta forma, o utilizador concentra os seus esforços na definição do sequenciamento dos recursos "críticos", deixando para a heurística a tarefa de sequenciar de forma automática todos os restantes centros de trabalho. O conjunto de centros de travalho críticos varia com o decorrer do tempo, devendo ser efectuada a respectiva identificação através dos resultados obtidos pela aplicação das heurísticas acima mencionadas.

Como o índice de prioridade tem extrema importância na ordem de afectação das ordens de fabrico aos centros de trabalho, o agente de planeamento deverá ter a possibilidade de alterar o índice de prioridade de uma dada ordem de fabrico se identificar que essa ordem, num determinado contexto e devido ao valor do índice associado, provoca uma redução global da *performance* do plano de fabrico e que apenas com essa alteração esse efeito será eliminado.

A avaliação dos planos de fabrico permite comparar planos diferentes, obter indicações sobre as alterações mais indicadas com vista a satisfazer determinado objectivo e analisar as alterações que o utilizador vai efectuando, permitindo igualmente avaliar o desempenho da heurística. Nos problemas industriais, os valores dos atrasos em relação às datas devidas de entrega (valores máximos, médios ou número de ordens atrasadas) são geralmente medidas de desempenho com mais significado do que a minimização do prazo de conclusão de todas as ordens de fabrico (*makespan*), que é frequentemente utilizada na literatura como medida de avaliação de sequenciamentos alternativos. As medidas de desempenho em que se baseia a avaliação destes planos de fabrico são:

(i) desvios entre a data devida e a data prevista (valor médio e variância),
(ii) atrasos das ordens de fabrico (valor total, valor médio e percentagem de ordens de fabrico nessa situação);
(iii) tempos de espera das ordens de fabrico (valores médios globais e por centro),
(iv) tempo de permanência no sistema (valor médio),
(v) taxa de utilização dos centros de trabalho (valor médio).

5. Resultados computacionais

Considera-se conveniente referir, antes de mais, que ainda não é possível a avaliação global do desempenho deste sistema de sequenciamento, na melhoria do processo de planeamento, devido ao facto da implementação ainda não estar completa. Contudo, já foram efectuadas experiências em ambiente industrial e os resultados obtidos mostraram uma significativa melhoria na qualidade dos planos de fabrico gerados, realçando-se igualmente a redução do tempo despendido na elaboração dos mesmos. Por outro lado, não foi possível a comparação efectiva entre os planos de fabrico gerados pelos métodos propostos e o sequenciamento de fabrico efectivamente realizado devido ao carácter dinâmico do processo de sequenciamento motivado pelas inúmeras alterações inerentes ao processo produtivo.

A avaliação dinâmica de um plano de fabrico é extremamente difícil. Além da qualidade estática do próprio plano, teria que ser considerada a sua robustez relativamente a possíveis alterações imprevistas que ocorrem durante a respectiva execução, o que obrigaria a quantificar os aspectos aleatórios que perturbam a execução efectiva do plano. Ou seja, obrigaria a uma descrição completa dos diferentes tipos de ocorrências imprevistas, bem como as respectivas probabilidades de ocorrência e uma especificação das tomadas de decisãodo agente de planeamento perantes as várias situações, o que é praticamente impossível, nomeadamente em ambientes industriais do tipo *job-shop*. Desta forma, a análise do desempenho do sistema restringe-se à avaliação estática dos planos de fabrico.

Os teste em ambiente industrial, para aplicação das heurísticas, foram efectuados num computador pessoal. São analisados os resultados da programação das ordens de fabrico de um conjunto de quatro planos de fabrico. A informação relativa às ordens de fabrico a programar foi recolhida em quatro momentos distintos, mais concretamente no início de cada um dos primeiros quatro meses de 1995, tendo sido consideradas apenas as ordens de fabrico cujo processamento ainda não tinha sido iniciado, por ser praticamente impossível definir claramente

o estado de cada uma das ordens de fabrico já iniciadas. No entanto, julga-se que a informação recolhida é representativa do processo de sequenciamento.

Na Tabela 2 são apresentados os valores das medidas de desempenho dos planos de fabrico, resultantes tanto da aplicação do método actual de sequenciamento como da aplicação da heurística API. Os resultados mais detalhados destas medidas de desempenho, bem como dos planos de fabrico gerados, são apresentados em Machado [1996]. Os planos de fabrico são caracterizados por um número médio de 57 ordens de fabrico e por um número médio de 8 operações por ordem. O número de operações a processar em cada centro de trabalho apresenta um comportamento bastante distinto, mas mantém-se a tendência em cada um dos planos analisados. O valor médio das taxas de utilização mensal dos centros é de 25%, apenas pelas ordens em "carteira" no início do mês e cujo processamento ainda não foi iniciado.

Medidas de Desempenho		PLANOS DE FABRICO				Valores médios
		1	2	3	4	
$L_{médio}$ (horas)	Método actual	-10,3	-34,3	-34,7	-17,3	
	Heurística API	-1,8	-18,9	-17,3	-12,7	
	Δ Desempenho (%)	-83	-45	-50	-27	-51
n_T/n (%)	Método actual	42	33	30	35	
	Heurística API	25	11	16	18	
	Δ Desempenho (%)	-40	-67	-47	-49	-51
T_{total} (horas)	Método actual	1308	619	837	1190	
	Heurística API	1103	303	477	604	
	Δ Desempenho (%)	-16	-51	-43	-49	-40
$F_{médio}$ (horas)	Método actual	155,7	106,4	99,1	118,1	
	Heurística API	135,6	83,9	88,3	98,8	
	Δ Desempenho (%)	-13	-21	-11	-16	-15
$W_{médio}$ (horas)	Método actual	80,5	51,9	44,2	57,1	
	Heurística API	60,4	29,4	33,3	37,8	
	Δ Desempenho (%)	-25	-43	-25	-34	-32
$B_{médio}$ (%)	Método actual	61	48	51	53	
	Heurística API	52	38	42	42	
	Δ Desempenho (%)	-15	-21	-18	-21	-19

Tabela 2 - Medidas de desempenho dos planos de fabrico com a aplicação do método actual e da heurística API

Notação: $L_{médio}$ - valor médio dos desvios em relação à data devida.
n_T/n - percentagem de ordens de fabrico atrasadas.
T_{total} - valor total dos atrasos relativamente à data devida.
$F_{médio}$ - valor médio dos tempos de permanência das ordens de fabrico na oficina.
$W_{médio}$ - valor médio dos tempos de espera nos centros de trabalho.
$B_{médio}$ - valor médio das taxas de ocupação dos centros de trabalho, associado ao intervalo de tempo entre o instante mais cedo possível para o início do processamento de alguma ordem de fabrico (independente da forma de definição das prioridades) e o instante de conclusão da última operação.

Da análise dos resultados obtidos (Tabela 2), constata-se que a aplicação da heurística API permite a obtenção de planos com maior qualidade na maioria das medidas de desempenho,

relativamente ao método de sequenciamento utilizado actualmente. Isto não é surpreendente dado que com a elaboração por processos manuais (tarefa morosa) não há possibilidade de ensaiar alterações ao plano gerado. Destacam-se os seguintes resultados:

(i) Redução significativa do valor médio (em termos absolutos) e da variância dos desvios entre as datas previstas e as datas devidas de conclusão, respectivamente em cerca de 50% e 60%, o que representa uma redução dos custos de armazenagem dos produtos acabados, concluídos antes das respectivas data devidas.

(ii) Redução significativa da percentagem de ordens de fabrico atrasadas (cerca de 50%) e do valor total do atraso associado a todas as ordens de fabrico (cerca de 40%).

(iii) Redução significativa dos valores médios dos tempos de permanência e de espera das ordens de fabrico na oficina, na ordem dos 15% e 32% respectivamente, tendo-se considerado o tempo de permanência efectivo das ordens de fabrico na oficina, isto é, desde o início do processamento da primeira operação visto que são os que apresentam significado prático em termos do nível de congestionamento da oficina e dos respectivos custos associados.

(v) Redução do valor médio das taxas de utilização dos centros de trabalho (cerca de 20%), devido essencialmente à metodologia seguida pela heurística, com vista a que o início do processamento das ordens de fabrico seja tão retardado quanto possível (dado que nesta taxa de utilização o limite superior do intervalo de tempo é definido pelo instante de conclusão da última operação). Isto deve-se ao facto de na primeira fase da heurística API ser gerado um plano de fabrico compacto em que raramente se encontram intervalor de tempo sem laboração, que pudessem ser evitados.

Medidas de Desempenho		PLANOS DE FABRICO				Valores médios
		1	2	3	4	
$L_{médio}$ (horas)	Método actual	-10,3	-34,3	-34,7	-17,3	-24,2
	1ª fase da API	-13,4	-33,7	-28,9	-19,0	-23,8
	Heurística API	-1,8	-18,9	-17,3	-12,7	-12,7
n_T/n (%)	Método actual	42	33	30	35	35
	1ª fase da API	35	33	28	33	32
	Heurística API	25	11	16	18	18
T_{total} (horas)	Método actual	1308	619	837	1190	989
	1ª fase da API	1162	632	782	1033	902
	Heurística API	1103	303	477	604	622
$F_{médio}$ (horas)	Método actual	155,8	106,4	99,1	118,1	119,9
	1ª fase da API	152,7	106,9	102,5	116,4	119,6
	Heurística API	135,6	83,9	88,3	98,8	101,7
$W_{médio}$ (horas)	Método actual	80,5	51,9	44,2	57,1	58,4
	1ª fase da API	77,5	52,4	47,5	55,3	58,2
	Heurística API	60,4	29,4	33,3	37,8	40,2
$B_{médio}$ (%)	Método actual	61	48	51	53	53
	1ª fase da API	60	48	52	53	53
	Heurística API	52	38	42	42	44

Tabela 3 - Medidas de desempenho dos planos de fabrico com a aplicação do método actual, da primeira fase da heurística e da aplicação completa da heurística API

A primeira fase da heurística API, que corresponde à geração do plano inicial de fabrico, difere do método actual de sequenciamento apenas na regra de ordenação das ordens de fabrico, que é definida através do índice de prioridade e do valor da folga ou do índice e do valor do quociente da folga sobre o número de operações, respectivamente para o método actual e para a heurística API. Por outro lado, a segunda e terceira fases, que constituem o ciclo iterativo da heurística API, têm o objectivo de deslocar os períodos de processamento das ordens de fabrico

de forma que seja mínimo o desvio entre as datas prevista e devida de conclusão. Na Tabela 3, apresentam-se os resultados obtidos através do método actual, da primeira fase da heurística API e da aplicação completa da heurística API, com vista à avaliação da eficácia das fases correspondentes aos ciclos iterativos.

Apesar das diferenças nos valores das medidas de desempenho (método àctual e primeira fase da heurística API - Tabela 3) não apresentarem a mesma tendência nos vários planos analisados, verifica-se que em termos globais (valores médios) os resultados obtidos através da primeira fase da heurística API são idênticos ou ligeiramente melhores em relação aos resultados obtidos com o método actual de sequenciamento. Donde se conclui que a variação significativa dos resultados das medidas de desempenho se deve primordialmente ao ciclo iterativo da heurística.

Com a aplicação da heurística COMPACT, após a heurística API, destacam-se os seguintes resultados (Tabela 4):

Medidas de Desempenho		PLANOS DE FABRICO				Valores médios
		1	2	3	4	
$L_{médio}$ (horas)	Heurística API	-1,8	-18,9	-17,3	-12,7	
	Heur.Compact	-3,7	-20,7	-20,6	-14,3	
	Δ Desempenho (%)	106	9	19	13	36
n_T/n (%)	Heurística API	25	11	16	18	
	Heur.Compact	25	11	16	18	
	Δ Desempenho (%)	–	–	–	–	–
T_{total} (horas)	Heurística API	1103	303	477	604	
	Heur.Compact	1103	295	477	604	
	Δ Desempenho (%)	–	-3	–	–	–
$F_{médio}$ (horas)	Heurística API	135,6	83,9	88,3	98,8	
	Heur.Compact	142,3	93,3	93,6	108,9	
	Δ Desempenho (%)	5	11	6	10	8
$W_{médio}$ (horas)	Heurística API	60,4	29,4	33,3	37,8	
	Heur.Compact	67,1	38,7	38,7	47,9	
	Δ Desempenho (%)	11	32	16	27	21
$B_{médio}$ (%)	Heurística API	52	38	42	42	
	Heur.Compact	57	42	47	49	
	Δ Desempenho (%)	10	11	12	17	12

Tabela 4 - Medidas de desempenho dos planos de fabrico com a aplicação da heurística COMPACT, após a heurística API

Face a estes resultados (Tabela 4) constata-se que uma das vantagens da "compactação" das operações é o aumento do valor médio das taxas de ocupação, o que é benéfico com vista à minimização dos custos de exploração dos centros, nomeadamente onde operam equipamentos. Adicionalmente, aumenta-se a flexibilidade do plano para acomodar novas ordens de fabrico que possam chegar entretanto. Por outro lado, o aumento do valor médio dos tempos de permanência das ordens de fabrico na oficina deve-se essencialmente ao adiantamento das datas

de início do respectivo processamento, o que provoca aumento no grau de congestionamento da oficina e obriga a que as matérias-primas estejam disponíveis mais cedo, motivando igualmente um aumento dos custos associados aos *stocks* dos produtos em vias-de-fabrico.

Com o intuito de avaliar o desempenho de heurística API comparativamente a um método de sequenciamento amplamente reconhecido, foi aplicada a regra de prioridade baseada na data devida mais cedo, sendo utilizado como critério de desempate a menor folga entre o período de tempo disponível e requerido para produção. Foi igualmente considerado o critério do índice de prioridade para ordenação prévia das ordens de fabrico, tal como na heurística API.

Medidas de Desempenho		PLANOS DE FABRICO				Valores médios
		1	2	3	4	
$L_{médio}$ (horas)	Heurística API	-1,8	-18,9	-17,3	-12,7	-13
	Regra *Due Date*	-14,8	-40,4	-47,0	-26,0	-32
n_T/n (%)	Heurística API	25	11	16	18	17
	Regra *Due Date*	42	24	23	35	31
T_{total} (horas)	Heurística API	1103	303	477	604	622
	Regra *Due Date*	1453	612	405	1130	900
$F_{médio}$ (horas)	Heurística API	135,6	83,9	88,3	98,8	102
	Regra *Due Date*	151,3	100,3	87,9	112,4	113
$W_{médio}$ (horas)	Heurística API	60,4	29,4	33,3	37,8	40
	Regra *Due Date*	76,1	45,8	32,9	51,4	52
$B_{médio}$ (%)	Heurística API	52	38	42	42	44
	Regra *Due Date*	58	48	50	51	51

Tabela 5 - Medidas de desempenho dos planos de fabrico com a aplicação da heurística API e da regra de prioridade baseada na data devida

A análise destes resultados (Tabela 5) permite verificar que o único critério de avaliação que apresenta melhor desempenho com a aplicação da regra baseada na data devida, relativamente à heurística API, é o aumento da taxa de utilização dos centros de trabalho (cerca de 7%). Por outro lado são várias as medidas cujos resultados apresentam significativamente pior desempenho, quando são comparados os dois métodos de sequenciamento na ordem acima referida, nomeadamente o aumento da percentagem de ordens de fabrico atrasadas (cerca de 14%), o aumento do atraso total das ordens de fabrico (cerca de 45%), o aumento dos tempos médios de permanência na oficina das ordens de fabrico e de espera nos centros de trabalho (cerca de 11% e 30%, respectivamente) e o aumento significativo do valor médio (em termos absolutos) e da variância dos desvios entre as datas previstas e devidas de conclusão (cerca de 150%).

Da análise dos valores médios dos tempos de espera das ordens de fabrico em cada um dos centros de trabalho constata-se que os centros em que as ordens de fabrico registam valores médios dos tempos de espera superiores a cerca de 10 horas são aqueles que normalmente são identificados como as máquinas "críticas" (pontos de estrangulamento), pelos responsáveis dos sectores produtivos [Machado, 1996]. Nestas máquinas são realizadas operações de impressão,

devendo ser obrigatoriamente seguido o pressuposto de não interrupção do processamento das operações.

Uma das vantagens da abordagem utilizando a heurística API reside no facto do procedimento de pesquisa das segunda e terceira fases ter provado ser bastante eficiente, desde que não sejam consideradas as ordens de fabrico atrasadas que se apresentam em posições superiores na ordenação de afectação (maior prioridade) que a posição mais elevada de qualquer ordem de fabrico adiantada e reescalonada. Isto é uma consequência do facto destas ordens de fabrico já terem sido escalonadas utilizando a heurística AP e apenas o movimento de uma ordem de fabrico adiantada pode originar a abertura de intervalos de tempo sem laboração no plano, sendo assim um limite efectivo no procedimento de pesquisa.

As vantagens fundamentais da aplicação da heurísitca API consistem na melhoria significativa do desempenho do plano de fabrico, em relação à aplicação apenas da heurística AP, em termos da redução do número de ordens de fabrico atrasadas, da redução dos desvios (valor médio e variância) em relação às datas devidas e da redução dos tempos médios de permanência na oficina e de espera nos centros de trabalho. O plano de fabrico apresenta, contudo, uma redução no valor médio das taxas de ocupação dos centros de trabalho. No entanto, este último resultado desfavorável poderá ser reduzido com a aplicação da heurística COMPACT provocando, por outro lado, um aumento nos tempos de permanência na oficina de algumas ordens de fabrico. Esta situação de compromisso terá que ser devidamente avaliada em cada caso específico.

6. Conclusões e desenvolvimentos futuros

O SAD descrito neste trabalho incorpora um número de procedimentos que fornecem efectiva assistência aos planeadores, nomeadamente na geração de soluções iniciais, verificando a admissibilidade das acções e a apresentação das consequências de decisões alternativas que o agente de planeamento pretender simular. Com a implementação completa do sistema espera-se que ocorra uma melhoria muito significativa no funcionamento do processo de sequenciamento da produção. Além da redução do tempo despendido na elaboração do plano, salientam-se outros aspectos relevantes relacionados com a facilidade de utilização do sistema e a possibilidade de incorporação da experiência e do conhecimento do agente de planeamento na solução do problema.

A aplicação da heurística API permite a obtenção de planos de fabrico com maior qualidade no que respeita às medidas de desempenho associadas ao cumprimento dos prazos de entrega, aos tempos de permanência na oficina e de espera nos centros de trabalho das ordens de fabrico.

Além dos desenvolvimentos inerentes à implementação do sistema, a melhoria das heurísticas aplicadas é outro dos assuntos que deverá merecer significativa atenção na investigação a desenvolver no futuro. Considera-se que o recurso a meta-heurísticas é uma abordagem promissora para investigação futura com vista à melhoria do desempenho das heurísticas de sequenciamento de operações.

Finalmente, o carácter multi-critério do problema suscita questões que importa investigar, nomeadamente sobre as formulações e processos de pesquisa da "solução de melhor compromisso" quando se está em presença de objectivos conflituosos e as relações de troca (*trade-offs*) entre objectivos não são explícitas e variarão com o agente de planeamento, o contexto e a pressão percebida em cada instante.

Referências

[1] Adams, J., Balas, E. and Zawack, D., *The Shifting Bottleneck Procedure for Job Shop Scheduling*, Management Science 34 (1988) 391-401.

[2] Baker, K.R, *Introduction to Sequencing and Scheduling*, John Wiley Publishers, New York (1974).

[3] Barnes, J., Laguna, M. and Glover F., *An Overview of Tabu Search Approaches to Production Scheduling Problems*, Intelligent Scheduling Systems, edited by Brown, D. and Scherer (1995) 101-128.

[4] Brown, D., Marin, J. and Scherer, W., *A Survey of Intelligent Scheduling Systems*, Intelligent Scheduling Systems, edited by Brown, D. and Scherer (1995) 1-40.

[5] Carlier, J. and Pinson, E., *Une Methode Arborescente Pour Optimiser la Durée d'un Job-Shop*, Les Cahiers de Le Institut de Mathématiques Appliquées, Angers (1988).

[6] French, S., *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, Ellis Horwood, Chichester, England (1982).

[7] Hurrion, R.D., *An Investigation of Visual Interactive Simulation Methods Using the Job-Shop Scheduling Problem*, Journal of Operational Research Society 29 (1978) 1085-1093.

[8] Keen, P.G. and Scott Morton, M., *Decision Support Systems: An Organizational Perspective*, Addison-Wesley Publishing Company, USA (1978).

[9] Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G., *Recent Developments in Deterministic Sequencing and Scheduling: A Survey*, Deterministic and Stochastic Scheduling, M.A.H. Dempster et al. (eds) (1982) 35-73.

[10] Machado, M.L., *Sequenciamento da Produção na Indústria Gráfica*, Dissertação para obtenção do grau de Mestre em Investigação Operacional e Engenharia de Sistemas, Instituto Superior Técnico (1996).

[11] Marques, M.P., *Programação de operações fabris em ambiente de "job-shop": Nova abordagem*, Tese de doutoramento, Faculdade de Engenharia da Universidade do Porto (1993).

[12] McMahon, G. and Florian, M., *On Scheduling with Ready Times and Dues to Minimize Maximum Lateness*, Operations Research 23 (1975) 475-482.

[13] Moreira, N.A. and Oliveira, R.C., *A Decision Support System for production planning in an industrial unit*, European Journal of Operational Research 55 (1991) 319-328.

[14] Speranza, M.G. and Woerlee, A.P., *A Decision Support System for operational production scheduling*, European Journal of Operational Research 55 (1991) 329-343.

[15] Sprague Jr., R.H. and Watson, H.J., *Decision Support Systems: Putting Theory into Practice*, 3rd edition, Prentice-Hall, United States of America (1993).

[16] White, C., *Production scheduling with applications in the printing industry*, European Journal of Operational Research 22 (1985) 304-309.

INVESTIGAÇÃO OPERACIONAL, vol. 16 (Dezembro 1996) 155-172

QUEUEING AND QUALITY SERVICE

M. F. Ramalhoto

Instituto Superior Técnico
Department of Maths
Av. Rovisco Pais
1096 Lisboa
Portugal

R. Syski

University of Maryland
Department of Maths
College Park
Maryland 20742
USA

Abstract

This paper develops methodology that provides a way of incorporating quality management concepts of customer's satisfaction into the queue design. It reviews Garvin's quality dimensions of performance, flexibility, serviceability, reliability, courtesy and appearance and interprets these in a queueing context. The last part of the paper deals with a queue-design that addresses some of those quality issues. It incorporates the concepts of alarm and action lines into the queueing model, through what is called, rule - 1: when the number of customers in the system is equal or larger than b (action line) add k extra servers, when it is equal or smaller than a (alarm line) remove k servers (a < b).

A queueing model with c exponential identical servers and Poisson arrivals, operating under rule - 1, is studied. The equilibrium distribution of the state of the two-dimensional Markov process that characterizes the queueing system is derived. Some first passage time problems useful in the quality design of the queueing system are solved.

Resumo

É apresentada uma metodologia que permite incorporar conceitos de gestão de qualidade nos modelos matemáticos clássicos de filas de espera. As características de qualidade-desempenho, flexibilidade, "serviçabilidade" (capacidade de resposta a reclamações), fiabilidade, cortesia e aparência, definidas por Garvin, são revistas e interpretadas no contexto dos sistemas de filas de espera. Para qualitifcar as quatro primeiras características de qualidade são apresentadas medidas de fila de espera. A qualidade de um sistema de filas de espera pode então ser avaliado através dos valores dessas medidas. A maior parte dos modelos matemáticos clássicos de filas de espera não são capazes de responder rapidamente a mudanças de taxa de chegada ou de serviço. O que de um modo geral, acarreta um número inaceitável de cleintes à espera de serem atendidos e consequentemente, descida de alguns dos valores das medidas de qualidade consideradas. Na secção-3, é feito o estudo analítico de uma fila de espera com processo de chegadas Poisson, tempos de serviço exponências e a seguinte regra de decisão-quando o número de clientes no sistema é igual ou maior que b (linha de acção) introduzem-se no sistema k extra servidores, ficando o sistema com k+c servidores, e quando é igual ou menor que a (linha de alarme ou prevenção) retiram-se os k extra servidores (a<b). É obtida a distribuição de equilíbrio do estado do processo de Markov bivariado que define esta fila de espera. São resolvidos problemas de tempos de primeiras passagens, também de utilidade no planeamento da melhoria da qualidade deste sistema.

Keywords

Garvin's quality dimensions; queueing models with two queue size boundaries; prevention line; action line; first passage times.

1. Introduction

The main contributions of this paper are, on one hand, the methodology presented, which provides a way to incorporate quality management concepts of costumer's satisfaction into the queueing design and to quantify them. And on the other hand, the development of a queue-design that addresses some of those issues by incorporating the concepts of prevention or alarm line and action line into the queue model, through what is called here rule-1. That queue-design illustrates one of the various types of queue-designs that migjt be useful, and the type of analytical and numerical questions that have to be addressed, in each queue-design.

In the service industry there are essentially two types of products to be considered; the product-service and the product-supply. The product-service can be defined as how has the service been provided and the product-supply is what has been provided.

The quality improvement of the product-supply is, in most cases, linked to stochastic reliability, stochastic quality control and experimental design techniques, and is not treated here. Clearly, in most cases, a poor product-service might ruin an excellent quality product-supply and vice-versa.

The product-service is usually provided through a queueing system. Therefore, the stochastic modeling of those queues plays an important role in the quality improvement of the product-service.

A methodology to establish the link between queues and quality management is introduced in section-2. Essentially, it provides a description of how quality management concepts of satisfying the customer can be incorporated into the design of queueing systems. Garvin's quality dimensions of performance, flexibility, serviceability, reliability, courtesy, and appearance are reviewed and interpreted in a queueing context. For the first four quality dimensions some queueing measurements are suggested to quatify them.

It is advocated that queueing systems in the service industry should be carefully designed to provide a product-service of high quality. The quality of the product-service is measured by high ranks in some or all of the six quality dimensions, flexibility, serviceability, reliability, courtesy, and appearance.

To design a queueing system to have a high rank in the quality dimension flexibility, it is not easy. In fact, most of the traditional queueing systems are unable to respond quickly to changes in their environment. Those quick changes, very often, are shown up trough, for instance, an unacceptable queue size. Moreover, a poor rank in flexibility, might lead to poor ranks in almost all the other quality dimensions.

Therefore, section-3 presents a queueing model that aims to provide managers with a way of dealing with some temporary peak situations. In considers a G/G/c queue under rule-1. Boundary "a" acts as a prevention (or alarm) line to call the attention to the manager that, most likely, in a short time some extra servers have to be put into action. Boundary "b" is the action

line, that is to say, when the queue length reaches b then the extra k servers will be added to the queueing system. They will be removed when boundary a is reached again.

Section-3 also presents a detailed steady-state analysis of that queue, with c exponential identical servers and Poisson arrivals. The equilibrium distribution of the state of the two-dimensional Markov process that characterizes the queueing system is derived. Some relevant first passage time questions are answered. For instance, the entrance probability to having a or less customers, (before having b or more customers), when starting at the boundary (b, c + k), $D_{b,c+k}$, is a simple function of parameters, a, b, c, k and the arrival and service rates. For instance, the maximization of $D_{b,c+k}$ in all or part of those parameters provide some interesting guidelines for the queue design, even when the steady-state analysis is inappropriate. The exact steady-state analysis provided is an important reference for any numerical approach to be performed in the nonstationary case.

The Kolmogorov system of equations for the associated nonstationary queue (non-homogeneous Poisson arrivals and service rates also functions of time) are outlined. For the associated nonstationary queue our steady-state solutions could provide a starting point in the search for a numerical method to calculate the time-dependent distribution of the number in the system (as the numerical solution of the Kolmogorov time-dependent differential equations). This method was successfully used by Green et al. (1991) in the context of 1-server queue and by Taafe and Ong (1987) for the Ph(t)/M(t) /c/s queue.

1.1 The quality movement

The quality movement has enjoyed a high profile through the eighties and the nineties all over the world. Its most formal and official expression is through the quality standards (the ISO 9000 series and BS 5750) that relate to the methods organizations use to ensure consistency of quality in goods or services that they provide. In addition to this very specific application of quality principles the term TQM is used to describe a management approach that aims to apply quality principles to all aspects of a company's activity.

1.2 The queueing system

Most of the design problems posed by the service industries concern facilities serving a community os users. Typically, both the times at which the users ask for service and the lenght of the service times themselves are stochastic, so inevitably congestion occurs and queues may build up.

To analyze resource allocation and the job flow through computer systems, perhaps, queueing theory is the only method available to understand the behaviour of their complex interconnections. In fact, for three quarters of a century, the field of communication has provided both a stimulus to the development of, and a rewarding application are for queueing theory methods.

Most of the significant problems in manufacturing industries can also be reduced to the problems of resource allocation and resource sharing.

Therefore, queueing methodology is relevant to the designers and users of complex systems perform under different conditions. However for being able to do it in more realistic practical terms, there is still a need to develop new types of queueing models under adequate decision frameworks, borrowing ideas from quality control, inventory policies, and other decision making areas. For example, Garvin (1988) stated that the quality of a product or service may have many dimensions and that among them:

- Performance
- Flexibility
- Serviceability
- Reliability
- Courtesy
- Appearance

are perceived by the user and are determined by the designer. It is also stressed that any quality improvement effort must take into consideration the customer requirements. Our point is that all of these dimensions are relevant to queueing systems of any use in practice and we should develop queueing models able to address them.

2. The quality dimensions - performace, flexibility, serviceability, reliability, courtesy, appearance - in a queueing system's context

Let us review Garvin's quality dimensions - performace, flexibility, serviceability, reliability, courtesy, appearance - and interpret these in the context of queueing systems in the service industry. The definitions above given are based on the books, Garvin (1988), Zeithami et al (1990) and Bergman and Klefsjo (1994, Chapter 15).

- *Performance* - the primary operating characteristics of the queueing system. It can be measured through, for instance, the "absence of waiting time", "total sojourn time in the system not exceed X units of time", "Y percent of all customers rate the 'product-service' received as excellent", etc.
- *Flexibility* - the queueing system's built-in ability to quickly respond to the changes of demand. It can be measured through, for instance, the mean duration of a traffic peak (how quickly it gets rid of it).
- *Serviceability* - the responsiveness of the queueing system. It can be measured, for instance, through mean time to answer enquires, mean time to answer complaints, etc.
- *Reliability* - the servers ability to always perform their job dependably, knowledgeably and accurately. It can be measured, for instance, through the number of errors and complaints due to the servers performance.
- *Courtesy* (empathy) - the caring, individualized attention provided to customers. Those are factors more linked to standards of preferential human behavior which are most

subjective and difficult to control and evaluate. They need separate attention and joint research work with human behavior specialists, in order to set up adequate measurements.

- *Appearance* (tangibles) - the organization, physical facilities, equipment, communication materials. To measure this queueing quality dimension, it is also required joint research wotk with other specialists to seut up the right questions to lead to the adequate measurements.

Nevertheless it is perhaps reasonable to conjecture that when the first four quality dimensions are doing well, that is to say, when the queue design is well adjusted to its customer's demands, the peaks are reasonably controlled and the servers reliable, the other two dimensions are easier to define and to improve. Otherwise, a very kind server who does not know the job well, very soon will be considered to be of little use for the customer. An office full of well dressed servers and sophisticated computers and goof furniture is not necessarily the most important factor for the customer, namely if the other four quality dimensions are not high. It also can be an insult, because all this luxury is going to be paid directly or indirectly (tax payers, for instance) by the customers, therefore a balance has to be reached.

The dimensions courtesy and appearance, as already mentioned, clearly call for the need to bring more human behavioral patterns and responses into the analysis of queueing systems. Nevertheless, they can also be made to look better if an adequate queueing system's design is provided.

The reduction of a peak duration is extremely important. It affects directly the quality dimension flexibility and might have an indirect effect on the other quality dimensions. Therefore, queueing models to reduce temporary peak situations have to be developed. The next section presents a queue design to address this issue.

2.1 The G/G/c queue under rule-1

Let us recall that the rule-1 is defined as follows:

Rule-1: If the queue exceeds b (the action line), introduce another server (or k servers, $k \geq 1$). If it falls below a (the prevention line), withdraw one server (or k servers, $k \geq 1$), where $b > a$.

In some of the queueing systems of the service industry the arrivals are either bursty in nature or the arrival rate to the system is subject to random fluctuations. To design these queueing systems to meet the peak demand is not always the best action to take. Because, it can be very costly, and the excess capacity can have negative psychological effects on customers. The rule-1 added, for example, to the traditional G/G/c/c+d queue, $c = 1, 2, \ldots$, $d = 0, 1, \ldots, +\infty$ make it more able to respond to the change of its operation environmental needs (namely, changes of its arrival rates), and contributes to the customer's satisfaction. It makes queueing systems closer to learning systems, to use the modern language of quality management. Moreover, in real-life, most of changes of arrival rate situations imply that, we have not a stable

queueing model, i.e., the intensity of traffic is not always less than one. Therefore, rule-1 can also be seen as a mechanism to handle "temporary unstability". That means, we do not interfere till for instance the queue length becomes "unacceptable", that is to say, exceeds b (the action line). Then we add k servers. In principal, the rule can be added to most of the classical queueing models, with any type of arrival process and service distribution, any number of servers, finite or infinite waiting capacity and for most queueing disciplines. The temporary removed k servers, when dispensed, could be engaged in "serviceability activities" - answering complaints and inquiries and reporting them to the queueing system's manager for system's learning purposes - or in continuing education and training activities (through flexible and distance learning approaches) or in quality circle activities, just to name a few important subsidiary activities. In fact, by adding rule-1 to a queueing model, for instance, of the G/G/c/c+d type, three extra parameters, a, b and k are also added. In some cases a and b might be fixed by space and management's reasoning rather than by optimal mathematical criterium. Instead k might be an interesting parameter to optimize, in the context of serviceability, quality circles, continuing education and training activities.

Before presenting an analytical study of the exponential version of the c-server queue with alarm and action lines, let us comment on the existing literature related to this paper.

2.2 Comments on literature

This paper is not really concerned with the classical control of queues. Where the earlier papers are in the "static" design category. Whereas the latest ones are concerned with search for optimal policies using semi-Markov decision theory and other dynamic programming methodologies. There are quite a lot of literature available on this topic, see, for instance, Gross and Harris (1986, section 6.5.2) and the papers cited there.

In Ramalhoto (1991) the class of problems following rule-1 was introduced and the equilibrium distribution of the state of the system, in the case of the M/M/c queueing model under rule-1, with one threshold was given, as well as some comparison with the usual M/M/c queue for first moments.

In that paper the two threshold case was briefly mentioned in section 3.1.2. The proposition in this section has some mistakes that will be completely clarified in section-3 of the present paper, where the complete proof of the proposition is given.

Ramalhoto et. al. (1990) also presents the equilibrium distribution of the M/M/c queue under rule-1 in the two threshold case (this paper is cited in the above paper, Ramalhoto (1991)).

Concerning the mathematical model presented in the section-3 of this paper, we are aware of the work by Ibe and Keilson (1992) who consider independently a related problem but not the actual queueing model presented here.

Romani (1957) considers also addition of servers, but his model is basically different from ours.

There are many books concerning quality approaches to the service industry. For example, Rosander (1985) extends the traditional statistical quality control tools to the service environment. As Deming (1986, 1993) stresses inspection is equivalent to planning for defects. Instead, processes must be improved. The approach to quality in queueing systems of the service industry that is followed in this paper, is very much linked to the so-called TQM methodology. For an overall picture of the current perception of TQM and its supporting methods see, for instance, Bergman and Klefsjö (1994) and the references cited there.

There is also a considerable literature on customer's perception of queueing systems such as, for instance, Hall (1991), Larson (1987, 1988), Maister (1985) and Osuna (1985).

3. The M/M/c queue under rule-1

Let us consider an M/M/c queue, that is to say, a queueing system with c exponential identical servers with rate μ, unlimited queue size, first-come-first served discipline, and Poisson arrivals with rate λ.

In addition, let us consider that this queueing system is under the rule-1, which means that - when the number of customers in the system is equal or larger than b, add k extra servers; when it is equal or smaller than a remove k extra servers ($a < b$). If the service of one customer is completed when there are a+1 customers in the system, and there are c+k servers operating, then k servers are removed.

It will be assumed that $b - a \geq 1$, with discussion restricted mainly to the case $b - a > 1$; the case $b - a = 1$ leads to great simplification of results.

3.1 The two-dimensional Markov process

Let us model the queueing system under consideration as a two-dimensional Markov process (X_t, Y_t) where X_t represents the number of customers in the system at time t, and Y_t indicates the number of servers at time t. The two-dimensional state space of the process is then a collection of pairs (i,n) where i = 0, 1, 2,..., and n = c,c+k only.

The absolute probabilies of the process, denoted by

$$P_{(i,n)}(t) = P(X_t = i, Y_t = n),$$

satisfy the system of the differential equations of the form:

$$dP(t)/dt = P(t)Q, \quad t \geq 0, P(0) = I,$$

where P(t) is a (row) vector of probabilities $P_{(i,n)}(t)$, I is the unit vector, and $Q = (q_{(r,m)(i,n)})$ is the generator matrix of intensities of the process.

Let $e_{(i,n)} = \lim_{t \to \infty} P_{(i,n)}(t)$, (1)

denote the equilibrium distribution of the state of the two-dimensional Markov process, and let $e = (e_{(i,n)})$ be the corresponding vector. The matrix equation for e is then given by

$$e\ Q = 0. \quad (2)$$

The matrix Q characterizes the structure of the system and its coefficients are determined by the rule-1. Then, the equation may be solved for the steady state distribution e (see sub-section 3.2).

Operation of the system indicates that the process (X_t,Y_t) behaves like a birth-and-death process, except at states when switching of number of servers take place.

Indeed, Poisson input implies that transitions of X_t from i to i+1 for any number of servers occur with constant rate equal to λ for all i. Thus the corresponding intensities have the form:

$$q_{(i,c)\,(i-1,c)} = \lambda,\ 0 \le i \le b-2$$
$$q_{(b-1,c)\,(b,c+k)} = \lambda,\ i = b-1 \qquad (3)$$
$$q_{(i,c+k)\,(i+1,c+k)} = \lambda,\ i \ge a+1.$$

However, terminations of X_t from i to i-1 depend on the number of servers engaged, that is on the value of Y_t. In fact, when i is between a and b, it is necessary to define two phases:

- *phase-upwards*, where there are c servers operating, because although a has been already reached, b has not, or because having revisited a after b has been reached, b has not yet been revisted.
- *phase-downwards*, where there are c+k servers operating, because b has been reached and a has not yet been revisited, after that.

Assume now for simplicity that c+k < a, so c < a. Hence, c+k < i for i > a. Then, the corresponding intensities have the form:

$$q_{(i,c)\,(i-1,c)} = i\mu,\ 1 \le i \le c$$
$$q_{(i,c)\,(i-1,c)} = c\mu,\ c+1 \le i \le a \qquad (4)$$
$$q_{(i,c)\,(i-1,c)} = c\mu,\ a+1 \le i \le b-1, \text{ upwards}$$
$$q_{(i,c+k)\,(i-1,c+k)} = (c+k)\mu,\ a+1 \le i \le b-1, \text{ downwards}$$
$$q_{(i,c+k)\,(i-1,c+k)} = (c+k)\mu,\ i \ge b.$$

All the other intensities for other pairs are zero, by usual convention, and as row sums of matrix Q are zero, the diagonal terms are negative of

$$q_{(i,c)} = \lambda,\ i = 0$$
$$q_{(i,c)} = \lambda + i\mu,\ 1 \le i \le c \qquad (5)$$
$$q_{(i,c)} = \lambda + c\mu,\ c+1 \le i \le b-1$$
$$q_{(i,c+k)} = \lambda + (c+k)\mu,\ i \ge a+1.$$

Note that states (i,c) and (i,c+k) for i = a+1,..., b-1 may be regarded as forming a hysteresis loop, with states (a,c) and (b,c+k) corresponding to changes of number of servers. It is presence of hysteresis which makes this problem different from the classical one-dimensional birth-and-death process (see Section 3.3).

3.2 Steady state solution

Substituting intensities given by equations (3), (4) and (5) into the equation (2) for the steady state (vector) e, the unique solution of the system (2) can be obtained by rather tedious

calculations of the usual type. Using the pivotal states (a,c) and (b,c+k), we may express the required solution in terms of $e_{(b-1,c)}$ as follows.

Write

$$\rho = \lambda/[(c+k)\mu], \quad \tau = \lambda/(c\mu) \tag{6}$$

and note that $\rho < \tau$. For the existence of the steady state probabilities $e_{(i,n)}$, it must be assumed that $\rho < 1$. However, no such restriction is needed for τ; see subsection 3.5.

Hence, the steady state probabilities have the form:

$$\begin{aligned} & e_{(i,c)} = \tau^i c^{i-c} (c!/i!)\, \varphi(0,\tau)\, e_{(b-1,c)}, && 0 \le i \le c-1 \\ & e_{(i,c)} = \tau^i \varphi(0,\tau)\, e_{(b-1,c)}, && c \le i \le a \\ \text{up,} \quad & e_{(i,c)} = \tau^a \varphi(i-a,\tau)\, e_{(b-1,c)}, && a+1 \le i \le b-1 \\ \text{down,} \quad & e_{(i,c+k)} = \Psi(i,\rho)\, \rho\, e_{(b-1,c)}, && a+1 \le i \le b-1 \\ & e_{(i,c+k)} = \Psi(b,\rho)\, \rho^{i-b+1} e_{(b-1,c)}, && i \ge b, \end{aligned} \tag{7}$$

where

$$\begin{aligned} & \varphi(i,\tau) = (\tau^i - \tau^{b-a})(\tau^{b-1} - \tau^b)^{-1}, \\ & \Psi(i,\rho) = (1 - \rho^{i-a})(1 - \rho)^{-1}. \end{aligned} \tag{8}$$

Moreover, $e_{(b-1,c)}$ is obtained from normalization:

$$\sum_i e_{(i,c)} + \sum_i e_{(i,c+k)} = 1. \tag{9}$$

It is of interest to see how this solution simplifies when $\tau = 1$. Taking the limit, we find that

$$\varphi(i,1) = b-a-i,$$

and $\Psi(i,\rho)$ has the same form but with

$$\rho = c/(c+k).$$

Then, formula (7) yields the steady state probabilities:

$$\begin{aligned} & e_{(i,c)} = (c!/i!)c^{i-c} (b-a)\, e_{(b-1,c)}, && 0 \le i \le c-1 \\ & e_{(i,c)} = (b-a)\, e_{(b-1,c)}, && c \le i \le a \\ \text{up,} \quad & e_{(i,c)} = (b-i)\, e_{(b-1,c)}, && a+1 \le i \le b-1 \\ \text{down,} \quad & e_{(i,c+k)} = \rho(1-\rho^{i-a})(1 - \rho)^{-1} e_{(b-1,c)}, && a+1 \le i \le b-1 \\ & e_{(i,c+k)} = \rho^{i-b+1} (1-\rho^{b-a})(1 - \rho)^{-1} e_{(b-1,c)}, && i \ge b. \end{aligned} \tag{10}$$

The expression for $e_{(b-1,c)}$, found from (9), is:

$$\begin{aligned} & (b-a)\left[(c!/c^c) \sum_{i=0}^{c-1} (c^i/i!) + a-c-1 + (b-a-1)/2\right] + \\ & +\rho(1-\rho)^{-1}\left[b-a-1 - \rho(1 - \rho^{b-a-1})/(1-\rho) + (1 - \rho^{b-a})/(1-\rho)\right] = 1/e_{(b-1,c)}. \end{aligned} \tag{11}$$

This expression simplifies considerably when $b-a = 1$.

3.3 First passage times

This section is concerned with probability of the first entrance from the set of states with either phase-up or phase-down (hysteresis loop) to the set of states where the system works with a fixed number of servers.

Denote by L the hysteresis loop, namely states (i,c) and (i,c+k) both for i = a+1,..., b-1, and by B the boundary states (a,c) and (b,c+k). Denote by H_1 the set of states (i,c) for i = 0, 1,..., a-1, and by H_2 the set of states (i,c+k) for i = b+1, b+2,... Let $H = H_1 \cup H_2$. Then, the complement H^c is the set which contains the loop L and the boundary B.

As H^c is a finite set, probability of entering H (starting in H^c) is 1. However, probability of entering H_1 (before H_2) is less than 1.

Let $T_1 = \inf(t : X_t \in H_1)$ and $T_2 = \inf(t : X_t \in H_2)$ be the first entrance time to H_1 and to H_2, respectively. Define $T = \min(T_1, T_2)$.

Let

$$D_{(i,n)} = \mathbb{P}\subset T < \infty,\ T_1 < T_2 \mid X_0 = i,\ Y_0 = n), \tag{12}$$

be the entrance probability to H_1 (before H_2), when starting at (i,n), with n = c,c+k. By definition, $D_{(i,c)} = 1$ when starting in H_1, and $D_{(i,c+k)} = 0$ when starting in H_2.

As it well known (see Syski (1992), page 173), the vector $D = (D_{(i,n)})$ is the unique solution of the equation $Q\,D = 0$ on H^c for the Dirichlet problem. More specifically,

$$\sum_{(j,m)} \sum q_{(i,n)\,(j,m)}\, D_{(j,m)} = 0,\ \ (i,n) \in H^c, \tag{13}$$

with the boundary conditions

$$D_{(i,c)} = 1 \ \text{ for } \ (i,n) \in H_1,\ D_{(i,c+k)} = 0 \ \text{ for } \ (i,c+k) \in H_2, \tag{14}$$

The situation corresponds to a genuine two-dimensional gambler's ruin problem.

Substituting intensities from equations (3), (4) and (5) into the equation (13) for the vector D, the solution of the system (13) is found for $a \le i \le b-1$ to be:

$$D_{(i,c)} = \xi(b+a-2,\tau) + \xi(i,\tau)\, \tau^{b-i}\, D_{(b,c+k)}, \tag{15}$$

$$D_{(i,c+k)} = \xi(i,\rho)\, D_{(b,c+k)}, \tag{16}$$

where

$$\xi(i,\tau) = (1-\tau^{b-i+1})\,(1-\tau^{b-a+1})^{-1},\ \ \xi(i,\rho) = (1-\rho^{b-i+1})\,(1-\rho)^{-1}, \tag{17}$$

with

$$1/D_{(b,c+k)} = 1 + \xi(a+1,\rho)\,\rho\,(1-\tau^{b-a+1})\,(1-\tau^{b-a})^{-1}. \tag{18}$$

For more details on first passage times for Markov chains see Syski (1992) pages 159-164.

Observe that letting $\tau \to 1$, we obtain in the limit:

$$D_{(b,c+k)} = \left[1 + \rho(1-\rho(1-\rho^{b-a})\,(1-\rho)^{-1}\,(b-a+1)\,(b-a)^{-1}\right]^{-1}, \tag{19}$$

with $\rho = c/(c+k)$.

3.4 Mean first passage times

It may be of interest to investigate the mean of the first entrance time T to the set H (duration of waiting time to absorption), when starting in H^c. Recall that probability of finite T is one.

Write

$$M_{(i,n)} = \mathbb{E}\left[T \mid X_0 = i,\ Y_0 = n\right] \text{ for } (i,n) \in H^c\,, \tag{20}$$

and M for the vector with components $M_{(i,n)}$. Then, the matrix equation for M is Q M = -1, or in terms of components

$$q_{(i,n)}\, M_{(i,n)} - 1 = \sum \sum q_{(i,n)\,(j,m)}\, M_{(j,m)}\,, \tag{21}$$

with summation extended over (j,m) in H^c such that $(j,m) \neq (i,n)$, and the boundary conditions

$$M_{(a-1,c)} = M_{(b+1,c+1)} = 0 \tag{22}$$

see Syski (1992), theorem 10, page 86.

Substituting intensities from equations (3), (4) and (5) into the equation (21), using the boundary conditions (22) and introducing auxiliary states (b,c) and (a,c+k), with identification:

$$M_{(a,c)} = M_{(a,c+k)}\,,\ M_{(b,c)} = M_{(b,c+k)}\,, \tag{23}$$

the equation (21) reduces to the system of two equations:

$$(\lambda+c\mu)\, M_{(i,c)} - 1 = \lambda M_{(i+1,c)} + c\mu M_{(i-1,c)},\ a \le i \le b-1 \text{ up}, \tag{24}$$

$$[\lambda+(c+k)\mu]\, M_{(i,c+k)} - 1 = \lambda M_{(i+1,c+k)} + (c+k)\mu M_{(i-1,c+k)},\ a+1 \le i \le b \text{ down}, \tag{25}$$

Thus, we obtain two coupled equations, each of the same form. Solving in the usual way (see Feller, vol.1 page 348), and using the boundary conditions (22), the solution of the equation (24) for the upwards transitions has the form for i = a-1,..., b:

$$M_{(i,c)} = (i-a+1)\,[c\mu(1-\tau)]^{-1} + B_1[\tau^{-i} - \tau^{-(a-1)}]\,, \tag{26}$$

In the same manner, the solution of the equation (25) for the downwards transition is for i = a,..., b+1:

$$M_{(i,c+k)} = (i-b-1)\,[(c+k)\,\mu(1-\rho)]^{-1} + B_2[\rho^{-i} - \rho^{-(b+1)}]\,, \tag{27}$$

Constants B_1 and B_2 are then obtained by substitution of (26) and (27) into relation (23), and solving the resulting system of two linear equations for B_1 and B_2.

This argument confirms observation already noted earlier that the effect of up and down phases must be considered jointly because of coupling.

The routine calculations are, however, not very informative, although some simplification is obtained in the special case b-a = 1. On the other hand, in the case when $\tau = 1$ and $b-a \geq 1$ equations (24) and (25) become:

$$2M_{(i,c)} - \lambda^{-1} = M_{(i+1,c)} + M_{(i-1,c)}\,,\ a \le i \le b-1\,, \tag{28}$$

$$(1+\rho)M_{(i,c+k)} - \lambda^{-1}\rho = \rho M_{(i+1,c+k)} + M_{(i-1,c+k)}\,,\ a+1 \le i \le b\,, \tag{29}$$

with $\rho = c/(c+k)$.

Proceeding in the same fashion (see Feller, vol.1, page 349), we find that equation (26) is replaced by

$$M_{(i,c)} = [-i^2 + (a-1)^2]\,(2\lambda)^{-1} + B_1\,(i-a+1)\,,\quad a-1 \le i \le b\,, \tag{30}$$

whereas equation (27) remains unchanged with $\rho = c/(c+k)$.

Consequently, the system of equations for constants B_1 and B_2 is in this case:

$$(1-2a)\,(2\lambda)^{-1} + B_1 = -(b-a+1)\,c(k\lambda)^{-1} + B_2\,[\rho^{-a} - \rho^{-(b+1)}]\,, \tag{31}$$

$$[-b^2 + (a-1)^2]\,(2\lambda)^{-1} + B_1\,(b-a+1) = -c(k\lambda)^{-1} + B_2\,[\rho^{-b} - \rho^{-(b+1)}]\,. \tag{32}$$

3.5 Remarks

1. Recall that coefficients τ and ρ have been defined in (6), and $\rho < \tau$. For stability reasons it is necessary to assume that $\rho < 1$. Concerning τ, there are no reasons for any restrictions. The case $\tau < 1$ is perhaps most natural. However, the case $\tau \ge 1$ may occur in sets H_1 and in upward part of hysteresis loop L (where the queue is essnetially finite). This is very interesting because it allows the possibility of stabilization of an unstable queue by addition of some servers. This situation has been mentioned in conclusion at the end of subsection 2.1, and the case $\tau = 1$ has been considered in subsection 3.2 and 3.4.
2. Though much messier, the analytical aproach followed in this section can still be used for finite waiting capacity Markovian queues.

   For more general service and inter-arrival times distribution, a computation approach as well as algorithmic analysis seems to be the appropriate way to follow.
3. As we mentioned earlier, the paper by Romani (1957) considers also addition of servers, but its model is basically different from the model treated here. In Romani model the number of servers ranges from 0 to ∞, but the waiting room is M. When the number of waiting customers reaches M, then one server is added; when there are no customers, then one server is removed. There is no hysteresis loop. Romani writes equations for state probabilities of two dimensional Markov chain, and obtains explicit solutions by interesting algebraic operations.

   Studies of the Ibe and Keilson, mentioned earlier, treat hysteresis models, consisting of several lines of a special structure. These authors do not consider analytic solutions for steady state distributions.
4. Of special interest is formula (18). It represents entrance probability $D_{(b,c+k)}$ to the set of states (i,c) for i = 0,..., a-1 (before entering the set of states (i, c+k) for i = b+1, b+2,...), when starting from the boundary state (b, c+k).

   Interpreting a as a prevention line and b as an action line, (18) gives indication of a tendency towards c servers, when starting with c+k servers. In other words, this may serve to measure preference of using c+k servers for a short time only. Indeed, one would prefer to have (18) as large as possible.

   Recall that letting $\tau = 1$ in (18), one obtains expression (19).

   Similarly the results presented at subsection 3.4, concerning the mean of the first entrance time T to the set H, when starting in H^c, are useful in the design of this type of threshold queues. This mean gives indication of a tendency towards exiting H^c.

5. The mathematical treatment of the problems discussed in section 2 would require time-dependent equations for the transition probabilities of states. The complexity of such systems makes analysis virtually impossible. Even in Markov processes with discrete state space, Kolmogorov time-dependent equations are rarely solved explicitely, and various transforms of required functions are used in order to obtain some partial information.
Even in the case of a birth-and-death process these equations are analytically intractable. For example, this happens precisely in the time-dependent version of the M/M/1 queue with intensities $\lambda(t)$ for input and $\mu(t)$ for service. The situation is more complex if the boundary conditions are imposed (like finite waiting room) or if X_t are random vectors describing several quantities. Solving such equations by numerical methods maybe possible, but is not treated here. (The fact that the system considered here has simple structure is the additional advantage, but analysis is complicated by the two-dimensional discussion of hyteresis loop and the boundary conditions).
We remark that Green and Kolesar (1991) for the 1-server queue and Taafe and Ong (1987) for the Ph(t)/M(t)/c/s queue were successful in developing numerical approximation methods to calculate the time-dependent distribution of the number in the system through the time-dependent Kolmogorov equations.

6. The queueing model studied in this paper is a generalization of the (traditional) M|M|c queue. Intuitively, it is expected that as b approaches ∞ the M|M|c queue under rule-1 approaches the M|M|c queue. Let us now briefly present an analytical comparison between these two models, in terms of their first moments. The equilibrium probability that there are j customers in the system, is now denoted by P_j, as usually done in the queueing literature. For the sake of simplicity, we consider b-a = 1. In this case the expressions in subsection 3.2 simplifies as follows:

$$P_j = \begin{cases} 1/j!\ (\lambda/\mu)^j\ P_0, & 0 \le j \le c-1 \\ 1/c!\ (\lambda/\mu)^c\ (\lambda/c\mu)^{j-c}\ P_0, & c \le j < a \\ 1/c!\ (\lambda/\mu)^c\ (\lambda/c\mu)^{a-c}\ (\lambda/(c+k)\mu)^{j-a}\ P_0, & j \ge a\ , \end{cases} \tag{33}$$

where P_0 is obtained by the condition $\sum_{i=0}^{\infty} P_i = 1$. When $\lambda \neq c\mu$

$$P_0 = \Bigg[\sum_{j=0}^{c-1} (\lambda/\mu)^j/j! + 1/c!(\lambda/\mu)^c \left(1-(\lambda/c\mu)^{a-c}\right)/\left(1-(\lambda/c\mu)\right) + \\ + 1/c!(\lambda/\mu)^c\ (\lambda/c\mu)^{a-c} \left(1/\left(1-(\lambda/(c+k)\mu)\right)\right) \Bigg]^{-1}. \tag{34}$$

By assumption $\lambda < (c+k)\mu$, therefore $\lambda = c\mu$ ($\tau = 1$) is allowed, and this case has to be considered separately.

$$P_j = \begin{cases} 1/j!\ c^j\ P_0, & 0 \le j \le c-1 \\ 1/c!\ c^c\ P_0, & c \le j < a \\ 1/c!\ c^c\ (c/(c+k))^{j-a}\ P_0, & j \ge a \end{cases} \tag{35}$$

$$P_0^{-1} = \sum_{j=0}^{c-1} c^j/j! + \sum_{j=c}^{a-1} 1/c!c^c + 1/c!c^c \sum_{j=a}^{\infty} (c/(c+k))^{j-a}$$

$$= \sum_{j=0}^{c-1} c^j/j! + c^c/c!\ [(a-c) + (c+k)/k], \tag{36}$$

When $c = 1$ and $k = 1$

$$P_0^{-1} = a+2. \tag{37}$$

For the M|M|c queue, let the equilibrium probability that the system is empty be denoted by P_0^*.

When $\lambda < c\mu$

$$P_0^* = \left[\sum_{j=0}^{c-1} 1/j!(\lambda/\mu)^j + 1/c!(\lambda/\mu)^c\ c\mu/(c\mu - \lambda) \right]^{-1}. \tag{38}$$

See, for instance, Gross and Harris, (1985) p.87, for details.

For the M|M|c queue the mean number in the queue, say $E[Q_1]$, is

$$E[Q_1] = \sum_{j=c}^{\infty} (j - c)P_j = \left[((\lambda/\mu)^c\ \lambda\mu)/((c - 1)\ !(c\mu - \lambda)^2) \right] P_0^* \tag{39}$$

See, for instance, Gross and Harris (1985), p.87, for details.

For the M|M|c queue under rule-1, let $E[Q_2]$ and $E[S_2]$ represent its mean number in the queue and mean number in the system, respectively. Obviously, when $b - a = 1$

$$E[Q_2] = \sum_{j=c}^{a} (j - c)\ P_j + \sum_{j=a+1}^{\infty} (j - (c + k))\ P_j \tag{40}$$

When $b - a = 1$ and $\lambda \neq c\mu$

$$E[Q_2] = P_0\ \lambda/c!\ (\lambda/\mu)^c\ (\lambda/c\mu)^{a-c} \left\{ (a - c) \left[1/((c + k)\mu - \lambda) - 1/(c\mu - \lambda) \right] - (k - 1) \right.$$
$$\left. \left[1/(c + k)\mu - \lambda \right] + c\ \mu/(c\mu - \lambda)^2 \left[(c\mu - \lambda)^{a-c} - 1 \right] + \lambda/((c + k)\mu - \lambda)^2 \right\}. \tag{41}$$

When $b - a = 1$ and $\lambda = c\mu$, by (35)

$$E[Q_2] = P_0 \left[\sum_{j=c}^{a} (j - c)/c!c^c + \sum_{j=a+1}^{\infty} (j - c - k)/c!c^c (c/(c + k))^{j-a} \right]$$
$$= P_0\ c^c/c! \left[(a - c)(a - c + 1)/2 + c \left[(a - c) - (k - 1) \right]/k + (c/k)^2 \right]. \tag{42}$$

When $b - a = 1$ and $c = 1$ and $k = 1$

$$E[Q_2] = P_0\ a(a + 1)/2 = a\ (a + 1)/2(a + 2). \tag{43}$$

Let $E[W_2]$ and $E[T_2]$ be the mean waiting time in the queue and the mean sojourn time in the system, respectively, for the M|M|c queue under rule-1. Quite clearly, $E[T_2] = 1/\mu + E[W_2]$ and

by the Little's formulae (that is still valid for the M|M|c queue under rule-1) $E[S_2] = \lambda E[T_2] = \lambda/\mu + E[Q_2]$. For details on the Little's formula, see, for instance, Ramalhoto et al (1983).

Wheb $\lambda < c\mu$, from (34) and (38) it is very easy to prove that

$$\lim_{a\to\infty} P_0 = P_0^*. \tag{44}$$

From (39), (41) and (44), and after straightforward calculation it is easy to prove that

$$\lim_{a\to\infty} E[Q_2] = E[Q_1], \tag{45}$$

similar limits could be found for $E[S_2]$, $E[W_2]$, $E[T_2]$.

Final conclusions

A framework to deal with quality improvement in the queueing systems of the service industry is set up in ad-hoc fashion. It is studied a queue-design to cope with temporary peak duration which incorporates some of those queueing quality dimensions. This queue-design is eseentially defined through the so called rule-1, that can be attached to any usual queue type and basically introduces two boundaries a and b, with a < b, that can be regarded in terms of the queueing quality improvement as prevention or alarm line and action line, respectively. Exact analytical results are provided for the M/M/c queue under rule-1. For the $M_t/M_t/c$ queue under rule-1 numerical approximations and simulation studies are expected to be possible but are not treated here.

Clearly, in complex real life situations, the rule-1 introduced, on its own accounts for many strategic decisions to be taken, some of them are short-term, other mid-term and long-term decisions. They are not always easy to implement. The education and training of the servers to improve their reliability and to improve the serviceability and performance of the queueing system are examples of mid-term strategies. A typical short-term decision would be, diverting other personnel to meet short term peaks in the service requirements. The prevention or alarm line, boundary a, would then alert them for this eventuality and the action line, boundary b, would put them into action. The results provided in subsection 3.3 and 3.4 are expected to be useful in setting up such strategic decisions.

In conclusion, quality programs to improve the product-service has to be created and implemented. Measures are the key to any quality program. This paper described the quality dimensions-performance, flexibility, serviceability, reliability, courtesy and appearance. It identifies what the key measures are to track quality in the product-service for the first four quality dimensions. It provides a control rule that can be added to any traditional queueing model to increase its flexibility, and also illustrates the type of queue-design that includes quality aspects. The exact analytical solutions given for the Markovian case are the basis for further research into numerical methods and quality analysis and cost control of more realistic situations. That is illustrated by the following research projects.

List of Research Projects:

(1) The model studied,in steady state, has six basic parameters: a, which defines the prevention or alarm line; b, which defines the action line; c, the number of regular servers; k, the number of extra diverted servers; λ, the arrival rate and μ, the service rate of each server. Most of those parameters will be fixed through managerial needs and limitations, but some others can be obtained by maximizing queueing quantities of paramount importance for the queueing quality improvement, and can be selected from some of the relations provided. For instance, the relation (18) gives $D_{(b,c+k)} = P(T < \infty, T_1 < T_2 \mid X_0 = b, Y_0 = c+k)$ the entrance probability to H_1 (before H_2) when starting at (b,c+k) in H^c, as a simple function of ρ, c and (b-a). The study of all the steady-state analytical results, presented in subsections 3.2 to 3.4, in terms of the basic parameters and associated cost functions, is in itself an interesting research project.

(2) In the service industry, the customers not always see the servers. Nevertheless, even when the customers do see the servers, if the duration of the peak is kept short and if it is clear for the customers that the queueing system is doing well and the service times remain reasonable, the withdraw of the k servers, all of a sudden, is not necessarily badly received by the customers. On the contrary, it can be regarded as a way to provide good service at a reasonable price. However, in some cases, the sucessful implementation of our queue-design demands a careful study of the basic parameters, specially parameters a and b, costs involved, quality and price of the product-supply, as well as, to conduct customer's opinion surveys, to be sure that the negative psychological feeling of seeing the withdraw of the extra k servers has been overcome. That is, of course, a research project of a more multidisciplinary nature.

(3) In most situations, the steady-state analysis is not the most appropriated one, when dealing with the peak duration of a queueing system of a service industry. The numerical evaluation of time-dependent distribution of the number in the system, for the $M_t/M_t/c$ queue under rule-1, through numerical solution, to the Kolmogorov system of time-dependent differential equations, for each given shape of $\lambda(t)$ and $\mu(t)$, respectively, is in itself an interesting research project. The steady-state results given in the previous section, still can help to establish first convenient approximations for the non-fixed parameters as mentioned above. It also provides a way to check the numerical solution of the time-dependent differential equations.

(4) In some real-life situations, the "unspent" service times at instants when the number of servers changes need to be considered (if $a > 0$, split or redundant service must occur). This is another problem and preservation of the Markov property must be proved. In general, the steady-state results presented, in this paper, can only approximate the true limiting results of this other problem.

(5) The queueing quality improvement approach proposed in this paper, never ends. It demands a continuous improvement in the queue design to fulfill the first four queueing quality dimensions; and a jointly approach to short-term, mid-term and long-term strategies. For instance, the short-term strategy considered above (by using rule-1) can on its own greatly increase the arrival rate which could lead to degradation of all the six queueing quality dimensions, if the manager has not anticipated this possibility and has not a strategy to cope with it, as part of a continuous improvement plan prepared for different scenarios. Like in modern manufacturing systems, looking for "quality" from the begining - the design - and in all sectors - reliable servers, customer's needs, product-supply, customer's delight, etc. - and being prepared to re-design the system whenever necessary (rather than through inspection, as in the old days of manufacturing systems), is a must in queueing system's quality research. That leads to interesting TQM research projects for the queueing systems of the service industry.

The TQM approach seems to be one of the quite reasonable approaches to improve quality in the queueing systems of the service industry. Furthermore, as Box (1994) stresses, the so-called "quality revolution" is about more than quality. It concerns knowledge. By learning more about the product, the process and the customers, we can do a better job whether we are making printed circuits in a factory or admitting patients to a hospital.

The six queueing quality dimensions defined, are the basis to set up a quality analysis and cost control index, for the queueing systems of the service industry. This index might also be used to assess the progress of our learning queueing system.

The authors believe that the queueing theorists have an important role to play in the quality improvement of the product-service of the service industry. A new epoch for queueing theory, in relation to the "quality revolution", is emerging today.

Acknowledgments

Section 1 and 2 have benefited from the discussions that the first author held with Professor Bergman and his research group during her visit to Linköping University.

References

[1] Bergman, B. and Klefsjö, B., *Quality from Customer Needs to Customer Satisfaction*, Studentliteratur (1994).
[2] Box, G., *Statistical and Quality Improvement*, J.Royal Statist.Soc.A 157 Part 2 (1994) 209-229.
[3] Deming, E., *Out of the Crisis*, Cambridge University Press, Massachusetts Institute of Technology, Massachusetts (1986).
[4] Deming, E., *The New Economics for Industry, Government and Education*, MIT Center for Advanced Engineering Study, Massachusets (1993).
[5] Feller, W., *An Introduction to Probability and its Applications*, Vol. I - 3rd edition and Vol. II - 2nd edition, Wiley (1968; 1971).
[6] Garvin, D.A., *Managing Quality*, The Free Press (1988).
[7] Green, L.V. and Kolesar, P.J., *The Pointwise Stationary Approximation for Queues with Nonstationary Arrivals*, Management Science 37 (1991) 84-97.
[8] Gross, D. and Harris, C., *Fundamentals of Queueing Theory*, 2nd edition, Wiley (1985).
[9] Hall, R.W., *Queueing Methods for Services and Manufacturing*, Prentice Hall (1991).

[10] Ibe, O.C. and Keilson, J., *Threshold Queues with Multiple Identical Servers and Hysteresis*, GTE Laboratories TM 0511-07092-414-06 (1992).
[11] Larson, R., *Perspectives on Queues: Social Justice and the Psychology of Queueing*, Operations Research 35 (1987) 895-905.
[12] Larson, R., *There's More to a Line than its Wait*, MIT Technology Review (1988) 60-67.
[13] Maister, *The Psychology of Waiting Lines*, in The Service Encounter: Managing Employee/Customer Interaction in Service Businesses, eds: Czepiel, Solomon and Suprenant, Lexington Books (1985).
[14] Neuts, M.F. and Rao, B.M., *On the Design of a Finite-Capacity Queue with Phase-Type Service Times and Hysteretic Control*, European Journal of Operational Research 62 (1992) 221-240.
[15] Osuna, *The Psychological Cost of Waiting*, Journal of Mathematical Psychology 29 (1985) 82-105.
[16] Ramalhoto, M.F., *Some Inventory Control Concepts in the Control of Queues*, in Modelling and Stimulation, Eds. Vogt, W.C. and Mickie, M.H., University of Pittsburgh Press, Vol. 22 (1991) 639-647.
[17] Ramalhoto, M.F., Amaral, J.A. and Cochito, M.T., *A survey of the Little's Formula*, International Statistical Review 51 (1983) 255-278.
[18] Ramalhoto, M.F., Nunes, C., Morais, M. and Silvia, R., *Estudo de uma Fila de Espera com Número Variável de Servidores*, Actas da 1ª Conferência em Estatística e Optimização, Troia, Portugal (1990) 281-300.
[19] Romani, J., *Un Modelo de la Teoria de Colas con Numero Variable de Canales*, Trabajos Estadestica 8 (1957) 175-189.
[20] Rosander, A.C., *Applications of Quality Control in Service Industries*, Milwaukee, WI: ASQC Press (1985).
[21] Syski, R., *Passage Time for Markov Chains*, IOS Press, Amsterdam, Holland (1992).
[22] Taafe, M.R. and Ong, K.L., *Approximating Nonstationary Ph(t)/M(t)/s/c Queueing Systems*, Annals of Operations Research 8 (1987) 103-116.
[23] Zeithami, V.A., Parasuraman, A. and Berry, L.L., *Delivering Quality Service*, The Free Press, NY (1990).

INVESTIGAÇÃO OPERACIONAL, vol. 16 (Dezembro 1996) 173-179

A COMPARISON BETWEEN LINE SEARCHES AND TRUST REGIONS FOR NONLINEAR OPTIMIZATION

Luís N. Vicente

Departamento de Matemática
Universidade de Coimbra
3000 Coimbra - Portugal

Abstract

Line searches and trust regions are two techniques to globalize nonlinear optimization algorithms. We claim that the trust-region technique has built-in an appropriate regularization of ill-conditioned second-order approximation. The question we ask and then answer in this short paper supports this claim. We force the trust-region technique to act like a line search and we accomplish this by always choosing the step along the quasi-Newton direction. We obtain global convergence to a stationary point as long as the condition number of the second-order approximation is uniformly bounded, a condition that is required in line seacrhes but not in trust regions.

Resumo

A pesquisa unidimensional e as regiões de confiança são técnicas de globalização de algoritmos para optimização não linear. A técnica de regiões de confiança incorpora também a regularização de aproximações de segunda ordem mal condicionadas. Neste artigo é discutida esta regularização numa situação em que a técnica de regiões de confiança é forçada a actuar como a pesquisa unidimensional, ao exigir-se que o passo sejam sempre na direcção de quasi-Newton. Neste caso, a convergência global para um ponto estacionário é verdadeira desde que o número de condição da aproximação de segunda ordem seja limitado uniformemente, hipótese que tradicionalmente é assumida para a pesquisa unidimensional mas não para as regiões de confiança.

Keywords

Line searches, trust regions, quasi-Newton.

1. Framework

Consider the unconstrained minimization problem

$$\text{minimize } f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is at least continuously differentiable, and $x \in \mathbb{R}^n$.

A quasi-Newton method for the solution of (1) generates a sequence of iterates $\{x_k\}$ and steps $\{s_k\}$ such that $x_{k+1} = x_k + s_k$. At x_k, a quadratic model of $f(x_k + s)$,

$$\Psi_k(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s,$$

is formed, where $g_k = \nabla f(x_k)$ and H_k introduces curvature into the model. We assume that H_k is a symmetric positive definite matrix of order n. The quasi-Newton step s_k is given by $s_k = -H_k^{-1} d_k$ and hence is unconstrained minimizer of $\Psi_k(s)$. Thus a quasi-Newton method consists of forming $x_{k+1} = x_k - H_k^{-1} g_k$, for k = 0, 1,..., but it is well-known that such an algorithm is not globally convergent. If we want to start with any choice of x_0 and still guarantee convergence then we need a globalization strategy.

A line search strategy considers $-H_k^{-1} g_k$ to be a direction from which a step will be obtained. The step s_k is of the form $-\lambda_k H_k^{-1} g_k$, where the step lenght λ_k is chosen in an appropriate way.

The trust-region technique does not necessarily choose the quasi-Newton direction. Here a step is an approximate solution of the trust-region subproblem

$$\begin{aligned} &\text{minimize} && \Psi_k(s), \\ &\text{subject to} && \| s \| \leq \delta_k, \end{aligned} \tag{2}$$

where δ_k is the trust radius, and $\| \cdot \|$ denotes a norm in $\mathbb{R}^n$, assumed in this paper to be the ℓ_2 norm.

2. Line searches and trust regions

The global convergence result we are looking at is

$$\lim_{k \to +\infty} \| g_k \| = 0. \tag{3}$$

Let us describe in detail the classical conditions under which both line searches and trust regions give us (3).

If a line search is used one has to ask the step $s_k = -\lambda_k H_k^{-1} g_k$ to satisfy the Armijo-Goldstein-Wolfe conditions:

$$f(x_k + s_k) \leq f(x_k) + \alpha_1 g_k^T s_k, \tag{4}$$

$$\nabla f(x_k + s_k)^T s_k \geq \alpha_2 g_k^T s_k, \tag{5}$$

where α_1 and α_2 are constants fixed for all k and satisfying $0 < \alpha_1 < \alpha_2 < 1$. After an s_k, or equivalently a λ_k, has been found that satisfies these conditions, a new iterate x_{k+1} is formed by setting $x_{k+1} = x_k + s_k = x_k - \lambda_k H_k^{-1} g_k$. A key ingredient to obtain global convergence to a stationary point is to keep the angle $\theta_k \in [0, \frac{\pi}{2}]$ between g_k and $-H_k^{-1} g_k$ uniformly bounded away from $\pi/2$. Let $cn(H_k) = \| H_k \| \| H_k^{-1} \| \geq 1$ be the condition number of the matrix H_k. If $cn(H_k)$ is uniformly bounded, i.e., if there exists a $\nu > 1$ such that

$$cn(H_k) \leq \nu$$

for every k, then we have

$$\cos(\theta_k) = \frac{g_k^T H_k^{-1} g_k}{\|g_k\| \, \|H_k^{-1} g_k\|} \geq \frac{1}{\nu}. \tag{6}$$

The inequality (6) is proved using

$$\frac{g_k^T H_k^{-1} g_k}{\|g_k\| \, \|H_k^{-1} g_k\|} \geq \frac{\lambda_{min}(H_k^{-1}) \, \|g_k\|^2}{\|H_k^{-1}\| \, \|g_k\|^2} = \frac{1}{\lambda_{max}(H_k) \, \|H_k^{-1}\|} = \frac{1}{\|H_k\| \, \|H_k^{-1}\|},$$

where $\lambda_{min}(H_k^{-1})$ and $\lambda_{max}(H_k)$ denote the smallest and largest eigenvalues of H_k^{-1} and H_k, respectively. The lower bound (6) on $\cos(\theta_k)$ is crucial to establish the following result.

Theorem 2.1 Let f the bounded below and ∇f be uniformly continuous. If s_k satisfies (4)-(5) and the condition number $cn(H_k)$ of H_k is uniformly bounded, then $\{x_k\}$ satisfies (3).

Some of the ground work that led to this result was provided by Armijo [1] and Goldstein [7]. It was established by Wolfe [23], [24] and Zoutendijk [25], under the assumption that the

gradient is Lipschitz continuous. However this condition can be relaxed and one can see that uniform continuity is enough (see Fletcher [5], Theorem 2.5.1). Some practical line-search algorithms are described by Moré and Thuente [10]. For more references see also the books [3], [11], and [13] and the review papers [4] and [12].

Now let us describe how the trust-region technique works. A step s_k has to decrease the quadratic model $\Psi_k(s)$ from $s = 0$ to $s = s_k$. The way s_k is computed determines the magnitude of the predicted model $\Psi_k(0) - \Psi_k(s_k)$ and influences the type of global convergence of the trust-region algorithm. One can ask s_k to satisfy two classical conditions, either fraction of Cauchy decrease (simple decrease) or fraction of optimal decrease.

The first condition forces the predicted decrease to be at least as large as a fraction of the decrease given for $\Psi_k(s)$ by the Cauchy step c_k. This step is defined as the solution of the one-dimensional problem minimize $\Psi_k(s)$ subject to $\| s \| \leq \delta_k$, $s \in \text{span}\{-g_k\}$, and it is given by

$$c_k = \begin{cases} -\dfrac{\|g_k\|^2}{g_k^T H_k g_k} g_k & \text{if } \dfrac{\|g_k\|^3}{g_k^T H_k g_k} \leq \delta_k \ , \\ -\dfrac{\delta_k}{\|g_k\|} g_k & \text{otherwise.} \end{cases} \tag{7}$$

The step s_k is said to satisfy a fraction of Cauchy decrease for the trust-region subproblem (2) if

$$\Psi_k (0) - \Psi_k(s_k) \geq \beta_1 (\Psi_k(0) - \Psi_k(c_k)), \tag{8}$$

where $\beta_1 \in (0,1]$ is fixed across all iterations. Two widely used algorithms to compute steps that satisfy (8) are the dogleg algorithm ([2], [14], and [17]) and the conjugate-gradient algorithm ([20] and [22]).

The second condition is more stringent and relates the predicted decrease to the decrease given on $\Psi_k(s)$ by the optimal solution s_k^* of the trust-region subproblem (2). The step s_k is said to satisfy a fraction of optimal decrease for the trust-region subproblem (2) if

$$\Psi_k (0) - \Psi_k(s_k) \geq \beta_2 (\Psi_k(0) - \Psi_k(s_k^*)), \tag{9}$$

where $\beta_2 \in (0,1]$ is fixed across all iterations. Algorithms to compute s_k that satisfy the fraction of optimal decrease (9) have been proposed in [9] and [19]. It is a simple matter to see that (9) implies (8).

The predicted decrease $\text{pred}(s_k)$ given by s_k is defined as $\Psi_k(0) - \Psi_k(s)$. The actual decrease $\text{ared}(s_k)$ is given by $f(x_k) - f(x_k + s_k)$. The trust-region strategy relates the acceptance of s_k and the update of the trust radius with the ratio $r_k = \dfrac{\text{ared}(s_k)}{\text{pred}(s_k)}$ in the following way:

If $r_k < \eta$ then s_k is rejected, $x_{k+1} = x_k$, and $\delta_{k+1} = \gamma \| s_k \|$.

If $r_k \geq \eta$ then s_k is accepted, $x_{k+1} = x_k + s_k$, and $\delta_{k+1} \geq \delta_k$.

Here γ and η are uniformly fixed and such that $0 < \gamma, \eta < 1$. Of course the rules to update the trust radius can be much more involved, but the above suffices to prove convergence results and to understand the trust-region mehcanism.

Theorem 2.2

Let f be bounded below and ∇f be uniformly continuous. If s_k satisfies (8) and $\| H_k \|$ is uniformly bounded, then $\{x_k\}$ satisfies {3}.

In in addition, f is twice continuously differentiable and s_k satisfies (9), then $\{x_k\}$ has a limit point x_* such $\nabla^2 f(x_*)$ is positive semi-definite.

The global convergence to a stationary point was established by Powell [15] and Thomas [21]. The global convergence to a point where the Hessian is positive semi-definite was established by Sorensen [18]. Related results can be found in references [6], [8], [9], and [17]. The assumption on $\| H_k \|$ can be weakened. Powell [16] proved a convergence result in the case where there is a bound on the second-order approximation H_k that depends linearly on the iteration counter k.

3. The scaled quasi-Newton step

A major difference between the results that describe global convergence to a stationary point is that a uniform bound on H_k^{-1} is required for line searches but not for trust regions. Of course we are not making a fair comparison because the form of the step for trust regions was left unspecified whereas for line searches the step was taken along the quasi-Newton direction. In order to compare these global convergence results, let us take away the flexibility that the trust-region technique has to pick a direction and force it to move along the quasi-Newton direction. In other words the step s_k is now given by $-\xi_k H_k^{-1} g_k$, where

$$\xi_k = \begin{cases} \dfrac{\delta_k}{\|H_k^{-1} g_k\|} & \text{if } \|H_k^{-1} g_k\| > \delta_k \,, \\ 1 & \text{otherwise.} \end{cases} \qquad (10)$$

We call this step a scaled quasi-Newton step and denote it by s_k^N.

If we want to establish global convergence to a stationary point, we need to make sure that the scaled quasi-Newton step satisfies the fraction of Cauchy decrease condition (8). The natural question to ask is: under what conditions does the scaled quasi-Newton step satisfy (8)? We can go even further and ask: what do we need to assume to guarantee that such a step also satisfies the fraction of optimal decrease condition (9)?

4. Global convergence for the scaled quasi-Newton step

We prove in this section that the answer to the question formulated above is the existence of a uniform bound on the condition number of H_k.

Theorem 4.1 If the condition number $\mathbf{cn}(H_k)$ of H_k is uniformly bounded, then the scaled quasi-Newton step $s_k^N = \xi_k H_k^{-1} g_k$ satisfies the fraction of Cauchy decrease condition (8).

Proof. If $\xi_k = 1$, s_k^N is the optimal solution of the trust-region subproblem (2) and there is nothing else to prove. So, suppose that $\| H_k^{-1} g_k \| > \delta_k$. It follows from this and $\xi_k < 1$ that

$$\begin{aligned} \Psi_k(0) - \Psi_k(s_k^N) &= \frac{1}{2} \xi_k (2 - \xi_k) g_k^T H_k^{-1} g_k \\ &> \frac{1}{2} \xi_k g_k^T H_k^{-1} g_k \end{aligned}$$

$$= \frac{1}{2}\xi_k \| g_k \| \frac{g_k^T H_k^{-1} g_k}{\|g_k\| \|H_k^{-1} g_k\|} . \qquad (11)$$

According to the definition of c_k given by (7), we either have $\frac{\|g_k\|^3}{g_k^T H_k g_k} \leq \delta_k$ in which case

$$\Psi_k(0) - \Psi_k(c_k) = \frac{\|g_k\|^4}{g_k^T H_k g_k} - \frac{1}{2} \frac{\|g_k\|^4}{(g_k^T H_k g_k)^2} g_k^T H_k g_k$$
$$> \frac{1}{2} \delta_k \| g_k \|,$$

or $\frac{\|g_k\|^3}{g_k^T H_k g_k} > \delta_k$ which in turn gives

$$\Psi_k(0) - \Psi_k(c_k) = \delta_k \| g_k \| - \frac{1}{2} \frac{\delta_k^2}{\|g_k\|^2} g_k^T H_k g_k$$
$$\leq \frac{1}{2} \delta_k \| g_k \|.$$

From this, (6), and (11), we get

$$\Psi_k(0) - \Psi_k(s_k^N) \geq \frac{1}{2\nu} \left(\Psi_k(0) - \Psi_k(c_k)\right).$$

Thus s_k^N satisfies (8) with $\beta_1 = \frac{1}{2\nu}$. ♦

The following example is taken from [2] and indicates that without the uniform bound on the condition number, the scaled quasi-Newton step might not satisfy the fraction of Cauchy decrease condition.

Example 4.1 Let us drop the subscripts k and consider $H = \text{diag}(1, \varepsilon^2, \varepsilon^4)$ and $g = (\varepsilon^2, \varepsilon^2, \varepsilon^3)^T$, where ε is positive and small. With these choices we have

$$H^{-1}g = \left(\varepsilon^2, 1, \frac{1}{\varepsilon}\right)^T, \| H^{-1}g \| = \mathcal{O}\left(\frac{1}{\varepsilon}\right), g^T H^{-1} g = \mathcal{O}(\varepsilon^2), \text{ and } \frac{\|g\|^3}{g^T H g} = \mathcal{O}(\varepsilon^2).$$

Note that

$$\frac{g^T H^{-1} g}{\|g\| \|H^{-1} g\|} = \mathcal{O}(\varepsilon).$$

If δ is chosen very small, say $\delta = \mathcal{O}(\varepsilon^3)$ then by (7) and (10), $c = -\frac{\delta}{\|g\|} g$ and $\xi = \frac{\delta}{\|H^{-1} g\|}$. As a result, $\Psi(0) - \Psi(s^N) = \mathcal{O}(\varepsilon^6)$ and $\Psi(0) - \Psi(c) = \mathcal{O}(\varepsilon^5)$, which shows that as ε gets smaller and smaller the fraction of Cauchy decrease condition becomes more and more difficult to satisfy.

Theorem 4.2 If the condition number $\mathbf{cn}(H_k)$ of H_k is uniformly bounded, then the scaled quasi-Newton step $s_k^N = -\xi_k H_k^{-1} g_k$ satisfies the fraction of optimal decrease condition (9).

Proof. Again if $\xi_k = 1$, s_k^N is the optimal solution of the trust-region subproblem (2) and there is nothing else to prove. Let us assume that $\| H_k^{-1} g_k \| > \delta_k$. Since $\| s_k^* \| \leq \delta_k < \| H_k^{-1} g_k \| \leq \|H_k^{-1}\| \| g_k \|$, we have

$$\Psi_k(0) - \Psi_k(s_k^*) = -g_k^T s_k^* - \frac{1}{2} (s_k^*)^T H_k (s_k^*)$$
$$\leq \| s_k^* \| \| g_k \| + \frac{1}{2} \| s_k^* \|^2 \| H_k \|$$

$$\leq \delta_k \| g_k \| + \frac{\nu}{2} \delta_k \| g_k \|$$
$$= \left(1 + \frac{\nu}{2}\right) \delta_k \| g_k \|.$$

From this, (6), and (11), we get

$$\Psi_k(0) - \Psi_k(s_k^N) \geq \frac{1}{2} \delta_k \| g_k \| \frac{g_k^T H_k^{-1} g_k}{\|g_k\| \|H_k^{-1} g_k\|}$$
$$\leq \frac{1}{2} \frac{1}{\nu} \frac{1}{1 + \frac{\nu}{2}} \left(\Psi_k(0) - \Psi_k(s_k^*)\right)$$
$$\geq \frac{1}{2\nu + \nu^2} \left(\Psi_k(0) - \Psi_k(s_k^*)\right),$$

and we see that the scaled quasi-Newton step satisfies (9) with $\beta_2 = \frac{1}{2\nu + \nu^2}$. ♦

Theorem 2 in [2] shows that if a step satisfies the fraction of Cauchy decrease (8) and there exists a uniform bound on the condition number of H_k, then such a step also satisfies the fraction of optimal decrease condition (9). Thus we could prove Threorem 4.2 by appealing to this earlier result, in conjunction with Theorem 4.1.

5. Final remarks

There are other interesting relationships between line searches and trust regions. For instance, the criteria to accept a step are very similar. Suppose that a line search only requires the Armijo-Goldstein-Wolfe condition (4) to accept a step s_k. This condition can be rewritten as

$$\frac{f(x_k) - f(x_k + s_k)}{-g_k^T s_k} \geq \alpha_1, \qquad (12)$$

and it becomes evident how similar this is to the condition

$$\frac{f(x_k) - f(x_k + s_k)}{-g_k^T s_k - s_k^T H_k s_k} \geq \eta,$$

used in the trust-region technique. One can see that trust regions use curvature to accept or reject a step but line searches do not. However many practical implementations of line searches include second-order information in the sufficient decrease condition (4), or (12).

One final comment about the regularization issue is in order. It is also possible to regularize a line search by adding to H_k a positive multiple μI of the identity matrix. Of course one must choose μ and this becomes a performance issue that does not arise in trust regions. The solution s_k^* of the trust-region subproblem (2) satisfies the first-order necessary optimality conditions

$$\left(H_k + \mu I\right) s_k^* = -g_k,$$
$$\mu\left(\delta_k - \| s_k^* \|\right) = 0,$$
$$\mu \geq 0, \| s_k^* \| \leq \delta_k.$$

Here the parameter μ is implicitly defined by the size of the trust-region radius δ_k.

Acknowledgments

The author would like to thank John Dennis, Mahmoud El-Alem, Mathias Heinkenschloss, Richard Tapia, and Virginia Torczon for many interesting discussions on the topic of this paper.

References

[1] Armijo, L., *Minimization of functions having Lipschitz-continuous first partial derivates*, Pacific J. Math. 16 (1966) 1-3.

[2] Byrd, R.H., Schnabel, R.B. and Shultz, G.A, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming 40 (1988) 247-263.

[3] Dennis, J.E. and Schnabel, R.B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey (1983).

[4] Dennis, J.E. and Schnabel, R.B., *A view of unconstrained optimization*, In G.L. Nemhauser, A.H.G. Rinooy Kan, and M.J. Todd, editors, Handbooks in Operations and Management Science, North Holland, Amsterdam (1988).

[5] Fletcher, R., *Practical Methods of Optimization*, John Wiley & Sons, Chichester, second edition (1987).

[6] Gay, D.M., *Computing optimal locally constrained steps*, SIAM J.Sci.Statist.Comput.2 (1981) 186-197.

[7] Goldstein, A.A., *On steepest descent*, SIAM J.Control Optim. 3 (1965) 147-151.

[8] Moré, J.J., *Recent developments in algorithms and software for trust regions methods*, In A. Bachem, M. Grotschel, and B. Korte, editors, Mathematical programming, The state of art, pages 258-287, Springer Verlag, New York (1983).

[9] Moré, J.J. and Sorensen, D.C., *Computing a trust region step*, SIAM J.Sci.Statist.Comput.4 (1983) 553-572.

[10] Moré, J.J. and Thuente, D., *Line search algorithms with guaranteed sufficient decrease*, ACM Trans.Math.Software 20 (1994) 286-307.

[11] Nash, S.G. and Sofer, A., *Linear and Nonlinear Programming*, McGraw-Hill, New York (1996).

[12] Nocedal, J., *Theory of algorithms for unconstrained optimization*, Acta Numerica, (1992) 199-242.

[13] Ortega, J.M. and Rheinboldt, W.C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York (1970).

[14] Powell, M.J.D., *A new algorithm for unconstrained optimization*, In J.B.Rosen, O.L. Mangasarian, and K. Ritter, editors, Nonlinear Programming, Academic Press, New York (1970).

[15] Powell, M.J.D., *Convergence properties of a class of minimization algorithms*, In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, Nonlinear Programming 2 (1975) 1-27, Academic Press, New York.

[16] Powell, M.J.D, *On the global convergence of trust region algorithms for unconstrained minimization*, Math.Programming 29 (1984) 297-303.

[17] Schultz, G.A, Schnabel, R.B. and Byrd, R.H., *A family of trust-region based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J.Numer.Anal. 22 (1985) 47-67.

[18] Sorensen, D.C., *Newton's method with a model trust region modification*, SIAM J. Numer.Anal. 19 (1982) 409-426.

[19] Sorensen, D.C., *Minimization of a large scale quadratic function subject to an ellipsoidal constraint*, Technical Report TR94-27, Department of Computational and Applied Mathematics, Rice University (1994).

[20] Steihaug, T., *The conjugate gradient method and trust regions in large sclae optimization*, SIAM J.Numer.Anal. 20 (1983) 626-637.

[21] Thomas, S.W., *Sequential Estimation Techniques for Quasi-Newton Algorithms*, PhD thesis, Cornell University, Ithaca, New York (1975).

[22] Toint, Ph. L., *Towards an efficient sparsity exploiting Newton method for minimization*, In I.S. Duff, editor, Sparse Matrices and Their Uses, pages 57-87, Academic Press, New York (1981).

[23] Wolfe, P., *Convergent conditions for ascent methods*, SIAM Rev. 11 (1969) 226-235.

[24] Wolfe, P., *Convergent conditions for ascent methods. II: Some correstions*, SIAM Rev.13 (1971) 185-188.

[25] Zoutendijk, G., *Nonlinear Programming, Computational Methods*, In J. Abadie, editor, Integer and Nonlinear Programming, pages 37-86, North-Holland, Amsterdam (1970).

INVESTIGAÇÃO OPERACIONAL, vol. 16 (Dezembro 1996) 181-199

O PROCESSAMENTO PARALELO E O APOIO MULTICRITÉRIO À DECISÃO: ALGUMAS EXPERIÊNCIAS COMPUTACIONAIS

Luis M.C. Dias

João P.Costa

João N. Clímaco

Faculdade de Economia
Universidade de Coimbra
Av. Dias da Silva
3000 Coimbra - Portugal

INESC
Rua Antero de Quental, 199
3000 Coimbra - Portugal

Abstract

As parallel computers become more popular and affordable, they have begun to be used in an increasing diversity of application areas. This paper discusses the application of parallel processing to solve the computational problems that may appear in multicriteria decision aid, focusing on the situations where the set of alternatives is finite. The main results of some experiments with parallel programs are presented. These results show that under some circumstances the response time of a decision support program can be usefully reduced.

Resumo

À medida que os computadores paralelos se tornam mais disseminados e acessíveis, a sua utilização começa a estender-se a uma variedade cada vez maior de áreas de aplicação. Considera-se neste artigo a aplicação de processamento paralelo aos problemas computacionais que se colocam em situações de apoio multicritério à decisão quando o número de alternativas é finito. Face a algumas experiências computacionais efectuadas verifica-se que, em determinadas situações, o uso de um computador paralelo permite reduzir apreciavelmente o tempo de resposta de um sistema de apoio à decisão às solicitações dos utilizadores.

Keywords

Multicriteria analysis, parallel processing, decision support systems, decision aid.

1. Introdução

No âmbito do apoio multicritério à decisão têm sido propostas diversas abordagens. Uma classificação possível para essas abordagens é a utilizada em [24] (semelhante às propostas em [30] e [32]), que distingue três classes de contornos difusos: a abordagem do critério único de síntese excluindo incomparabilidade, a da relação de prevalência de síntese aceitando incomparabilidade e a do julgamento local interactivo com iterações de tentativa e erro. As duas

primeiras são geralmente dedicadas a julgamentos sobre um conjunto não demasiado grande de alternativas (acções potenciais) definido por enumeração explícita, enquanto a terceira se destina em geral a decisões sobre um conjunto de alternativas (soluções) definido por via analítica.

As metodologias dedicadas a situações em que as alternativas são definidas por via analítica recorrem habitualmente à programação matemática, como na programação linear multi-objectivo. Neste âmbito, o processamento paralelo tem sido utilizado para resolver os problemas auxiliares de optimização mono-objectivo (algumas referências podem encontrar-se em [3]; veja-se também [14]) e apresenta potencialidades para permitir a exploração simultânea, por caminhos diferentes, da região eficiente do espaço das soluções [7].

No que respeita às metodologias dedicadas a situações em que as alternativas são definidas por enumeração (metodologias multiatributo), a aplicação do processamento paralelo não é tão visível, constituindo o objecto deste trabalho. Este artigo aborda a utilização de um computador paralelo para a execução de métodos de apoio multicritério à decisão, apresentando um estudo de paralelização de dois conhecidos métodos de prevalência, o PROMETHEE e o ELECTRE III, descrito em pormenor em [9] e [10], respectivamente. Pretendeu-se sobretudo avaliar até que ponto, e em que situações, é vantajoso utilizar um computador paralelo para executar os métodos referidos.

A secção seguinte apresenta as motivações subjacentes à paralelização de metodologias de agregação multicritério, seguindo-se uma descrição do ambiente computacional utilizado neste trabalho. A secção 4 refere, para os métodos de agregação por um critério de síntese e para os métodos de agregação por uma relação de prvalência de síntese, quais as principais possibilidades de paralelização existentes. Algumas dessas possibilidades são ilustradas na secção 5 com os principais resultados de experiências computacionais efectuadas com os métodos PROMETHEE e ELECTRE III. Apresenta-se por fim um sumário e conclusões.

2. Motivação

Segundo [21], a complexidade computacional das metodologias dedicadas a situações em que as alternativas são definidas por enumeração exclui quase sempre a resolução manual, mas não é exorbitante para um vulgar computador pessoal, quando se considera um número de critérios e de alternativas "razoável" (na ordem de 10 e 100, respectivamente). Torna-se, portanto, questionável a utilização para este fim de computadores de elevado desempenho como são os computadores paralelos. Sugere-se no que se segue como a utilização do processamento paralelo para executar este tipo de metodologias pode ser, afinal, considerada pertinente.

Um processo de decisão inicia-se habitualmente por uma fase de estruturação do problema. Nessa fase são identificados os vários objectivos/pontos de vista em que se baseará a análise, é escolhida a metodologia a aplicar (eventualmente uma sucessão de metodologias) e determina-se um conjunto inicial de parâmetros necessários à aplicação desta. Alguns parâmetros definem os critérios que operacionalizam cada ponto de vista, outros definem a importância de cada ponto

de vista (informação inter-critério) e alguns poderão ser puramente técnicos. A aplicação da metodologia com esses parâmetros produz então um resultado inicial.

A aceitação do resultado inicial pelos intervenientes no processo de decisão reveste-se frequentemente de algumas dificuldades. A primeira deriva da necessidadè de confiança nos resultados sentida pelos decisores, e/ou da necessidade de justificar e defender as recomendações quando o estudo é feito em nome de outrem (decisor final). A segunda respeita ao desconhecimento, por parte dos intervenientes, da influência de alguns parâmetros no âmbito da metodologia escolhida. Por fim, uma terceira dificuldade é a sentida pelos decisores na resposta a algumas questões sobre os seus valores, mormente as relacionadas com a importância a atribuir a cada ponto de vista.

A presença destas dificuldades pode ser minorada através de uma análise de robustez aos resultados, em que se afere como se altera o resultado fornecido pelo método face à utilização de outros jogos de valores aceitáveis para os parâmetros de entrada. Naturalmente, quanto mais se manifestarem as dificuldades referidas, mais exaustiva deverá ser esta análise. Porém, esta poderá ser muito demorada, sobretudo se houver vários intervenientes, cada um dos quais com experiências de alteração de parâmetros a propor. Por esse motivo, será indispensável recorrer a meios computacionais que assegurem tempos de resposta curtos às solicitações dos vários internvenientes, ou seja, computadores rápidos.

A necessidade de respostas rápidas por parte de um computador depende da actividade em curso no âmbito do processo de decisão. Poucos se importarão se o computador levar algum tempo na obtenção de um primeiro resultado. Afinal, os intervenientes no processo de decisão terão em princípio participado anteriormente numa fase de estruturação relativamente longa. No entanto, a rapidez é crucial para uma análise de robustez interactiva. Se o computador demorar a fornecer um resultado sempre que se alterar um parâmetro nessa análise, então os intervenientes não se sentirão encorajados a ser exaustivos nesta importante fase. Poderão ficar impacientes e desistir precocemente, ressentindo-se a confiança nos resultados, a qualidade da decisão e a satisfação com o processo de decisão. A importância da rapidez será tanto maior quanto o número de intervenientes e o "custo de oportunidade" do seu tempo.

O processamento paralelo, sendo uma tecnologia que permite que vários processadores cooperem na execução de uma tarefa computacional, constituindo sistemas computacionais com um rácio desempenho/custo muito favorável, poderá ser uma resposta a esta necessidade de rapidez. Também importante é o facto de os tempos de execução poderem permanecer toleráveis à medida que o tamanho do problema aumenta, conforme a interpretação dada por Gustafson à lei de Amdahl [13]. Por fim, a crescente popularidade e vulgarização dos computadores paralelos implica que seja cada vez mais oportuna a familiarização do comunidade da Investigação Operacional com as técnicas de processamento paralelo [3].

3. Ambiente computacional

Na actualidade o processamento paralelo é, mais que uma moda ou uma curiosidade, parte integrante do *mainstream* das ciências da computação e uma tecnologia apoiada por muitas das principais empresas do ramo das tecnologias de informação. Contudo, o processamento paralelo envolve uma complexidade muito superior ao clássico compromisso entre portabilidade e eficiência do código, que se tornam objectivos ainda mais conlfituosos.

Apresenta-se na Fig.1 um esquema possível para classificar as várias arquitecturas existentes, inspirado na taxinomia usada por [11]. O computador paralelo utilizado para este trabalho foi um Parsytec MC-3/DE de arquitectura MIMD. A expressão MIMD (Multiple Instruction Multiple Data), introduzida em [12], designa computadores que podem processar em paralelo múltiplos fluxos de instruções (programas), cada um operando sobre um fluxo de dados distinto. Os computadores MIMD, também denominados multiprocessadores, distinguem-se consoante o meio utilizado para a comunicação entre processadores. Nos MIMD de memória partilhada a comunicação processa-se através de leitura/escrita em memória de acesso comum, constituindo o tipo de computadores paralelos mais em voga na actualidade. Nos MIMD de passagem de mensagens (por vezes denominados multicomputadores) a comunicação processa-se através de canais físicos de ligação entre os processadores - cada processador só pode endereçar a sua memória local, que é privada. Estes computadores são em geral mais escaláveis, embora mais difíceis de programar.

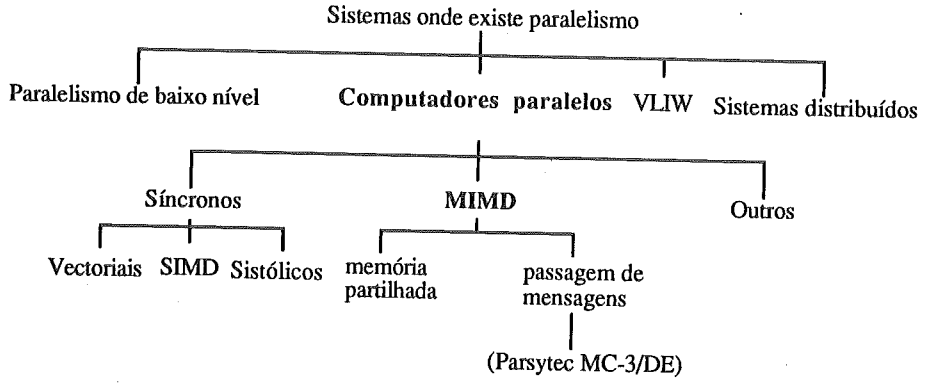


Fig. 1 - Esquema de classificação

O multicomputador utilizado possui dezasseis processadores ligados por uma topologia de rede matricial (Fig.2). Cada nó desta topologia é composto por um processador de 32 bits com unidade de vírgula flutuante integrada, o transputer T805/30 [15], e por oito Mbytes de memória RAM privada. O desempenho de cada processador cifra-se em 30 MIPS (4,3 MFLOPS). Os processadores podem comunicar em modo síncrono com um máximo de quatro vizinhos à velocidade de 20 Mbits/s por ligação, através de ligações série bidireccionais.

A interacção entre o programador ou o utilizador com o computador paralelo efectua-se por intermédio de uma estação de trabalho SUN (hospedeiro), que corre o sistema operativo UNIX. Esta é utilizada como ambiente de desenvolvimento de novos programas e como meio de comunicação de dados entre o multicomputador e o exterior (um disco rígido, um écran ou outro dispositivo de entrada/saída da SUN). A SUN está ligada a um dos transputers (que designaremos "processador de interface") por intermédio de uma placa BBK-S4 que oferece quatro ligações físicas com velocidade idêntica à das ligações entre transputers (20 Mbits/s).

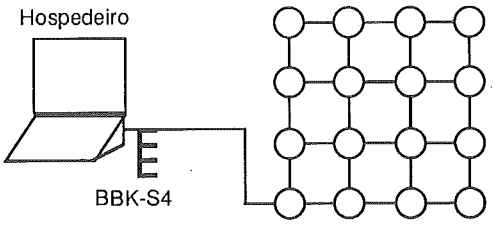


Fig. 2 - Configuração do ambiente de desenvolvimento

A gestão dos recursos do computador paralelo é efectuada pelo software PARIX [20], uma extensão do UNIX. O modelo de programação sugerido pelo PARIX consiste na execução em paralelo de vários processos da aplicação, um em cada processador. Cada um destes processos poderá conter "fios de processamento" a partilhar o tempo do processador em concorrência (pseudo-paralelismo). A principal utilização dos fios de processamento está relacionada com a comunicação entre processadores. Os fios de processamento no ambiente PARIX permitem o tratamento diferenciado de múltiplos eventos que podem surgir. Por exemplo, um processo possuirá tipicamente um fio de processamento por canal de comunicação. Desta forma, quando se executa uma instrução de comunicação que cause um bloqueio (enquanto a outra parte não estiver pronta a comunicar), só ficará bloqueado um fio de processamento e os restantes fios do processo poderão ser executados.

O ciclo de desenvolvimento de uma aplicação tem início no hospedeiro, no qual se editam os ficheiros de texto contendo o código em linguagem C (ANSI C). O compilador utilizado para obter um ficheiro executável é o ACE EXPERT [1,2].

4. Perspectivas de paralelização de metodologias de agregação multicritério

Quando estiverem definidos os critérios e as alternativas a considerar, pode construir-se uma matriz de decisão se o conjunto das alternativas for finito. Essa matriz é definida para m alternativas $(a_1,\ldots, a_m)$ e n critérios $(g_1,\ldots, g_n)$. A matriz tem dimensão m × n e os seus elementos são os desempenhos de cada alternativa segundo cada critério:

$$G_{ij} = g_j(a_i),\ i = 1,\ldots, m \quad e \quad j = 1,\ldots, n.$$

O problema da síntese da informação contida nesta matriz recorrendo, habitualmente, a informação de natureza inter-critério constitui a chamada agregação multicritério. Discute-se nesta secção como a resolução deste problema poderá beneficiar da utilização de processamento paralelo, no que respeita às duas famílias de metodologia de agregação.

4.1 Agregação por um critério de síntese

Nesta abordagem associa-se a cada alternativa um valor global $V(g_1, g_2,\ldots, g_n)$ que agrega todos os critérios $g_1\ g_2,\ldots, g_n$ considerados. A filosofia subjacente às metodologias que se inserem nesta abordagem é a de que o decisor tenta maximizar a função V(.), que constitui assim um critério de síntese. Segundo este critério uma alternativa a_1 é preferível a outra alternativa a_2 ($a_1\ P\ a_2$) se $V(a_1) > V(a_2)$ e só são indiferentes ($a_1\ I\ a_2$) se $V(a_1) = V(a_2)$.

É neste contexto que se insere um dos métodos mais utilizados para agregar os desempenhos das alternativas segundo cada critério numa avaliação multicritério: o da soma ponderada (ou, com maior rigor, da função de valor aditiva). Seja k_j (com $k_j \geq 0$) o coeficiente de ponderação do j-ésimo critério. Então o valor global de cada alternativa a_i será dado por

$$V(a_i) = \sum_{j=1}^{n} k_j g_j(a_i), \text{ com } k_j > 0 \text{ e } \sum_{j=1}^{n} k_j = 1.$$

Após efectuar estes cálculos é fácil indicar qual a melhor alternativa recomendada pelo método a partir do valor global das alternativas, sendo igualmente simples obter uma pré-ordem completa das alternativas por ordem decrescente do valor global. A determinação do valor global de todas as alternativas e posterior ordenação é computacionalmente muito simples mesmo para centenas destas. O mesmo se aplica a outras fórmulas de agregação diferentes da aditiva. Disto decorre que a realização destes cálculos em paralelo poderá não proporcionar vantagens perceptíveis.

A simplicidade dos cálculos conducentes à determinação do valor global de cada alternativa esconde, no entanto, o problema da prévia determinação dos coeficientes de ponderação k_j, incompatível com julgamentos pouco fundados baseados na importância intuitiva de cada ponto de vista. Existem vários métodos para determinar esses coeficientes (ver [21], Cap.4), entre os quais se destacam o método de *tradeoff* [17], o processo analítico hierárquico (AHP) [29], o método UTA [16] e o método MACBETH [4]. De entre estes métodos, apenas os três últimos requerem habitualmente o uso de um computador.

Pelo método AHP podem-se obter não só os coeficientes de ponderação como também o valor de cada alternativa. Para isso é necessário, em primeiro lugar, construir uma hierarquia de critérios. O topo da hierarquia representa o critério de síntese, enquanto nos níveis sucessivamente inferiores se dispõem os critérios que tenham algum impacto num critério de nível superior. Posteriormente, é necessário preencher matrizes de comparação par-a-par em que se comparam pares de alternativas em relação a cada critério e pares de critérios em relação a critérios do nível imediatamente superior. Os coeficientes de ponderação são depois obtidos combinando os vectores próprios dessas matrizes que correspondam ao seu maior valor

próprio. De acordo com as conclusões de um estudo efectuado noutro ambiente computacional [8], a paralelização é tanto mais vantajosa quanto maior for a dimensão destas matrizes. Porém, as situações em que as matrizes são grandes possuem limtado interesse prático dado que o seu preenchimento corresponde a uma elevada quantidade de julgamentos a exigir aos decisores.

O método UTA começa por pedir aos decisores que ordenem por ordem de preferência, com eventuais *ex-aequo*, um subconjunto das alternativas. Posteriormente, através da resolução de um problema de programação linear, determina o conjunto de coeficientes k_j e a forma das funções $g_j(.)$ que melhor reconstitui a ordenação definida pelos decisores. Este método pode beneficiar dos avanços na paralelização dos algoritmos de resolução de problemas de programação linear. Porém, a dificuldade computacional do programa linear a resolver aumenta sobretudo com o número de alternativas a ordenar de um modo holístico pelos decisores.

A aplicação do método MACBETH para a determinação dos coeficientes k_j requer que os decisores comparem entre si alternativas fictícias (uma por critério) par-a-par. De seguida, determinam-se os coeficientes, para além de outra informação, através da resolução de alguns problemas de programação linear. Este método, tal como o anterior, pode beneficiar dos avanços na paralelização dos algoritmos de resolução destes problemas. Contudo, a dificuldade computacional dos programas lineares a resolver só aumenta quando o número de critérios cresce, ou seja, quando aos decisores é exigido um maior número de comparações.

Em conclusão, neste tipo de abordagem verifica-se que, para os três métodos em que o uso de um computador é mais necessário, a utilidade do processamento paralelo é limitada pelo facto do esforço computacional envolvido estar directamente relacionado com a quantidade de informação (julgamentos) a exigir aos decisores.

4.2 Agregação por uma relação de prevalência de síntese

A riqueza dos resultados da agregação por um critério de síntese deriva do requisito de muita informação coerente por parte do decisor, porventura mais informação do que aquela que o decisor pode fornecer sem se sentir inseguro. Por exemplo, a existência de um critério de síntese implica que, perante duas alternativas, se consegue imediatamente indicar qual a melhor ou concluir que são indiferentes.

Os chamados métodos de prevalência sacrificam alguma operacionalidade para não exigirem um resultado mais rico do que aquele que o decisor pode aceitar com segurança. O resultado da agregação dos desempenhos segundo os vários critérios conduz a uma relação binária, a relação de prevalência. A aplicação das metodologias desenvolvidas neste contexto divide-se em duas fases. Primeiro contrói-se uma relação de prevalência agregando todos os critérios. Numa segunda fase explora-se a relação obtida.

A relação de prevalência define-se para pares de alternativas, afirmando-se que uma alternativa a_1 prevalece sobre uma alternativa a_2 (abreviadamente a_1 S a_2) se existirem razões para considerar que a alternativa a_1 é pelo menos tão boa com a alternativa a_2. As razões que justificam, dadas duas alternativas a_1 e a_2, a afirmação de que a_1 prevalece sobre a_2, dependem

de como cada método agrega a informação presente na matriz de decisão e outra informação de natureza inter-critério. A partir dessa relação podem deduzir-se três situações de preferência: a_1 é presumivelmente melhor que a_2 se a_1 S a_2 e não se verifica a_2 S a_1; a_1 é indiferente a a_2 se a_1 S a_2 e a_2 S a_1; e a_1 é incomparável a a_2 se não se verifica a_1 S a_2 nem a_2 S a_1.

O percursor dos métodos de prevalência, o ELECTRE I [22], construía uma única relação de prevalência (como na Fig.3a) para apoiar a decisão. Mais tarde, o ELECTRE II [26] apresenta a construção de duas relações de prevalência (como na Fig.3b) que diferem na força dos argumentos exigidos para concluir se uma alternativa prevalece sobre outra. Uma exige argumentos mais fortes, enquanto a outra exige argumentos mais fracos, pelo que sempre que se verifica a primeira também se verifica a segunda. O método ELECTRE III [23] apresenta uma estratégia distinta: ao comparar uma alternativa com outra tenta indicar até que ponto a primeira prevalece sobre a segunda, em vez de tentar dictomizar entre sim e não. A relação de prevalência é difusa, quantificada por um grau de credibilidade da prevalência para cada par ordenado de alternativas (como na Fig.3c). A fase de construção pode portanto resultar num número de relações de prevalência superior a um ou mesmo indeterminado (caso difuso) [25].

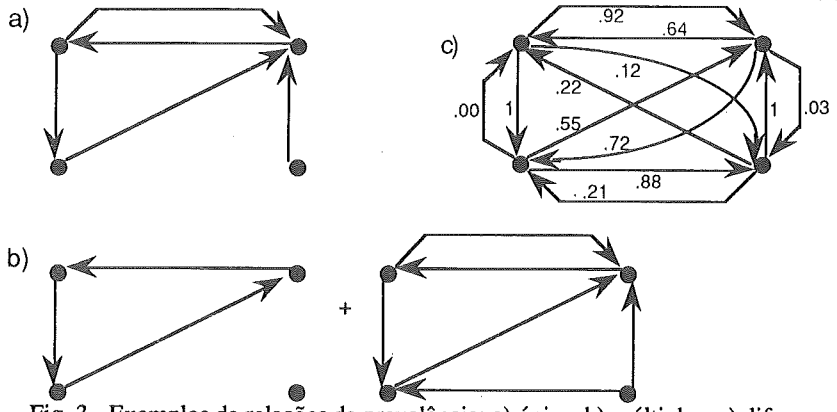


Fig. 3 - Exemplos de relações de prevalência: a) única, b) múltiplas, c) difusa

A segunda fase dos métodos de prevalência visa explorar a relação (ou relações) de prevalência obtida no apoio à decisão no âmbito da problemática em causa. Existem métodos especialmente concebidos para selecção das melhores alternativas, para ordenação das alternativas e para afectação das alternativas a categorias definidas *a priori*. Referem-se de seguida alguns dos métodos mais conhecidos, as relações de prevalência que utilizam e a problemática a que se dedicam:

- ELECTRE I [22] e ELECTRE IS [28]: relação única, selecção;
- ELECTRE II [26]: duas relações, ordenação;
- ELECTRE III [23]: relação difusa, ordenação;
- ELECTRE IV [27]: cinco relações, ordenação;
- PROMETHEE I e II [5,6]: relação difusa, ordenação;
- ELECTRE TRI [33]: relação difusa, afectação.

- Outros (veja-se p.ex. [21]).

Uma possibilidade de paralelização presente nos métodos de prevalência é a característica de, na fase de construção da relação de prevalência, estes métodos avaliarem todas as alternativas par a par, tomando em consideração todos os critérios, de modo a obter uma ou mais relações de prevalência. Esta possibilidade resulta da independência entre as diversas avaliações de pares de alternativas, isto é, dadas quatro alternativas quaisquer a_1, a_2, a_3 e a_4, a avaliação do par (a_1,a_2) é independente da avaliação do par (a_3,a_4).Trata-se de uma possibilidade interessante, dado que o volume de cálculos a efectuar cresce com o quadrado do número de alternativas, podendo tornar-se muito grande. A paralelização da fase seguinte (exploração da relação obtida) já não constituirá uma possibilidade de paralelismo idêntica para todos os métodos de prevalência, dada a variedade de estratégias de exploração existente.

A secção seguinte apresenta algumas conclusões de uma experiência de paralelização dos métodos PROMETHEE e ELECTRE III, que possuem em comum a construção de relações de prevalência difusas e a dedicação à problemática da ordenação. O facto das relações serem difusas implica que se tenha de calcular na primeira fase um índice para cada par ordenado de alternativas. O facto de a problemática ser a de ordenação torna plausível o aparecimento de situações de decisão envolvendo um número grande de alternativas.

5. Experiências computacionais

O estudo efectuado sobre a paralelização dos métodos PROMETHEE e ELECTRE III seguiu uma estratégia de experimentação, com o intuito de observar quais das soluções de paralelização adoptadas proporcionariam melhores resultados, e em que circunstâncias. Para cada método construíram-se vários programas, que paralelizam sucessivamente mais etapas do algoritmo de resolução respectivo. Consegue-se deste modo detectar as etapas em que a paralelização é mais eficiente. Os dados que definem os problemas a resolver foram gerados aleatoriamente, atendendo ao significado de cada parâmetro. Apenas a memória disponível limitou a dimensão destes problemas, definida pela número de critérios e de alternativas, dado que se anteviu que o paralelismo será tanto mais interessante quanto maior for essa dimensão.

No âmbito da problemática a que se dedicam estes métodos (a da ordenação) é plausível encontrar situações em que o número de alternativas é grande. Por exemplo, no caso do método PROMETHEE, pode encontrar-se referência a uma ferramenta, denominada BANKADVISER [18], que foi utilizada numa situação de decisão com 557 alternativas. A propósito, refira-se que essa ferramenta está baseada num vulgar computador pessoal que foi considerado insuficiente para lidar esse número de alternativas, tendo os seus autores recorrido a uma versão menos exigente do método PROMETHEE.

A possibilidade de existir um conjunto com muitos critérios é, à partida, inverosímil dadas as características que esse conjunto deve possuir e face às limitações cognitivas do decisor. No entanto, dado que se desejava estudar o comportamento dos programas em tais casos, foi estipulada uma situação em que fosse plausível encontrar um grande número de critérios. Nessa

situação, o conjunto de critérios resulta da combinação dos vários pontos de vista (em pequeno número) com várias maneiras distintas de os operacionalizar. Permite-se assim um cenário no qual múltiplos intervenientes operacionalizam individualmente, sob a forma de critérios, os vários pontos de vista, fazendo intervir todos os critérios resultantes no estabelecimento de uma relação de prevalência de grupo. Pressupõe-se que os intervenientes estão de acordo em relação aos pontos de vista a considerar e em relação ao desempenho de cada alternativa, mas em desacordo sobre o papel a conferir a cada critério.

O desempenho dos programas construídos foi medido em termos absolutos e relativos. Em termos absolutos, mediu-se o tempo de execução T(n,p), que é função do tamanho n do problema e do número p de processadores utilizado. Em termos relativos mediu-se o rácio de tempos de execução (*speedup*) real, S(n,p), que relaciona o tempo de execução de uma aplicação em paralelo com o tempo de execução do melhor algoritmo sequencial que executa a mesma aplicação. Indica, assim, quantas vezes mais rápido é o programa paralelo.

5.1 Experiências relativas ao método PROMETHEE

Numa primeira fase, o método PROMETHEE (cf.[6]) agrega a informação na matriz de decisão e a informação inter-critério numa relação de prevalência difusa, definida pelos denominados índices de preferência multicritério $\pi(.,.)$. Esses índices, calculados para todos os pares ordenados das m alternativas, são guardados numa matriz quadrada de dimensão m, adiante designada por M. Na fase de exploração, este método cosiderar, para cada alternativa a_i, que a soma dos elementos de M na i-ésima linha representa a sua "força" $\Phi^+(a_i)$ e a soma dos elementos de M na i-ésima coluna representa a sua "fraqueza" $\Phi^-(a_i)$.

A variante PROMETHEE I permite obter uma pré-ordem parcial (P, I, R) das alternativas a partir dos valores calculados de $\Phi^+(.)$ e $\Phi^-(.)$: uma alternativa é preferível a outra se tiver maior força sem ter maior fraqueza ou se tiver menor fraqueza sem ter menor força; as alternativas são indiferentes se tiverem a mesma força e a mesma fraqueza; são incomparáveis caso não se verifique nenhum dos anteriores casos. A variante PROMETHEE II produz uma pré-ordem completa (P, I), ordenando as alternativas por ordem do seu "fluxo líquido" $\Phi(.)$, que se calcula subtraindo à força de cada alternativa a sua fraqueza.

O primeiro programa a ser elaborado foi, naturalmente, um programa sequencial. Este programa pretende, por um lado, ser uma referência para a validação dos resultados obtidos pelos programas paralelos e por outro lado constituir uma referência para o cálculo das medidas de desempenho. Esse programa, bem como os programas paralelos que lhe sucederam, permite que se efectue uma análise de robustez à caracterização dos vários critérios, podendo o utilizador alterar um ou vários parâmetros de cada vez.

Um primeiro programa paralelo para o PROMETHEE (pp1) foi construído a partir do programa sequencial, paralelizando apenas a obtenção dos índices de preferência multicritério que definem a relação de prevalência. Para isso, o processador de interface (P_0) divulga os dados do problema pela rede de transputers, incumbindo cada processador do cálculo de uma

parte (um "rectângulo") da matriz M. A partição dos pares ordenados de alternativas pelos vários processadores é feita de modo tão equitativo quanto possível, processando-se nestes o cálculo dos correspondentes índices de preferência em simultâneo. À medida que cada índice é calculado, este é de imediato enviado ao P_0, aproveitando a capacidade que os transputers possuem de comunicar e efectuar cálculos em paralelo. Logo que todos os índices tenham sido recolhidos pelo P_0, este realiza a fase de exploração da relação obtida através do PROMETHEE I e/ou PROMETHEE II (conforme o desejo do utilizador) em modo sequencial, enquanto os restantes processadores permanecem inactivos. No caso de o utilizador proceder a uma análise de robustez tudo se passa como no cálculo da solução inicial, mas já não será necessário divulgar aos processadores todos os dados relativos ao problema, bastando comunicar as alterações introduzidas pelo utilizador.

Um segundo programa paralelo (pp2) foi construído a partir do primeiro (pp1), de modo a permitir uma paralelização da fase de obtenção da pré-ordem parcial (P, I, R) do PROMETHEE I. Os processadores calculam em paralelo a força e a fraqueza de cada alternativa, decidindo depois, também em paralelo, que situação de preferência se aplica a cada par de alternativas. Só a fase de exploração da relação de prevalência pelo PROMETHEE II é executada em modo sequencial pelo processador P_0.

A terceira e última estratégia adoptada (pp3) foi a de paralelizar a fase de cálculo da pré-ordem completa do PROMETHEE II, para além de toda a paralelização efectuada no programa pp2 (obtenção dos índices de preferência e exploração pelo PROMETHEE I). Poder-se-ia ter elaborado um programa que paralelizasse a exploração pelo PROMETHEE II, sem contudo paralelizar a exploração pelo PROMETHEE I. No entanto, a parte mais complexa da paralelização da exploração pelo PROMETHEE I, o cálculo da força e fraqueza de cada alternativa, é necessário à exploração pelo PROMETHEE II. Por esse motivo, optou-se por elaborar o programa pp3 a partir do pp2, sem sacrificar a obtenção em paralelo da pré-ordem parcial do PROMETHEE I. Em paralelo, procede-se ao cálculo do fluxo líquido de cada alternativa e à ordenação destas, paralelizando-se assim todo o método PROMETHEE (I e II).

Os programas paralelos referidos envolvem grande quantidade de comunicação entre processadores, o que lhes degrada o desempenho. Por esse motivo, construíram-se variantes para alguns destes programas que, à custa da incapacidade de obter algumas saídas (resultados), obtêm uma redução significativa da comunicação entre processadores. Em muitas situações, é possível que o utilizador do programa deseje apenas conhecer o resultado da exploração da relação de prevalência pelo PROMETHE I e/ou II, sem necessitar de conhecer os índices de preferência multicritério, um resultado intermédio que os decisores muitas vezes não pretendem analisar. Note-se que se o número de alternativas for de algumas centenas haverá dezenas de milhar de índices de preferência, que constituirão um conjunto pouco inteligível. Construíram-se assim os programas pp2/R e pp3/R, semelhantes aos programas pp2 e pp3, respectivamente, mas diferentes no facto de não comunicarem ao P_0 os índices de preferência

multicritério, permitindo apenas conhecer as pré-ordens produzidas pelo PROMETHEE I e/ou II. Por razões análogas, construíram-se os programas pp2/RI e pp3/RI, que não comunicam ao P_0 a pré-ordem (P, I, R): apenas permitem conhecer o resultado da exploração pelo PROMETHEE II (tal como na já referida ferramenta BANKADVISER).

Foi planeado um conjunto de experiências que permitisse aferir as vantagens das paralelizações efectuadas, em diversas situações. Geraram-se para isso vinte problemas de teste, correspondentes à combinação de um número de critérios igual a 5, 10, 50 (10 pontos de vista por 5 intervenientes) ou 100 (10 pontos de vista por 10 intervenientes), com um número de alternativas igual a 5, 10, 50, 100 ou 500. Experimentou-se executar os vários programas construídos, com diferentes níveis de exigência no que respeita às saídas (relação de prevalência, PROMETHEE I e PROMETHEE II) e em três tarefas distintas: computação de uma primeira solução, computação de outra solução após alterar um parâmetro de um critério e computação de outra solução após alterar parâmetros de todos os critérios. Estas duas últimas tarefas permitem estimar limites de tempo de computação (mínimo e máximo, respectivamente) para uma iteração de análise de robustez. Os resultados referem-se sempre à utilização dos 16 processadores. Outras experiências efectuadas, mas que saem do âmbito deste artigo, são as referentes à execução dos programas paralelos com outro número de processadores (4 e 8).

Os resultados mostram que o aproveitamento do computador paralelo foi relativamente baixo enquanto se pretendeu obter os índices de preferência que definem a relação de prevalência. Nessas situações, o melhor programa foi o pp1, sempre com um *speedup* inferior a 9. Obtiveram-se os melhores resultados quando o decisor sacrifica a possibilidade de conhecer esses índices, o que permite utilizar os programas pp2/R, pp3/R, pp2/RI e pp3/RI. As necessidades dos decisores ditam o melhor programa a utilizar: sacrificando funcionalidade obtém-se maior aproveitamento da capacidade computacional e menores tempos de resposta. Nessas circunstâncias, os melhores programas, pp2/R e pp2/RI chegam a obter *speedups* superiores a 15 (o que corresponde a cerca de 95% de eficiência no uso dos processadores).

A diminuição dos tempos de execução dos problemas é maior para os problemas que implicam maiores tempos de computação no programa sequencial, ou seja, para um número elevado de critérios ou alternativas (a Fig.4 ilustra o comportamento típico dos programas /R e /RI), quando o decisor abdica de conhecer alguns resultados e quando a análise de robustez afecta vários critérios de cada vez (cf. Tab.1). Ressalve-se, porém, que mesmo quando a análise de robustez afecta só um critério os resultados são muito satisfatórios.

As implicações destes resultados para o apoio à decisão são importantes. Considere-se o caso extremo em que se pretende avaliar 500 alternativas por 100 critérios. Com tal número de alternativas é muito plausível que os decisores apenas pretendam como resultado a pré-ordem completa do PROMETHEE II. Durante a fase de análise de robustez, a alteração dos parâmetros dos critérios poderá demorar entre um mínimo de 5m 21s e um máximo de 1h 17m. Case se

utilize o programa pp2/RI o mesmo resultado surge aos utilizadores entre um mínimo de 21s e um máximo de 5 minutos.

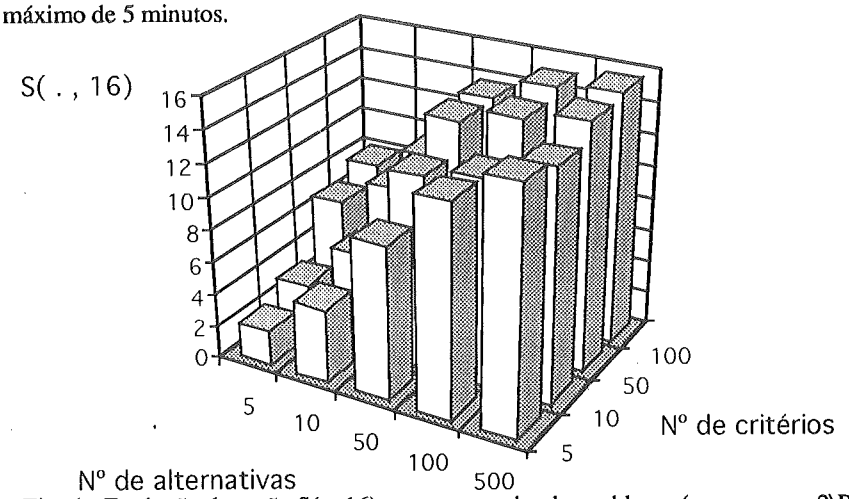


Fig. 4 - Evolução da razão S(., 16) com o tamanho do problema (programa pp2\RI)

nº de critérios	nº de alternat.	critérios alterados	programa sequencial	programa paralelo (16)	Speeup S(. , 16)
5	10	todos um	instantâneo instantâneo	instantâneo instantâneo	4.6 4.2
10	100	todos um	22s 13s	2s 1s	12.6 12.1
10	500	todos um	9m 10s 5m 21s	39s 22s	14.1 14.6
10×10	500	todos um	1h 17m 33s 5m 21s	5m 0s 21s	15.5 15.1

Tab.1 - Tempos de execução durante a análise de robustez (programa pp2\RI)

5.2 Experiências relativas ao método ELECTRE III

O segundo conjunto de experiências efectuadas refere-se ao método ELECTRE III, tal como exposto em [31]. Numa primeira fase, o método agrega a informação na matriz de decisão e a informação inter-critério numa relação de prevalência difusa, definida pelos denominados índices de credibilidade $\sigma(.,.)$. De seguida, procede à exploração dessa relação através das chamadas destilações descendente e ascendente.

O algoritmo da destilação descendente é um processo iterativo que em cada passo calcula uma "qualificação" para cada alternativa e escolhe sucessivamente as melhores alternativas (aquelas com maior qualificação) de modo a construir uma pré-ordem completa (P, I). Este algoritmo tenta escolher em cada iteração o menor número possível de alternativas, de modo a evitar os *ex-aequo*. Na sua execução é necessário resolver subproblemas de determinação do maior índice de credibilidade para um conjunto de pares de alternativa (subproblemas do tipo I)

e subproblemas de determinação da qualificação das alternativas (subproblemas do tipo II). O algoritmo da destilação ascendente é análogo, mas escolhe sucessivamente as piores alternativas por forma a construir uma pré-ordem completa, geralmente distinta da anterior.

No fim dos procedimentos de destilação, as duas pré-ordens completas resultantes podem ser combinadas numa pré-ordem parcial (P, I, R), como segue: para cada par de alternativas (a_1, a_2), a_1 é preferível a a_2 se estiver melhor posicionado que a_2 segundo uma das destilações e não estiver pior posicionado segundo a outra destilação; a_1 é indiferente a a_2 se a_1 e a_2 estiverem posicionados *ex-aequo* em ambas as destilações; e a_1 e a_2 são incomparáveis se a_1 estiver melhor posicionada que a_2 numa das destilações e o contrário acontecer na outra.

Tal como para o método PROMETHEE, e com a mesma finalidade, o primeiro programa a ser elaborado foi um programa sequencial. Todos os programas construídos permitem que se efectue uma análise de robustez à caracterização dos vários critérios, podendo os utilizadores alterar um ou vários parâmetros de cada vez e/ou a forma do denominado limiar de discriminação s(.).

Um primeiro programa paralelo para o ELECTRE (pe1) foi construído a partir do programa sequencial, paralelizando apenas a obtenção dos índices de credibilidade que definem a relação de prevalência. Cada processador P_i será responsável por um subconjunto A_i do conjunto alternativas A, de modo tão equitativo quanto possível. O processador de interface (P_0) divulga os dados do problema pela rede de transputers, incubindo cada processador P_i do cálculo de todos os índices de credibilidade $\sigma(a_1, a_2)$ tais que $a_1 \in A_i$ e $a_2 \in A$. Esta partição do conjunto dos pares ordenados das alternativas é distinto do utilizado para o método PROMETHEE, tendo as escolhas sido influenciadas pela paralelização das etapas seguintes dos métodos. Não se crê que haja diferenças significativas, nesta primeira fase, entre estas duas formas de efectuar a partição. Os índices vão sendo recolhidos pelo P_0 à medida que são calculados, realizando este a fase de exploração da relação obtida enquanto os restantes processadores permanecem inactivos. No caso de o utilizador proceder a uma análise de robustez tudo se passa como no cálculo da solução inicial, mas já não será necessário divulgar aos processadores todos os dados relativos ao problema, bastando comunicar as alterações introduzidas pelo utilizador.

Face ao carácter sequencial dos algoritmos de destilação, paralelizaram-se apenas os subproblemas do tipo I e do tipo II acima referidos. É nestes subproblemas que o programa sequencial gasta a maior parte do tempo de execução. O segundo programa paralelo (pe2) foi construído a partir do programa pe1, acrescentando-lhe a característica de paralelizar uma parte da fase de exploração da relação de prevalência - os subproblemas do tipo I. Um terceiro programa paralelo (pe3) é uma extensão do programa pe2, que além de paralelizar os subproblemas do tipo I, paraleliza os de tipo II. Em ambos os programas a destilação descendente precede a destilação ascendente e os restantes cálculos são efectuados em modo sequencial pelo P_0, que coordena a execução das destilações.

Elaborou-se um quarto programa paralelo (pe4) que apenas se distingue do pe3 pelo facto de as rotinas de cálculo das destilações descendente e ascendente serem executadas em simultâneo (pseudo-paralelismo), entrando em concorrência pelo tempo de processamento em cada processador. Dado que a contribuição de cada processador para o cálculo de uma destilação é proporcional, em cada momento, ao número de alternativas em A_i ainda por classificar, um processador fica rapidamente "sem trabalho" se estas forem muito boas (muito más) durante a destilação descendente (ascendente). Ao usar-se a estratégia de colocar cada processador a repartir o seu tempo pelas duas destilações, espera-se que haja um efeito de compensação que assegure maior equilíbrio entre o trabalho atribuído a cada processador.

O último programa paralelo contruído (pe5) é uma extensão ao programa pe4, que executa em modo paralelo a combinação das pré-ordens completas resultantes das destilações numa pré-ordem parcial (P, I, R). Para todo o par de alternativas que lhe seja atribuído, cada processador decide qual a situação de preferência que se verifica.

Os programas paralelos pe1 a pe5 são inteiramente funcionais, no sentido em que podem mostrar ao utilizador, se este o entender, os índices de credibilidade, bem como os resultados da exploração da relação definida por esses índices. No início dos cálculos é pedida ao utilizador a informação de saída por este pretendida e esta é utilizada pelos diferentes processadores de modo a não originar mais comunicação do que a necessária: os processadores só comunicarão os índices de credibilidade se os decisores assim o desejarem.

Conduziram-se várias experiências de modo a estudar os desempenhos dos programas paralelos em diversas situações. Face a restrições de memória, foram gerados dois bancos de problemas especializados: o banco 1 distingue-se por possuir muitas alternativas e combina 4, 5, 8 ou 10 critérios com 16, 32, 64, 128, 192 ou 256 alternativas; o banco 2 distingue-se por possuir muitos critérios, combinando 8 a 100 critérios com 16, 20, 24, 28, 32 ou 50 alternativas. A obtenção dos critérios resulta da combinação de 4, 5, 8 ou 10 pontos de vista com 2, 5 ou 10 intervenientes (maneiras distintas de operacionalizar cada ponto de vista). Experimentou-se executar os vários programas construídos, com diferentes níveis de exigência no que respeita às saídas (relação de prevalência, resultado da exploração) e em quatro tarefas distintas: computação de uma primeira solução, computação de outra solução após alterar um parâmetro de um critério, computação de outra solução após alterar parâmetros de todos os critérios e computação de outra solução após alterar o limiar de discriminação s(.). Tal como para o PROMETHEE, os resultados referir-se-ão sempre à utilização dos 16 processadores, estando fora do âmbito deste artigo as referentes à utilização de 4 e 8 transputers.

Os melhores desempenhos foram obtidos ora pelo programa pe1 ora pelo pe4. As Fig.5 e 6 ilustram as situações em que cada programa se superioriza, utilizando-se o asterisco para assinalar quando se exige conhecer os índices de credibilidade (no caso do pe4). Tal como sucedeu no método PROMETHEE, os melhores desempenhos pressupõem que os decisores abdicam de conhecer a relação de prevalência difusa. No entanto, é razoável supor estes não

terão muito interesse em conhecer tais índices quando o número de alternativas é elevado. Nas situações em que o programa pe1 é superior, o programa sequencial é satisfatoriamente rápido (até 10 segundos), o que não sucede nas situações em que o programa pe4 é melhor (cf.Tab.2). Os desempenhos mais interessantes para o utilizador, que não correspondem necessariamente aos maiores *speedups*, são por isso os obtidos pelo programa pe4.

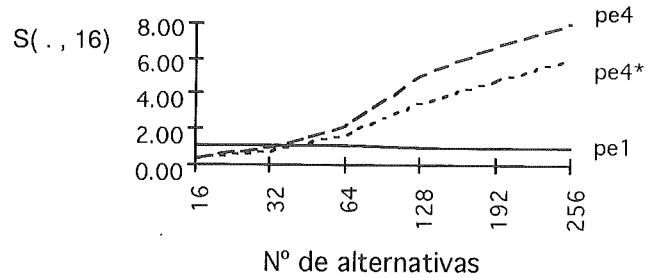


Fig. 5 - Razão S(., 16) na análise da alteração de um critério (problemas com 4 critérios)

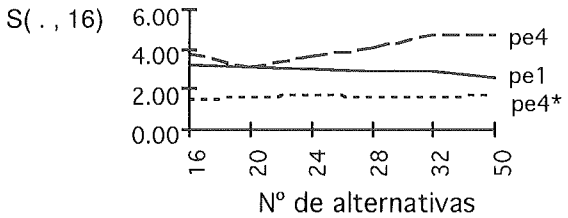


Fig. 6 - Razão S(., 16) na análise da alteração de um critério (problemas com 100 critérios)

Os desempenhos mais interessantes na execução do método ELECTRE III (pe4) são atraentes por ocorrerem nas situações em que o programa sequencial é mais lento, i.e. problemas com grande número de alternativas (Fig.7). Outro facto que contribui para valorizar os desempenhos obtidos é o facto de estes ocorrerem na fase de análise de robustez, que se considera ser a ocasião em que a rapidez de execução é mais importante. Face aos resultados obtidos, tempos de resposta que poderiam ser considerados desapontadores (os obtidos pelo programa sequencial) são agradavelmente reduzidos quando se utiliza o programa pe4 (Tab.2).

nº de critérios	nº de alternat.	critérios alterados	programa sequencial	programa paralelo (16)	Speeup S(. , 16)
4	32	todos um	instantâneo instantâneo	instantâneo instantâneo	1.08 0.95
4	128	todos um	30s 27s	8s 5s	3.94 4.97
5	256	todos um	3m 50s 3m 14s	37s 25s	6.15 7.72
10×10	32	todos um	10s 2s	2s 2s	4.73 1.00

Tab.2 - Tempos de execução durante a análise de robustez (programa pe4)

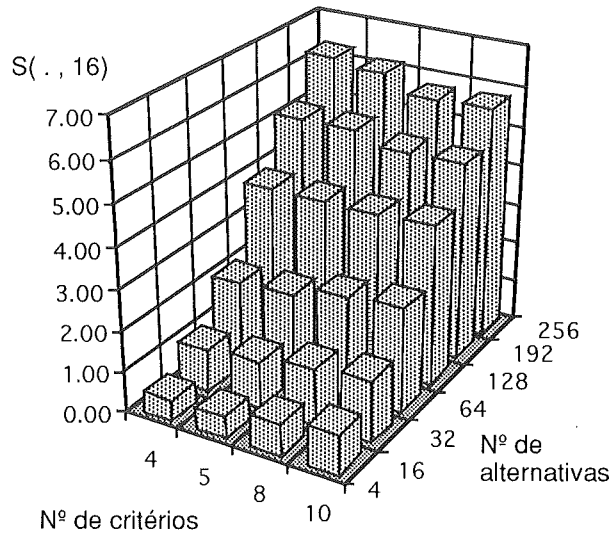


Fig. 7 - Evolução da razão S(., 16) com o tamanho do problema (programa pe4)

6. Sumário e conclusões

Discutiu-se a aplicação de computadores paralelos para executar métodos de apoio multicritério à decisão, tendo-se defendido a sua pertinência durante a fase de análise de robustez. Identificaram-se as metodologias de prevalência como aquelas em que o paralelismo poderia oferecer maiores benefícios, tendo-se abordado dois dos mais representativos métodos de prevalência, o PROMETHEE e o ELECTRE III, para ilustrar tal facto. Foram para isso construídos vários programas paralelos para cada método e efectuou-se um conjunto de experiências que, não pretendendo ser exaustivo, fosse suficientemente multifacetado para permitir julgar os seus méritos relativos.

Os resultados obtidos permitem concluir que as estratégias de paralelização seguidas obtêm os melhores desempenhos quando os decisores sacrificam a possibilidade de conhecer os índices que definem a relação de prevalência. Nesse caso, a diminuição do tempo de resposta às solicitações dos decisores poderá ser bastante apreciável, encorajando a realização de análises de robustez e tornando os problemas de grande dimensão mais fáceis de estudar. Este sacrifício limita os programas, mas em menor grau do que se poderia supor. De facto, os números que os decisores são impedidos de conhecer não formam um conjunto inteligível em problemas de grande dimensão, face à quantidade de informação necessária para definir uma relação de prevalência difusa, que cresce com o quadrado do número de alternativas. Por outro lado, nada impede que a relação de prevalência, ou parte da mesma, não possa ser fornecida *a posteriori*, enquanto os decisores discutem os resultados da exploração dessa relação.

Os desempenhos para o método ELECTRE III foram inferiores aos conseguidos pelos programas que executam o PROMETHEE, esses de facto muito bons, para os quais foi possível experimentar situações com maior número de critérios e alternativas. Todavia, em

contrapartida, no ELECTRE III os tempos de execução parecem ser susceptíveis de melhoramento (caso se consiga testar casos de maior dimensão) e foram satisfatórios para as dimensões de problema utilizadas, sobretudo na importante fase de análise de robustez.

A razão entre os tempos de execução do programa sequencial e dos melhores programas paralelos aumenta em geral com a dimensão do problema e, consequentemente, com a morosidade de resolver o problema num computador sequencial. Em ambos os estudos foi possível identificar situações em que a diferença entre os tempos de execução de um programa paralelo e do programa sequencial pode constituir a diferença entre a utilização intensiva do programa e a não utilização do mesmo.

O trabalho desenvolvido pode dar origem a sistemas de apoio à decisão, com bases de dados incorporadas e interfaces amigáveis, porventura especializados em situações com grande número de alternativas (como por exemplo a avaliação de projectos, como na aplicação BANKADVISER, ou situações em que o conjunto de alternativas é definido como combinação de vários subconjuntos de acções fragmentarias), ou com elevado número de critérios (por exemplo, a sugerida situação de decisão em grupo). Outra via de investigação é a extensão deste estudo a outros métodos de prevalência, tais como os dedicados à problemática de afectação (p. ex. ELECTRE TRI [33]), para os quais é possível antever algumas situações de decisão com grande número de alternativas.

Referencias

[1] ANSI-C front-end documentation, release 92.1, ACE, Novembro (1992).

[2] T800/T9000 back-end documentation, release 92.1, ACE, Dezembro (1992).

[3] Adams, D.A., *Parallel processing implications for management scientists*, Interfaces 20 (1990) 88-90.

[4] Bana e Costa, C.A. and Vansnick, J.-C., *Sur la quantification de judgements de valeur: l'approache MACBETH*, Cahier du LAMSADE 117, Université Paris-Dauphine, Paris (1993).

[5] Brans, J.P., *L'ingéniére de la décision. Elaboration d'instruments d'aide à la décision. Méthode PROMETHEE*, Université Laval, Colloque d'Aide à la Décision, Québec (1982) 183-213.

[6] Brans, J.P. and Vincke, Ph., *A preference ranking organisation method (the PROMETHEE method for multiple-criteria decision making)*, Management Science 31 (1985) 647-656.

[7] Costa, J.P. and Clímaco, J.N., *A multiple reference point parallel approach in MCDM*, in G.H. Tzeng et al. (Eds.), Expand and enrich the domain of thinking and application, Springer-Verlag, Berlim (1994) 255-263.

[8] Dias, L., Costa, J.P. and Clímaco, J.N., *A parallel approach to the Analytical Hierarchy Process decision support tool*, Computing Systems in Engineering 6 (1995) 431-436.

[9] Dias, L., Costa, J.P. and Clímaco, J.N., *A parallel implementation of the PROMETHEE method*, a aparecer em European Journal of Operational Research.

[10] Dias, L., Costa, J.P. and Clímaco, J.N., *Parallelism in the ELECTRE III Outranking Method: Implementation Issues*, Relatório de investigação, INESC - Núcleo de Coimbra (1994).

[11] Duncan, R., *A survey of parallel computer architectures*, IEEE Computer 23 (1990) 5-16.

[12] Flynn, M.J., *Very high-speed computing systems*, Proceedings of the IEEE 54 (1966) 1901-1909.

[13] Gustafson, J.L., *Reevaluating Amdahl's law*, Communications of the ACM 31 (1988) 532-533.

[14] Hulberg, T.H., Cardoso, D.M. and Gondzio, J., *Uma implementação paralela do método Simplex generalizado*, Apresentado no 7º Congresso da APDIO (IO96), 1-3 Abril (1996).

[15] The transputer data book, INMOS (1989).

[16] Jacquet-Lagrèze, E. and Siskos, J., *Assessing a set of additive utility functions for multicriteria decision making - the UTA method*, European Journal of Operations Research 10 (1982) 151-164.

[17] Keeney, R.L. and Raiffa, H., *Decisions with multiple objectives: preferences and value tradeoffs*, John Wiley and Sons, New York (1976).

[18] Mareshal, B. and Brans, J.P., *BANKADVISER: An industrial evaluation system*, European Journal of Operations Research 54 (1991) 318-324.

[19] Nussbaum, D. and Agarwal, A., *Scalability of parallel machines*, Communications of the ACM 34 (1991) 57-61.
[20] PARIX, Release 1.2, Software documentation, Parsytec, Aachen, Março (1993).
[21] Pomerol, J.-C. and Barba-Romero, S., *Choix multicritère dans l'entreprise: principe et pratique*, Editions Hermes, Paris (1993).
[22] Roy, B., *Classement et choix en présence de points de vue multiples (la méthode ELECTRE)*, Revue Informatique et Recherche Opérationelle, 2e. Année, 8 (1968) 57-75.
[23] Roy, B., *ELECTRE III: un algorithme de classement fondé sur une représentation flue des préférances en présence de critères multiples*, Chaiers du Centre d'Etudes de Recherche Opérationelle 20 (1978) 3-24.
[24] Roy, B., *Méthodologie multicritère d'aide à la décision*, Economia, Paris (1985).
[25] Roy, B., *The outranking approach and the foundations of ELECTRE methods* in C.A. Bana e Costa (ed.) Readings in Multiple Criteria Decision Aid, Springer-Verlag, Berlim (1990).
[26] Roy, B. and Bertier, P., *La méthode ELECTRE II: une méthode de classement en présence de critères multiples*, Note de travail nº 142, SEMA, Direction Scientifique, Paris (1971).
[27] Roy, B. and Hugonnard, J.-C., *Classement de prolongements des lignes de métro en banlieue parisiénne*, Chaiers du CERO 24 (1982) 153-171.
[28] Roy, B. and Skalka, J.M., *ELECTRE IS: aspects méthodologiques et guide d'utilisation*, Document du LAMSADE 30, Université Paris-Dauphine, Paris (1985).
[29] Saaty, T.L., *The analytic hierarchy process*, McGraw-Hill, New York (1980).
[30] Schärlig, A., *Décider sur plusieurs critères*, Collection Diriger l'Entreprise, Presses Polytechniques Romandes (1985).
[31] Skalka, J., Bouyssou, D. and Bernabeu, Y., *ELECTRE III et IV - Aspects methodologiques et guide d'utilisation*, Document du LAMSADE 25, Université Paris-Dauphine, Paris (1986).
[32] Vincke, Ph., *L'aide multicritère à la décision*, Éditions de l'Université Libre de Bruxelles, Bruxelles (1989).
[33] Yu, W., *ELECTRE TRI. Aspects méthodologiques et guide d'utilisation*, Document du LAMSADE 74, Université Paris-Dauphine, Paris (1992).

INSTRUÇÕES AOS AUTORES

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

Prof. Joaquim J. Júdice
Departamento de Matemática
Universidade de Coimbra
3000 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.

Fotografia, Montagem
Impressão e Acabamentos
Tip.Nocarnil
COIMBRA

ÍNDICE