

# 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 prevalência 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 necessidade 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 intervenientes, 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 conflituosos.

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.

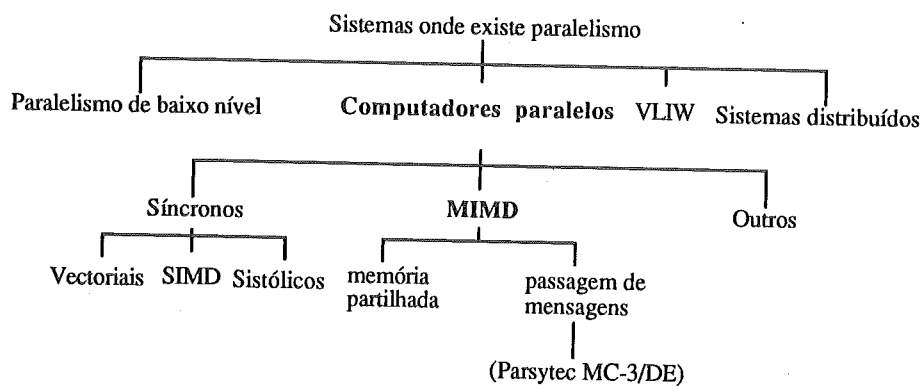


Fig. 1 - Esquema de classificação

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).

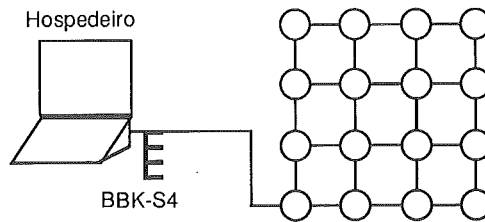


Fig. 2 - Configuração do ambiente de desenvolvimento

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  $(a_1, \dots, a_m)$  e  $n$  critérios  $(g_1, \dots, g_n)$ . A matriz tem dimensão  $m \times n$  e os seus elementos são os desempenhos de cada alternativa segundo cada critério:

$$G_{ij} = g_j(a_i), \quad i = 1, \dots, m \quad \text{e} \quad j = 1, \dots, n.$$

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  $V(g_1, g_2, \dots, g_n)$  que agrega todos os critérios  $g_1, g_2, \dots, g_n$  considerados. A filosofia subjacente às metodologias que se inserem nesta abordagem é a de que o decisor tenta maximizar a função  $V(\cdot)$ , 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 noutra 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 limitado 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  $g_j(\cdot)$  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 precursor 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].

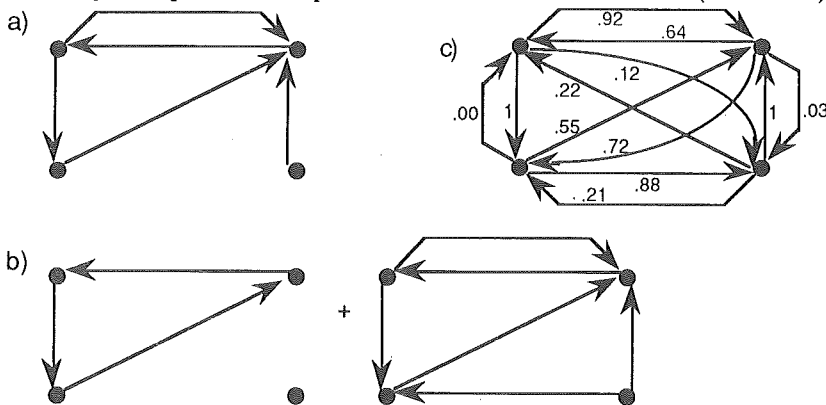


Fig. 3 - Exemplos de relações de prevalência: a) única, b) múltiplas, c) difusa

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 considera, 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 PROMETHEE 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<sub>0</sub> 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.

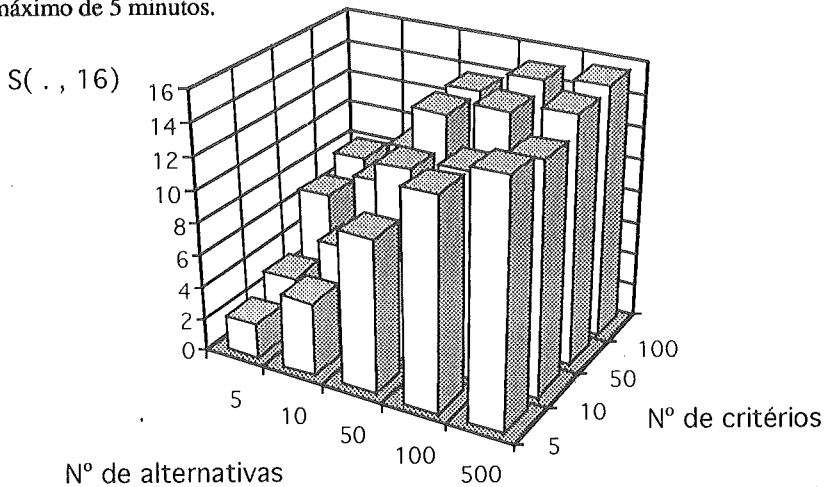


Fig. 4 - Evolução da razão  $S(., 16)$  com o tamanho do problema (programa pp2RI)

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 pp2RI)

## 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  $s(\cdot)$ .

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.

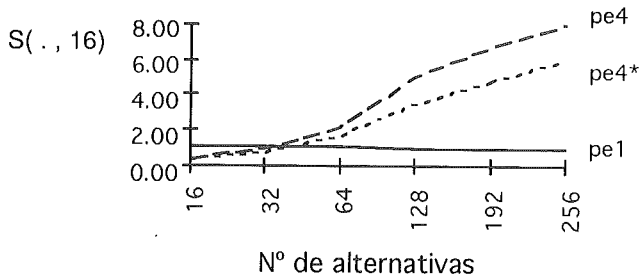


Fig. 5 - Razão  $S(., 16)$  na análise da alteração de um critério (problemas com 4 critérios)

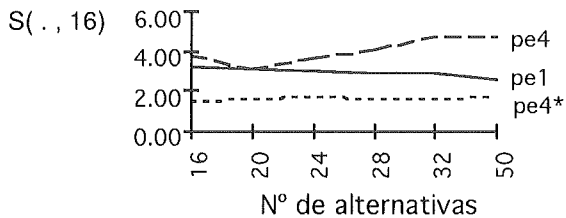


Fig. 6 - Razão  $S(., 16)$  na análise da alteração de um critério (problemas com 100 critérios)

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)



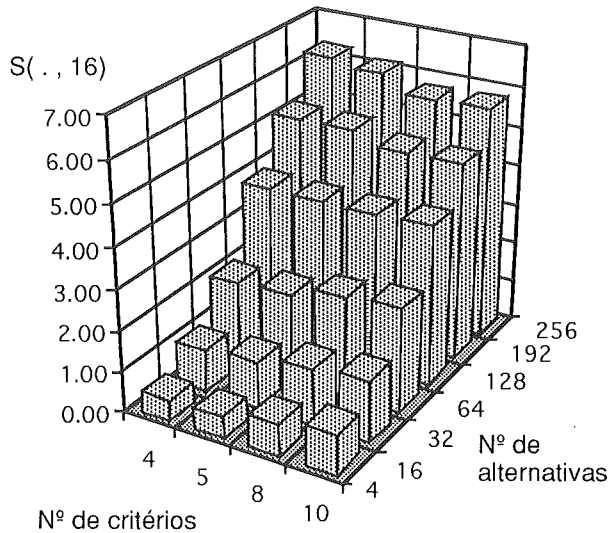


Fig. 7 - Evolução da razão  $S(., 16)$  com o tamanho do problema (programa pe4)

## 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, conseqüentemente, 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 juggements de valeur: l'approche MACBETH*, Cahier du LAMSADE 117, Université Paris-Dauphine, Paris (1993).
- [5] Brans, J.P., *L'ingénierie 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, Berlin (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 floue des préférences 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*, Economica, 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, Berlin (1990).
- [26] Roy, B. and Bertier, P., *La méthode ELECTRE II: une méthode de classement en présence de critères multiples*, Note de travail n° 142, SEMA, Direction Scientifique, Paris (1971).
- [27] Roy, B. and Hugonnard, J.-C., *Classement de prolongements des lignes de métro en banlieue parisienne*, 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] Schärflig, 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 méthodologiques 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).