

FACULDADE DE ECONOMIA
Universidade de Coimbra

TRABALHO DE SÍNTESE

O Processamento Paralelo e os Métodos PROMETHEE e ELECTRE III no Apoio Multicritério à Decisão

**Alguns Programas, Experiências Computacionais,
e Discussão dos Resultados**

Provas de Aptidão Pedagógica e Capacidade Científica

Luis Miguel Cândido Dias

COIMBRA 1995

Agradecimentos

Ao Professor João Clímaco, pela sua orientação e inestimável apoio, aqui expresso o meu reconhecimento.

Ao Eng^o João Paulo Costa, pelas suas sugestões e críticas oportunas, a minha gratidão.

À Eng^a Teresa Martinez e ao Professor Carlos Henggeler, pela amabilidade que tiveram ao ler capítulos destas Provas, o meu obrigado.

Agradeço também aos meus colegas do INESC de Coimbra e da Faculdade de Economia, pelo apoio, amizade e ambiente de trabalho dispensados.

ÍNDICE GERAL

| | página |
|---|--------|
| RESUMO | 3 |
| 1 - NOTAS INTRODUTÓRIAS | 4 |
| 1.1 Motivação e objectivos do trabalho | 4 |
| 1.2 Aplicação de processamento paralelo a MAMD | 4 |
| 1.3 Resumo dos capítulos seguintes | 6 |
| 2 - APOIO MULTICRITÉRIO À DECISÃO | 7 |
| 2.1 Porquê "apoio multicritério" ? | 7 |
| 2.2 Processo de decisão | 8 |
| 2.3 Agregação de preferências | 9 |
| 2.4 Métodos conducentes a um critério de síntese | 11 |
| 2.5 Métodos conducentes a uma relação de prevalência de síntese | 14 |
| 2.6 Algumas noções importantes | 18 |
| 3 - PROCESSAMENTO PARALELO | 24 |
| 3.1 Introdução | 24 |
| 3.2 Esquema de classificação | 25 |
| 3.3 Processamento paralelo em arquitecturas MIMD | 29 |
| 3.4 Multiprocessadores de memória partilhada | 34 |
| 3.5 Multiprocessadores de passagem de mensagens (multicomputadores) | 36 |
| 3.6 Ambiente utilizado | 39 |
| 4 - PARALELIZAÇÃO DO MÉTODO PROMETHEE | 45 |
| 4.1 Introdução | 45 |
| 4.2 O método PROMETHEE | 46 |
| 4.3 Um algoritmo para a execução do método PROMETHEE | 52 |
| 4.4 Possibilidades de paralelização no método PROMETHEE | 54 |
| 4.5 Programas construídos | 55 |
| 4.6 Experiências efectuadas | 65 |
| 4.7 Resultados obtidos | 67 |
| 4.8 Algumas conclusões | 74 |
| 5 - PARALELIZAÇÃO DO MÉTODO ELECTRE III | 76 |
| 5.1 Introdução | 76 |
| 5.2 O método ELECTRE III | 77 |
| 5.3 Um algoritmo para a execução do método ELECTRE III | 83 |
| 5.4 Possibilidades de paralelização no método ELECTRE III | 86 |
| 5.5 Programas construídos | 87 |
| 5.6 Experiências efectuadas | 97 |
| 5.7 Resultados obtidos | 99 |
| 5.8 Algumas conclusões | 109 |
| 6 - CONCLUSÕES E VIAS DE INVESTIGAÇÃO FUTURA | 111 |
| APÊNDICE A: RESULTADOS RELATIVOS AO MÉTODO PROMETHEE | 114 |
| APÊNDICE B: RESULTADOS RELATIVOS AO MÉTODO ELECTRE III | 122 |
| REFERÊNCIAS E BIBLIOGRAFIA | 133 |

RESUMO

No contexto do apoio à avaliação de alternativas considerando diversos critérios potencialmente conflituosos têm sido propostas várias abordagens. Entre estas abordagens encontram-se os métodos de apoio multicritério à decisão ELECTRE III e PROMETHEE, dois conhecidos métodos de prevalência, que apresentam possibilidades de exploração algorítmica no âmbito dos paradigmas habitualmente utilizados em processamento paralelo.

O processamento paralelo tem vindo a ser cada vez mais divulgado e potencia a resolução de problemas computacionais de complexidade crescente. A utilização de processamento paralelo, como meio para tornar mais rápida a execução de métodos multicritério, é especialmente útil na fase de análise de sensibilidade dos resultados face à variação dos parâmetros iniciais. Nesta fase, experimenta-se variar os parâmetros de modo a conhecer melhor a situação de decisão, aumentar a confiança nos resultados obtidos e, em alguns casos, conhecer melhor o funcionamento do método escolhido. O número de variações poderá ser muito grande, quer por existirem vários intervenientes a desejar efectuar análises de sensibilidade, quer por a decisão em causa a isso obrigar (em termos de complexidade ou responsabilidade). Nesses casos, torna-se extremamente importante que a aplicação que executa os métodos seja suficientemente rápida a fornecer resultados, de modo a não desencorajar os intervenientes.

Apresenta-se inicialmente uma introdução ao paradigma do apoio à decisão multicritério bem como uma descrição do ambiente computacional utilizado, enquadrado no conjunto de sistemas onde existe paralelismo. De seguida referem-se algumas possibilidades de paralelização dos métodos PROMETHEE e ELECTRE III, apresentando-se alguns programas que exploram essas possibilidades num multiprocessador MIMD ("Multiple Instruction Multiple Data") com dezasseis processadores (Inmos Transputers T805), que comunicam através do envio/recepção de mensagens. Os programas foram utilizados para efectuar experiências computacionais perante várias situações de decisão fictícias, cujos resultados são apresentados e discutidos.

Os resultados obtidos permitem concluir que em certas situações a diminuição dos tempos de resposta às solicitações dos utilizadores pode ser apreciável, nomeadamente quando não se deseja obter os índices que definem uma relação de prevalência e nos casos em que o programa sequencial é mais lento. Por fim, apresentam-se algumas vias para investigação futura.

1 - NOTAS INTRODUTÓRIAS

1.1 Motivação e objetivos gerais do trabalho¹

À medida que os computadores paralelos se tornam mais disseminados e acessíveis, a sua utilização começa a estender-se a aplicações que não exigem tipicamente capacidades de processamento numérico intensivo. Uma aplicação em que o processamento paralelo tem começado a ser utilizado é a dos sistemas de apoio à decisão (SAD) “multicritério”, que apoiam a tomada de decisões considerando simultaneamente múltiplos critérios. Neste contexto, o processamento paralelo tem sido aplicado em ferramentas de apoio multicritério à decisão que consideram um conjunto contínuo de soluções alternativas, definido através de restrições (veja-se p. ex. [Costa e Clímaco, 1994]).

Neste trabalho faz-se um estudo sobre a utilização de um computador paralelo para a execução de alguns métodos de apoio multicritério à decisão (MAMD), considerando um conjunto de alternativas discreto. O estudo, que se crê ser pioneiro², foca dois dos mais conhecidos métodos de prevalência, o PROMETHEE e o ELECTRE III, que possibilitam a exploração de diversas possibilidades de processamento paralelo, especialmente na fase de comparação de pares de alternativas, que é típica neste tipo de métodos. Pretende-se sobretudo avaliar até que ponto, e em que situações, é vantajoso utilizar um computador paralelo para executar os métodos focados e também identificar algumas vias para investigação futura com base nos resultados obtidos. Espera-se também que este trabalho contribua para que exista no futuro maior investigação sobre a utilização de processamento paralelo para este tipo de métodos.

1.2 Aplicação de processamento paralelo a MAMD

A utilização de computadores no apoio à decisão não deve ser entendida como um mero meio de obter resultados com maior rapidez. Mais do que isso, pode entender-se como uma oportunidade para executar uma quantidade de trabalho maior do que a que se executaria de outro modo. Esse trabalho adicional, que poderá contribuir para que o decisor (ou decisores) sinta maior confiança (e, conseqüentemente, maior satisfação) nos resultados do apoio à decisão, consistiria nas tarefas de experimentação apresentadas de seguida.

¹ Alguns conceitos a que se faz referência neste capítulo serão apresentados nos capítulos 2 e 3.

² Conjuntamente com [Dias et al., 1993], que utiliza uma máquina paralela diferente da utilizada neste trabalho, no qual é descrita uma paralelização do método AHP.

Por um lado, o decisor só sentirá confiança nos resultados se conhecer bem os métodos, o que implica experimentar a execução desses métodos com vários conjuntos de dados. Por outro lado, mesmo que o modo como o método obtém os resultados apresentados seja compreendido, é pouco natural que o decisor se sinta completamente seguro com os resultados obtidos utilizando um conjunto inicial de dados. Para isso contribuem a necessidade, presente nalguns métodos (p.ex. ELECTRE III), de fixação de parâmetros com um significado técnico de percepção não imediata e a eventual falta de convicção com que o decisor responde a questões de natureza subjectiva (p.ex. atribuição de pesos aos diferentes critérios). Por esse motivo, dever-se-á efectuar uma análise de sensibilidade, em que se verifica o impacto da variação de alguns parâmetros nos resultados fornecidos pelo método.

Em [Antunes et al., 1994] mostra-se que este processo de aprendizagem do decisor poderá beneficiar da utilização de um sistema de apoio à decisão dedicado a MAMD baseado numa grande flexibilidade e num interface amigável. Este sistema foi construído para um computador sequencial, podendo-se levantar a questão da necessidade de utilizar computadores paralelos como impulsionadores do processo de experimentação/aprendizagem do decisor ou dos decisores.

Segundo [Pomerol e Barba-Romero, 1993, sec. 9.1], a complexidade computacional dos MAMD 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). Tornam, deste modo, questionável a necessidade de utilizar computadores de elevado desempenho como são os computadores paralelos. No entanto, acredita-se que poderá ser útil aplicar técnicas de processamento paralelo para executar este tipo de métodos, pelas razões a seguir apresentadas.

A principal razão deriva da necessidade, que pode ser satisfeita por computadores paralelos, de oferecer elevados desempenhos durante a aprendizagem do decisor, sobretudo na fase de análise de sensibilidade. É nesta fase que se torna crucial um tempo de resposta rápido (se possível imediato) por parte do SAD, tendo em vista a interactividade desejada. Note-se que a obtenção de um primeiro resultado por parte do SAD não é crítica em termos de tempo de resposta. Afinal, o decisor terá em princípio participado anteriormente numa fase relativamente longa para estruturar a situação de decisão. No entanto, a fase de análise de sensibilidade já é crítica, porque se os tempos de resposta dificultarem a interacção com o SAD então o decisor não se sentirá encorajado a ser exaustivo nesta importante fase. Pode ainda suceder que existam vários decisores a analisar a situação de decisão em conjunto, sugerindo, cada um, diferentes modos de variar os parâmetros utilizados pelo método. Nesta situação, ainda se torna mais importante que a ferramenta computacional que apoia a decisão seja rápida, de modo a encorajar a realização de todas as experiências necessárias à satisfação dos decisores.

Uma outra razão é a de os computadores paralelos criarem oportunidades para aumentar a dimensão dos problemas a resolver para além do considerado "razoável", conforme a interpretação dada por Gustafson à lei de Amdahl, que será apresentada na secção 3.3. O aparecimento de aplicações que executem MAMD de grande dimensão (número de alternativas e número de critérios), poderá redefinir o que é considerado "razoável" para os computadores sequenciais.

Por fim, a crescente popularidade e vulgarização dos computadores paralelos implica que seja cada vez mais importante e oportuna a familiarização da comunidade científica com as técnicas de processamento paralelo. Segundo [Zionts, 1992], o processamento paralelo será o paradigma de computação em larga escala dominante nas próximas décadas e [Adams, 1990] afirma mesmo que a familiarização com o processamento paralelo será cada vez menos um "luxo" e cada vez mais uma necessidade.

1.3 Resumo dos capítulos seguintes

O capítulo seguinte representa uma breve introdução ao paradigma do apoio multicritério à decisão. O terceiro capítulo pretende apresentar o ambiente computacional utilizado neste trabalho e enquadrá-lo no conjunto de sistemas onde existe paralelismo. Os capítulos 4 e 5 descrevem, respectivamente para os métodos PROMETHEE e ELECTRE III, os seus algoritmos, as experiências conduzidas e os resultados obtidos. O capítulo 6 apresenta as conclusões deste trabalho e vias de investigação futura sugeridas pelos resultados obtidos. Por fim apresenta-se uma lista de referências e bibliografia.

2 - APOIO MULTICRITÉRIO À DECISÃO

2.1 Porquê "apoio multicritério" ?

Tradicionalmente, as decisões baseadas em modelos matemáticos eram tomadas considerando um objectivo que se pretendia maximizar ou minimizar (paradigma da optimização monocritério). Porém, muitas vezes, o critério único esconde várias perspectivas, ou pontos de vista, sobre a situação de decisão. Por exemplo, essas perspectivas são frequentemente convertidas em ganhos ou custos monetários, numa situação em que se deseja maximizar o lucro (ou minimizar os prejuízos).

Alguns autores (p. ex. [Pomerol e Barba-Romero, 1993], p.12) contrapõem que as decisões no "mundo real" envolvem por natureza múltiplos critérios, e anunciam o fim daquela que seria a "ficção monocritério". Por outras palavras, a realidade exigiria múltiplos critérios e a abordagem monocritério seria apenas um "pálido reflexo" da complexidade do real. A expressão "apoio à decisão multicritério" remete de algum modo para esta linha de argumentação.

No entanto, muitas técnicas de investigação operacional monocritério têm permanecido válidas e de grande utilidade, pelo que será por vezes preferível concordar com [Bouyssou, 1993], quando este afirma que a adopção de uma abordagem multicritério consiste também num "acto de fé" (e não somente numa necessidade imposta pela realidade). Este "acto de fé" resume-se a acreditar que a tomada em consideração de múltiplos critérios poderá jogar um papel benéfico no processo de decisão. Bouyssou apresenta simultaneamente três argumentos para justificar o referido "acto de fé": a constituição de uma base para o diálogo entre os intervenientes utilizando diversos pontos de vista comuns, a maior facilidade em incorporar incertezas nos dados sobre cada ponto de vista e, sobretudo, o encarar de cada solução como um compromisso entre objectivos em conflito. Este último argumento sublinha que raramente se encontrará uma situação em que exista um curso de acção superior aos restantes sobre todos os pontos de vista, pelo que a explicitação de vários critérios ajudará a evidenciar o carácter conflituoso dos pontos de vista dos actores (ver sec. 2.6), tanto a nível individual como a nível colectivo.

O objectivo do apoio à decisão (ver sec. 2.6) será então o de ajudar o decisor (ou decisores) a agir de acordo com as preferências que representa. O apoio à decisão é multicritério quando se tornam explícitos vários pontos de vista, através de critérios (ver sec. 2.6), em vez de um só. Falar-se-à, por esse motivo, em apoio multicritério à decisão em detrimento de apoio à decisão multicritério, renunciando assim ao argumento exclusivamente "realista" (ver também [Vincke, 1989], [Roy, 1990a]).

2.2 Processo de decisão

No início de um processo de decisão raramente surgem pré-especificados o conjunto dos cursos de acção possíveis (acções potenciais, ou alternativas) e o conjunto dos critérios. É portanto necessário procurar essas alternativas³ e esses critérios. Em [Keeney, 1992, p.49] distinguem-se dois modos de o fazer: com ênfase nas alternativas e com ênfase nos valores ou critérios. No primeiro caso identificam-se as alternativas, depois identificam-se os critérios e por fim avaliam-se as alternativas utilizando os critérios. No segundo caso começa-se por identificar os critérios (a partir dos valores dos decisores), depois identificam-se as alternativas e por fim procede-se à avaliação, sendo esta a abordagem defendida por Keeney. No entanto, talvez não seja despropositado admitir um processo iterativo, em que após identificar os valores, se especificarem as alternativas, voltando a reiterar o processo as vezes necessárias.

A avaliação das diversas alternativas pode iniciar-se após a escolha de um método de avaliação. Essa escolha não é simples (é considerada um problema multicritério *per se*), pelo que se pode recorrer a vários métodos diferentes. A escolha dos métodos a utilizar será influenciada pelo tipo de recomendação que deve resultar do processo de decisão, isto é, depende da problemática de apoio à decisão (ver sec. 2.6). Da escolha efectuada resulta o conjunto de dados que será necessário recolher, dado que diferentes métodos têm diferentes requisitos a nível de informação. Os dados serão frequentemente de natureza subjectiva, obtidos através da colocação de questões aos decisores. É através dessas questões que se pode estabelecer, por exemplo, a importância relativa ou o "peso" de cada critério na avaliação das alternativas.

Em traços gerais, no caso em que o conjunto de alternativas é discreto, pode dividir-se a avaliação das alternativas em duas fases. Numa primeira fase avalia-se cada alternativa segundo cada critério, obtendo-se uma matriz de decisão (ver sec. 2.6) da qual se podem retirar as preferências por critério. Numa segunda fase agregam-se as avaliações segundo cada critério de modo a obter uma avaliação global, expressa por uma estrutura de preferências multicritério (ver sec. 2.6). Esta segunda fase, a que se pode chamar agregação de preferências, será tratada nas secções seguintes.

Por fim, é desejável efectuar uma análise de sensibilidade para verificar a robustez dos resultados obtidos. Esta fase é tanto mais importante quanto tiver sido difícil obter respostas subjectivas por parte dos intervenientes (p. ex. para indicar diferentes importâncias para os vários critérios), ou no caso de o método escolhido requerer a fixação de parâmetros de natureza técnica cuja compreensão não seja imediata.

As descobertas feitas ao longo do processo de decisão podem ser sempre postas em causa, sendo por vezes de admitir recuos no processo e a exploração de novos

³ Usar-se-á o termo "alternativa" por se considerar que as acções potenciais são mutuamente exclusivas. Este termo é mais intuitivo que o termo "acção" e mais conciso que "curso de acção".

caminhos. Essa exploração de diferentes caminhos será influenciada por todo o conhecimento que o decisor possui na altura sobre a situação de decisão, e que vai aumentando à medida que se realizam novas experiências. Nesse sentido, o processo de decisão pode ser encarado como uma aprendizagem (ver a este propósito [Bana e Costa, 1993]).

2.3 Agregação de preferências

Doravante considera-se que existe apenas um interveniente (o decisor), eventualmente um grupo de indivíduos, que representa um único sistema de valores. Situações de decisão que envolvem múltiplos intervenientes remetem para a área de decisão em grupo e negociação, sendo a agregação das preferências de múltiplos intervenientes um problema distinto, embora relacionado, da agregação de preferências multicritério de um único interveniente.

Alguns métodos de apoio multicritério à decisão (MAMD) são elementares, por serem bastante intuitivos e pouco exigentes em termos computacionais e em termos de informação por parte do decisor. No entanto, podem ser importantes em determinada fase do processo de decisão.

Um método elementar é o método da dominância, que visa retirar do conjunto de alternativas todas as que forem dominadas. Diz-se que uma alternativa a_0 é dominada por uma alternativa a_1 se o desempenho de a_1 for igual ou melhor que o de a_0 em todos os critérios, sendo estritamente melhor em pelo menos um critério. O método consiste em comparar sucessivamente pares de alternativas, eliminando as que forem dominadas. Este método pode ser especialmente útil numa filtragem inicial no âmbito de uma situação de escolha da melhor alternativa (problemática α , ver "problemáticas de apoio à decisão", sec. 2.6), dado que uma alternativa dominada nunca poderá ser considerada a melhor.

Dois outros métodos elementares, os métodos de Borda e de Condorcet, surgiram no contexto da votação (agregação de preferências de múltiplos decisores). Suponha-se por exemplo que se vão considerar 4 critérios e 10 alternativas. Estes dois métodos requerem que a avaliação das alternativas segundo cada critério resulte apenas numa ordenação das alternativas, da melhor para a pior. Haverá assim 4 ordens monocritério, uma por critério, e cada uma das 10 alternativas possuirá uma posição em cada ordem. No contexto unidecisor o problema consiste em agregar essas diferentes ordenações.

Segundo o método de Borda, atribui-se uma pontuação a cada alternativa correspondendo à soma das suas posições (p. ex. se uma alternativa estiver em 3º, 4º, 1º e 6º lugares, respectivamente, em cada uma das quatro ordens, a sua pontuação será de $3+4+1+6=14$). Ordenam-se depois as alternativas por ordem crescente de pontuação para obter uma ordenação agregada. A estrutura de preferências multicritério que se obtém é uma pré-ordem completa (ver sec. 2.6), o que corresponde a uma ordenação com

eventuais *ex-aequo*. A aplicação método de Condorcet (ou regra da maioria) para agregar as ordens monocritério pode não resultar numa pré-ordem completa. Pretende, contudo, estabelecer mais convincentemente os casos em que uma acção a_1 deve ser preferida a uma acção a_0 (o que se representa por " $a_1 P a_0$ "). Segundo este método, $a_1 P a_0$ se o número de critérios segundo os quais a_1 surge posicionado à frente de a_0 é maior que o número de critérios em que sucede o contrário. Este método pode, no entanto, conduzir a ciclos de intransitividade (p. ex. $a_1 P a_0$, $a_2 P a_1$ e $a_0 P a_2$ constitui um ciclo de intransitividade).

Uma das razões pelas quais os métodos de Borda e de Condorcet são importantes é a de o primeiro poder ser considerado um precursor dos métodos conducentes a um critério de síntese, enquanto o segundo pode ser considerado um precursor dos métodos de prevalência [Vincke, 1989, p.140]. Estes métodos têm porém a limitação de desaproveitarem a informação de natureza cardinal da avaliação das alternativas segundo cada critério, quando esta está disponível.

Quando não existem limitações à recolha de dados, o conjunto de métodos que pode ser escolhido na fase de agregação de preferências é muito vasto. Uma classificação possível para estes métodos é a utilizada por [Roy, 1985], [Schärlig, 1985] e [Vincke, 1989], que distingue três classes (para uma classificação mais detalhada ver [Hwang e Yoon, 1981]):

- Abordagem do critério único de síntese excluindo incomparabilidade (Roy), também conhecidos por métodos de agregação completos (Schärlig), teoria da utilidade multiatributo (Vincke) ou métodos da escola americana;
- Abordagem da relação de prevalência de síntese aceitando incomparabilidade (Roy), também conhecidos por métodos de agregação parciais (Schärlig), métodos de prevalência (Vincke) ou métodos da escola francesa;
- Abordagem do critério julgamento local interactivo com interações de tentativa e erro (Roy), também conhecidos por métodos de agregação locais (Schärlig) ou métodos interactivos (Vincke).

Outra distinção frequentemente usada (p. ex. por [Hwang e Yoon, 1981]) diferencia entre os métodos "multiatributo" e os métodos "multiobjectivo". Os primeiros são dedicados a problemas com um conjunto de alternativas discreto e finito em que os "atributos" são explicitados. Os segundos destinam-se a problemas com um conjunto de alternativas (soluções) contínuo, definido através de restrições, em que os "objectivos" são explicitados. Porém, a terminologia empregue nesta classificação é pouco compatível com a utilizada neste texto (ver "critério", sec. 2.6).

No âmbito deste trabalho será suficiente estabelecer uma classificação a dois níveis, que pretende tipificar muitos dos métodos existentes, sem contudo pretender ser exhaustiva ou até mutuamente exclusiva. Num primeiro nível distinguem-se os métodos originalmente desenvolvidos no contexto da programação matemática (p. ex. programação

linear com múltiplos objectivos), dos métodos que pressupõem a existência de uma matriz de decisão. Os primeiros são interactivos⁴, no sentido em que alternam etapas de cálculo com etapas de diálogo e destinam-se maioritariamente a conjuntos de alternativas contínuos, embora muitos sejam adaptáveis ao caso discreto. O conjunto destes métodos é extremamente vasto, não sendo tratado neste texto. Para iniciar um estudo aprofundado ver [Vincke, 1989, Cap. 4], [Roy e Bouyssou, 1993, Cap. 7] e sobretudo [Steuer, 1986].

A um segundo nível distinguem-se os métodos que pressupõem a existência de uma matriz de decisão, o que os limita ao caso discreto, e fora do âmbito dos métodos interactivos acima referidos. A este nível faz-se uma distinção entre métodos conducentes a um critério de síntese e métodos conducentes a uma relação de prevalência de síntese (tal como fazem Roy, Schärli e Vincke), e acrescenta-se a classe dos métodos elementares. Esta última classe contém métodos simples e intuitivos que podem ser interessantes para a compreensão do paradigma do apoio multicritério à decisão e são úteis em situações específicas que podem ocorrer durante o processo de decisão, incluindo para além dos já referidos métodos de dominância, de Borda e de Condorcet, métodos como o maximin, o minimax, o conjuntivo, o disjuntivo ou o lexicográfico (ver descrição destes métodos em [Hwang e Yoon, 1981]). Esta classificação exclui métodos menos conhecidos que não se enquadram nas escolas americanas ou francesa e não podem ser considerados elementares, como por exemplo o TOPSIS, o LINMAP, o MDS, os métodos de permutação ou o método de afectação linear (ver ainda em [Hwang e Yoon, 1981] e também [Pomerol e Barba-Romero, 1993, Cap. 8]).

2.4 Métodos conducentes a um critério de síntese

Um dos métodos mais utilizados para agregar os desempenhos das alternativas segundo cada critério numa avaliação multicritério é o da média ponderada (ou soma ponderada). Considere-se que o desempenho de uma alternativa a_i segundo o j -ésimo critério é $g_j(a_i)$, o valor que se encontra na intersecção da linha i com a coluna j da matriz de decisão, e seja k_j (com $k_j \geq 0$) o coeficiente de ponderação, ou peso, do j -ésimo critério ($i=1, \dots, m; j=1, \dots, n$). Então o valor global de cada alternativa é dado por

$$V(a_i) = \sum_{j=1}^n k_j g_j(a_i), \text{ com } k_j > 0 \text{ e } \sum_{j=1}^n k_j = 1.$$

A popularidade deste método deriva por um lado da sua simplicidade e, por outro lado, da facilidade com que se obtêm recomendações a partir dos seus resultados. Por exemplo, é muito fácil indicar qual a melhor alternativa recomendada pelo método a partir do valor global das alternativas, sendo igualmente simples obter uma ordenação das alternativas por ordem decrescente do valor global.

⁴ Em relação às classificações de Roy, Schärli e Vincke omite-se o adjectivo "local" por se considerar que pode induzir em erro: o julgamento nas fases de diálogo não tem de ser local.

Todavia, o método encerra algumas falácias para o utilizador menos atento, a maior das quais se prende com a determinação dos coeficientes de ponderação. De facto, é muito comum a associação destes factores de ponderação com a importância relativa de cada critério, e a facilidade em obter estes números por parte do decisor. No entanto, o factor de ponderação associado a um dado critério é apenas um factor de conversão da escala do critério numa escala de valor global. O mesmo pode observar-se na equação acima, em que se verifica que, se houver uma mudança na ordem de grandeza das unidades na escala de medida de um critério, o valor da alternativa não permanece constante. A estes factores de ponderação está também associado o conceito de taxa de substituição (ver sec. 2.6), que deriva do carácter compensatório deste método: se o desempenho de uma alternativa segundo um critério g_j decrescer numa unidade, então basta acrescentar k_j/k_r ao desempenho da alternativa segundo um critério g_r para que o valor global da alternativa não se altere. Para este modelo, a quantidade k_j/k_r é a taxa de substituição do critério g_j em relação ao critério g_r , e não depende da alternativa considerada. O carácter compensatório deste método deve ser compreendido e aceite por quem o utiliza.

A determinação dos coeficientes de ponderação neste método é, pelos motivos apresentados, incompatível com julgamentos pouco fundados baseados na intuição da importância de cada critério. Existem vários métodos para determinar um conjunto de pesos, uns mais simples, outros mais bem fundados e adaptados ao carácter compensatório deste modelo. Destacam-se, entre muitos outros (ver [Pomerol e Barba-Romero, 1993], Cap. 4), o método de *tradeoff* [Keeney e Raiffa, 1976], o processo analítico hierárquico (AHP) [Saaty, 1980], o método UTA [Jacquet-Lagrèze e Siskos, 1982] e o método MACBETH [Bana e Costa e Vansnick, 1993].

O valor global de cada alternativa pode ser encarado como um critério que sintetiza os desempenhos segundo todos os critérios. A obtenção de um resultado tão útil quanto este caracteriza o conjunto de métodos ditos da escola americana, os métodos da teoria da utilidade multiatributo (MAUT, segundo o acrónimo em inglês), dos quais o método da média ponderada é um caso particular. A filosofia subjacente a estes métodos é a de que o decisor tenta maximizar uma função $V(g_1, g_2, \dots, g_n)$ que agrega todos os critérios considerados [Vinke, 1989, Cap 4]. Esta função permite concluir que 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 $V(a_1) = V(a_2)$. O papel principal do apoio à decisão é, neste caso, o da estimação da função $V(\cdot)$ a partir das respostas a questões judiciosamente colocadas ao decisor.

O critério de síntese pode ser de dois tipos, consoante inclua ou não incerteza sobre as consequências de cada alternativa. No caso de não haver incerteza, chama-se ao critério de síntese função valor, representado por $V(\cdot)$. A este tipo de função, em que se inclui a função do modelo da média ponderada, também se chama função de utilidade

ordinal e função de utilidade de Marshall. No caso de haver incerteza chama-se ao critério função utilidade, representado por $U(.)$. Também se encontram as designações de função de utilidade cardinal, função de utilidade probabilística e de função de utilidade de Von Neumann.

No caso de existir incerteza as consequências de cada alternativa são modeladas através de distribuições probabilísticas. Neste caso o problema torna-se complexo mesmo quando se considera apenas um critério (ver [Keeney e Raiffa, 1976], Cap. 4). Em termos muito gerais a avaliação de uma consequência x é expressa por um número (utilidade) relacionado com a probabilidade p tal que as duas acções seguintes sejam consideradas indiferentes pelo decisor:

- obter de certeza a consequência x ;
- participar numa lotaria de que pode resultar a melhor consequência possível, com probabilidade p , ou a pior consequência possível, com probabilidade $1-p$.

As alternativas são então avaliadas pela esperança da utilidade das suas consequências, pressupondo-se que o objectivo único do decisor é a maximização da utilidade esperada.

Excepto em casos muito simplificados, é impossível aferir directamente a forma de uma função utilidade (ou função valor). Por esse motivo, a prática da MAUT consiste geralmente num iterar de três etapas: na primeira etapa postulam-se alguns pressupostos sobre as preferências do decisor (p. ex. atitude em relação ao risco, independência entre diferentes critérios), a segunda etapa consiste na construção de formas funcionais para a função $U(.)$ sugeridas pelos pressupostos⁵, e a terceira etapa consiste na verificação da coerência entre os resultados previstos pelo modelo e as respostas do decisor a mais algumas questões hipotéticas.

Segundo Keeney e Raiffa (ver [Keeney e Raiffa, 1976, p.271]), após reiterar várias vezes as três etapas, é geralmente possível construir um modelo que não esteja em séria contradição com o sentimento do decisor. Em [Keeney, 1992], propõe-se também um processo iterativo com início numa "etapa zero", a estruturação dos objectivos, antecedendo as restantes três. Segundo este autor (ver [Keeney, 1992], p.141), quando não se verificam alguns pressupostos (geralmente sobre a independência dos critérios) deve rever-se a estruturação dos objectivos.

No âmbito da MAUT há um corpo de resultados que foca três questões principais: como inferir pressupostos mais fortes a partir da associação de pressupostos mais fracos, como construir funções analíticas a partir dos pressupostos verificados pelas preferências

⁵ A atitude face ao risco é útil para determinar a forma de uma função utilidade mono-critério, p. ex. se o decisor possuir uma aversão ao risco constante para um dado critério j então pode concluir-se que a função de utilidade para esse critério será da forma $U_j(x) = -e^{-cx}$. A independência entre critérios sugere funções de agregação multicritério, p. ex. se para o decisor os critérios possuírem independência aditiva então o modelo aditivo da média ponderada pode ser usado para expressar a utilidade multicritério $U(.)$ em função das utilidades por critério $U_j(.)$

do decisor e quais as questões a colocar ao decisor em cada circunstância. Entre a extensa bibliografia sobre este assunto destaca-se o trabalho pioneiro em [von Neumann e Morgenstern, 1947], uma obra de referência [Keeney e Raiffa, 1976], e uma obra mais recente e de linguagem mais acessível [Keeney, 1992].

2.5 Métodos conducentes a uma relação de prevalência de síntese

Os métodos ELECTRE III e PROMETHEE, abordados neste trabalho, pertencem à família dos métodos de prevalência, também conhecidos por métodos de subordinação ou por métodos da escola francesa. Esta família de métodos nasceu do trabalho de investigadores franceses, entre os quais se destaca Bernard Roy, iniciado nos anos sessenta. Segundo [Vincke, 1989, pp.85-86], os métodos de prevalência constituem um compromisso entre os métodos da MAUT (sec. 2.4), cuja riqueza de resultados não é garante de fiabilidade, e o método da dominância (sec. 2.3), muito fiável, mas demasiado pobre para ser útil. De facto, a riqueza dos resultados da MAUT 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 indicar qual a melhor ou concluir que são equivalentes. Porém, é admissível que para o decisor essas duas alternativas sejam incomparáveis, situação eventualmente provisória no decurso de um processo de decisão que evolui (como sugere [Vincke, 1989, pp.85]), resultante do facto do decisor não poder, não saber como, ou simplesmente não querer comparar as alternativas. No que respeita ao método da dominância, este assegura resultados tão fiáveis quanto os dados da matriz de decisão, mas não permite em geral estabelecer uma recomendação útil, porque não faz distinções entre as alternativas não dominadas.

Os 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 não conduz a um critério de síntese (pré-ordem completa), mas a uma relação binária (ver "modelação de preferências", sec. 2.6) chamada relação de prevalência. Este tipo de métodos divide-se em duas fases: em primeiro lugar constrói-se uma relação de prevalência de síntese agregando todos os critérios; em segundo lugar explora-se a relação de prevalência obtida tendo em vista a natureza da situação de decisão, ou seja, a problemática de decisão (ver sec. 2.6).

Assume-se que a estruturação da situação de decisão culminou no estabelecimento de um conjunto de acções potenciais (alternativas) e de uma família coerente de critérios (ver sec. 2.6). A primeira fase de um método de prevalência utiliza os dados da matriz de decisão e alguma informação suplementar para construir uma relação de prevalência. Esta relação define-se para pares de alternativas, que podem ser reais ou fictícias, 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 como a alternativa a_2 . A partir dessa relação podem deduzir-se três situações de preferência (ver "modelação de preferências", sec. 2.6). Na Figura 2.1, em que as setas indicam prevalência, exemplificam-se essas situações:

- a_1 é presumivelmente melhor que a_2 porque $a_1 S a_2$ e não se verifica $a_2 S a_1$;
- a_1 é indiferente a a_3 ($a_1 I a_3$ e $a_3 I a_1$) porque $a_1 S a_3$ e $a_3 S a_1$;
- a_2 é incomparável a a_3 ($a_2 R a_3$ e $a_3 R a_2$) porque não se verifica $a_2 S a_3$ nem $a_3 S a_2$;

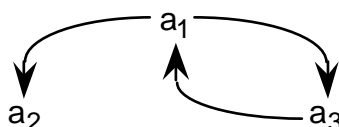


Figura 2.1 - Ilustração da relação de prevalência através de um grafo

As razões que justificam, dadas duas alternativas a_1 e a_2 , a afirmação de que a_1 prevalece sobre a_2 , dependem de método para método. Tipicamente verifica-se simultaneamente a satisfação de uma condição de concordância, que reúne os critérios que sustentam $a_1 S a_2$, e de uma condição de não-discordância, que considera os critérios que contrariam $a_1 S a_2$. A maioria dos métodos (uma exceção é o ELECTRE IV [Roy e Hugonard, 1982]) permite distinguir diferentes níveis de importância, ou "pesos", para os critérios na verificação das condições de concordância/não-discordância. Estes são de natureza muito distinta dos coeficientes de ponderação apresentados na secção 2.4.

A relação de prevalência define-se sobre um conjunto que depende da natureza da problemática de decisão (ver sec. 2.6). Tanto nos casos em que se pretende escolher a melhor alternativa, como nos casos em que se pretende ordenar as alternativas, define-se a relação para todos os pares de alternativas: para cada par possível de alternativas verifica-se se a primeira prevalece sobre a segunda e se a segunda prevalece sobre a primeira. No caso em que a problemática consiste em afectar cada alternativa a uma categoria definida *a priori*, a abordagem que tem sido seguida (ver [Roy e Bouyssou, 1993], sec. 6.1.3) consiste em não comparar as alternativas umas com as outras, mas sim com alternativas "de referência", que podem ser reais ou fictícias. As alternativas de referência devem definir cada categoria, podendo essa definição ocorrer de dois modos: ou as alternativas de referência definem as fronteiras entre categorias, ou definem o elemento típico de cada categoria. Na problemática de afectação, cada alternativa que se pretende categorizar é, portanto, comparada com todas as alternativas de referência.

O precursor dos métodos de prevalência, o ELECTRE I [Roy, 1968], construía uma única relação de prevalência (como na Figura 2.2a) para apoiar a decisão. Mais tarde, o ELECTRE II [Roy e Bertier, 1971] apresenta a construção de duas relações de prevalência (como na Figura 2.2b) que diferem na força dos argumentos exigidos para

concluir da prevalência de uma alternativa sobre outra. Uma exige argumentos mais fortes, enquanto a outra exige argumentos mais fracos, pelo que sempre que se verifica a prevalência "forte" também se verifica a prevalência "fraca". O método ELECTRE III [Roy, 1978] já apresenta uma estratégia distinta: ao comparar uma alternativa com outra tenta-se indicar até que ponto a primeira prevalece sobre a segunda, em vez de tentar decidir segundo a lógica binária sim/não. A relação de prevalência é portanto difusa, concretizando-se através de um grau de credibilidade da prevalência para cada par ordenado de alternativas (como na Figura 2.2c).

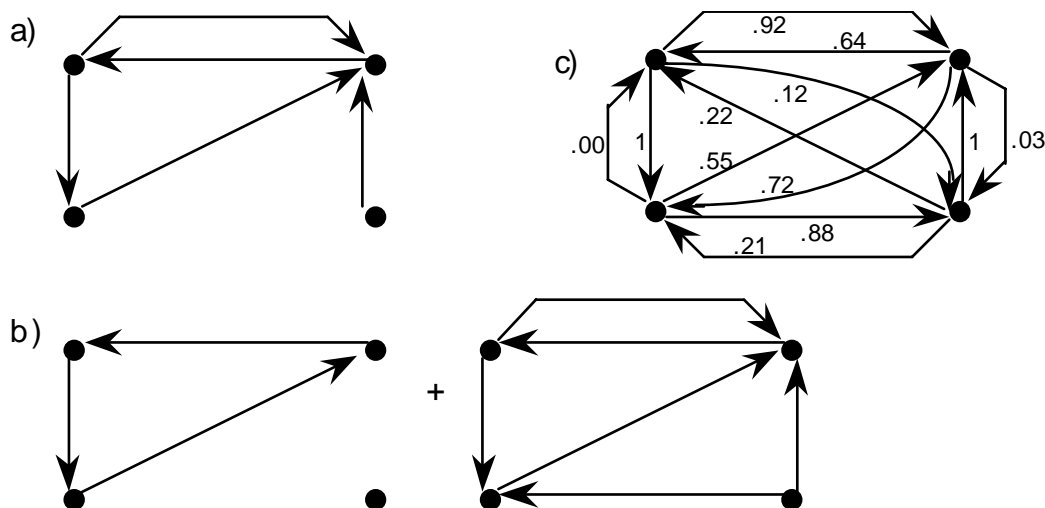


Figura 2.2 - Exemplos de relações de prevalência: a) única, b) múltiplas, c) difusa

Em conclusão, a fase de construção pode resultar num número de relações de prevalência superior a um ou mesmo indeterminado, podendo-se distinguir dois tipos de modelos (ver p. ex. [Roy, 1990b]):

- tipo 1: construção de um conjunto de r ($r \geq 1$) relações de prevalência; considerando uma ordenação das relações por ordem decrescente da força exigida aos argumentos para concluir que uma alternativa prevalece sobre outra (sendo S_1 a mais exigente e S_r a menos exigente) então pode afirmar-se que as r relações estão "encaixadas" umas dentro das outras, no sentido em que $S_i \subseteq S_{i+1} \subseteq \dots \subseteq S_r$, ou seja, $S_1 \subseteq S_2 \subseteq S_3 \dots \subseteq S_r$;
- tipo 2: construção de uma única relação de prevalência difusa, em que se associa a cada par de alternativas a_1 e a_2 um número $\sigma(a_1, a_2)$ que indica a credibilidade da afirmação " $a_1 \succ a_2$ ", em vez de dictomizar entre " $a_1 \succ a_2$ " e " $\text{não } a_1 \succ a_2$ "; a relação difusa pode originar um número indeterminado de relações do tipo 1, através da introdução de limiares de credibilidade.

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 (da melhor alternativa), para ordenação

das alternativas e para afectação das alternativas a categorias definidas *a priori*. Em qualquer dos casos a tarefa apresenta dificuldades devido à possível ocorrência de características pouco desejáveis como a incomparabilidade e a intransitividade (ver Figura 2.2), inexistentes nos métodos conducentes a um critério de síntese. Estas características tornam difícil a conciliação entre a operacionalidade (em termos de riqueza do resultado) das técnicas de exploração da relação de prevalência e a segurança que o decisor pode ter nos resultados obtidos.

Quando foi definida mais que uma relação de prevalência do tipo 1, utilizam-se sucessivamente as relações mais fracas, que são menos credíveis por serem menos exigentes, para refinar os resultados da exploração prévia das relações mais fortes. No caso de se ter definido uma relação de prevalência do tipo 2, pode optar-se por gerar múltiplas relações do tipo 1 através de níveis de corte no índice de credibilidade, e proceder como anteriormente. Estas são as estratégias seguidas pelos métodos da família ELECTRE, provavelmente os métodos de prevalência mais conhecidos. No seu conjunto formam uma família constituída pelos seguintes métodos:

- ELECTRE I [Roy, 1968]: utiliza uma relação de prevalência do tipo 1 para apoio à selecção da melhor alternativa;
- ELECTRE IS [Roy e Skalka, 1985]: utiliza uma relação de prevalência do tipo 1 para apoio à selecção da melhor alternativa, admitindo uma generalização da noção de critério (ver "pseudo-critério", sec. 2.6);
- ELECTRE II [Roy e Bertier, 1971]: utiliza duas relações de prevalência do tipo 1 para apoio à ordenação das alternativas;
- ELECTRE III [Roy, 1978]: utiliza uma relação de prevalência do tipo 2 para apoio à ordenação das alternativas, admitindo pseudo-critérios; será descrito em detalhe mais adiante;
- ELECTRE IV [Roy e Hugonard, 1982]: utiliza cinco relações de prevalência do tipo 1 para apoio à ordenação das alternativas, admitindo pseudo-critérios; não distingue diferentes importâncias entre os critérios;
- ELECTRE TRI [Yu, 1992] utiliza uma relação de prevalência do tipo 2 para apoio à afectação das alternativas a categorias definidas *a priori*, admitindo pseudo-critérios.

Um método que também é muito conhecido é o PROMETHEE [Brans, 1982], sobretudo nas suas variantes PROMETHEE I e PROMETHEE II, que serão descritas em pormenor mais adiante.

Termina-se esta secção referenciando alguns textos relevantes no âmbito dos métodos de prevalência. Sobre a construção de relações de prevalência veja-se [Roy, 1990b] e [Roy e Bouyssou, 1993, Cap. 5]. O tema da exploração é tratado, por exemplo, em [Vanderpooten, 1990] e [Roy e Bouyssou, 1993, Cap. 6]. [Maystre et al., 1994] apresenta o conjunto dos métodos ELECTRE e algumas aplicações. No que respeita ao método PROMETHEE veja-se [Brans e Mareshal, 1990], que agrega num só texto vários

desenvolvimentos deste método. Por fim, [Pomerol e Barba-Romero, 1993, Cap. 7] apresenta de forma sucinta os métodos ELECTRE e PROMETHEE, e ainda os métodos REGIME, ORESTE, TACTIC, MAPPAC, PRAGMA e MELCHIOR.

2.6 Algumas noções importantes

Esta secção, que pretende complementar as restantes secções deste capítulo, congrega um conjunto de algumas noções importantes, a que se faz apelo ao longo do texto, apresentado de forma enciclopédica⁶. Pretende-se que seja uma secção de consulta, que de algum modo possa orientar os leitores para um estudo mais aprofundado e simultaneamente especifique com maior rigor a linguagem utilizada. Tentou-se tanto quanto possível ser conciso, mas sem omitir a inclusão de terminologia oriunda das chamadas "escola francesa" e "escola americana".

Acção : Numa determinada situação de decisão podem discernir-se várias acções potenciais (para uma definição mais formal de acção ver [Roy e Bouyssou, 1993], p.64). Uma acção potencial é uma acção real ou fictícia provisoriamente julgada realista por pelo menos um actor. As acções são geralmente globais, isto é, mutuamente exclusivas (caso contrário denominam-se fragmentárias), pelo que se utiliza também o vocábulo alternativa. Perante um conjunto A de acções potenciais, que pode ser discreto (definido por enumeração ou em extensão) ou contínuo (definido em compreensão através de restrições), procede-se à sua avaliação utilizando os diversos pontos de vista, por intermédio de critérios.

Actor : Os actores num processo de decisão (ver [Roy e Bouyssou, 1993], p.64) são todos os indivíduos, ou grupos de indivíduos, que influenciam directa ou indirectamente a decisão. Aos que influenciam directamente a decisão poder-se-à chamar intervenientes, agentes de decisão ou decisores. Os actores não intervenientes influenciam indirectamente a decisão através dos interesses que possuem na mesma. Por exemplo, as decisões de um político são indirectamente influenciadas pelos seus potenciais eleitores que sejam afectados pelas consequências das suas acções. O processo de decisão é orientado por um analista (consultor, assessor), que frequentemente não pertence ao conjunto dos actores.

Agente de decisão : ver **actor**.

Alternativa : ver **acção**.

Apoio à decisão : Bernard Roy (ver [Roy, 1985], [Roy e Bouyssou, 1993], p.21) define apoio à decisão como uma actividade que, com apoio de modelos explícitos, ajuda a obter

⁶ No sentido etimológico do termo enciclopédia.

respostas às questões que se colocam aos intervenientes de um processo de decisão, com vista a recomendar (prescrever) comportamentos, em coerência com a evolução do processo de decisão e com o sistema de valores que os intervenientes representam. Esta definição remete para uma abordagem construtivista (ver p. ex. [Roy, 1993] e [Roy e Bouyssou, 1993], p.62), que não procura impor respostas, nem procura encontrar respostas escondidas na mente dos intervenientes. Procura, em vez disso, ajudar os intervenientes a construir um edifício de respostas, sempre provisórias, com base em preferências pouco estruturadas que evoluem ao longo de um processo de aprendizagem.

Crítério : Neste texto, critério designa (ver [Roy e Bouyssou, 1993], sec. 1.6) uma função que visa resumir as avaliações de uma acção sobre diversas dimensões que se possam reportar ao mesmo eixo de significação (ponto de vista). Como exemplo, Roy e Bouyssou apresentam a possível definição de um critério "danos ambientais" agregando as dimensões impacto sobre a fauna, sobre a flora, sobre a qualidade do ar, etc. Dado um conjunto A de acções potenciais (ou alternativas), e quaisquer duas acções a, b [A, um critério g é uma função real definida de tal modo que se $g(a) \geq g(b)$ então a acção a será "pelo menos tão boa como" a acção b, segundo o ponto de vista operacionalizado pelo critério. Ao critério g está associada uma escala pela qual se podem ordenar todas as alternativas (com possíveis *ex-aequo*). A noção de atributo está associada à de critério, podendo ser considerados como duas perspectivas, respectivamente medida de eficácia e característica mensurável, de um mesmo conceito. Os critérios ou atributos operacionalizam os pontos de vista, podendo definir o seu significado com maior precisão [Keeney, 1992, p.203] e permitindo a comparação dos desempenhos das acções potenciais segundo o ponto de vista. Os critérios (atributos) podem-se dividir em três categorias [Keeney, 1992, p.100]: naturais, construídos e indirectos (*proxy*, ou "por procuração"). Idealmente deverão ser critérios (atributos) naturais, cuja interpretação é objectiva, como por exemplo "velocidade máxima em km/h" para um ponto de vista "velocidade do automóvel". Os critérios (atributos) construídos expressam-se através de uma aplicação que atribui um determinado nível a uma determinada descrição, como na Tabela 2.1. Os critérios (atributos) indirectos constituem uma medida indirecta para um ponto de vista para o qual não existe um critério natural (ver também [Keeney e Raiffa, 1976], sec. 2.5). Por exemplo, o ponto de vista "prestígio da empresa" pode ser indirectamente medido pelo critério indirecto "quota de mercado presente", que por sua vez também pode medir indirectamente o ponto de vista "lucros futuros da empresa". Um aspecto frequentemente esquecido é o de que a selecção dos critérios (atributos) é uma parte importante da expressão dos valores dos actores [Keeney, 1992, p.118].

| Nível de g(.) | Descrição |
|---------------|---|
| 0 | experiência inferior ou igual a 1 ano, ou superior mas não relevante para o cargo |

| | |
|---|--|
| 1 | experiência superior a 1 ano e inferior a 3 anos em empresa do ramo ou função idêntica |
| 2 | experiência igual ou superior a 3 anos em empresa do ramo ou função idêntica (mas não cumulativamente) |
| 3 | experiência igual ou superior a 3 anos em empresa do ramo e em função idêntica |

Tabela 2.1 Definição de um critério construído para o ponto de vista "experiência relevante para o lugar" (num contexto de contratação)

Decisor : ver actor.

Estrutura de preferências : ver **modelação de preferências**.

Família coerente de critérios : O conjunto de critérios (ver **critério**) utilizados numa situação de decisão deverá permitir modelar preferências a um nível global (multicritério). Para isso é desejável que satisfaçam os três axiomas que definem uma família coerente de critérios [Roy e Bouyssou, 1993, Cap. 2]: exaustividade, coerência entre preferências monocritério e multicritério (axioma de coesão), e não redundância. O axioma da exaustividade requer que duas acções sejam consideradas equivalentes se os seus desempenhos forem iguais em todos os critérios. Tal não se verifica quando há algum critério pertinente esquecido. O segundo axioma, da coesão, implica *grosso modo* que se uma acção a' for em tudo igual a uma acção a , exceptuando o desempenho num critério j em que $g_j(a') > g_j(a)$, então a acção a' não poderá nunca ser considerada inferior à acção a (considerando todos os critérios). O último axioma requer que não haja critérios redundantes.

Interveniente : ver actor.

Matriz de decisão: Quando estiverem definidos os critérios (ver **critério**) e as acções a considerar pode construir-se uma matriz de decisão (ou de impacto, ou de desempenho) se o conjunto das acções potenciais for discreto. 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:

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

Modelação de preferências : A modelação de preferências pretende ser uma ferramenta no apoio à decisão para tomar em consideração as preferências dos actores num processo de decisão (ver p. ex. [Vincke, 1989], Cap. 2, e [Roy e Bouyssou, 1993], sec. 1.4). Um conceito base para a modelação de preferências é o da relação binária sobre um conjunto A de acções. Uma relação binária H é um subconjunto de $A \times A$ que contém os pares

ordenados (a,b) tais que $a H b$ ($a,b \in A$). As relações binárias podem possuir algumas propriedades, como as seguintes:

- reflexividade: $\forall a \in A, a H a$;
- irreflexividade: $\forall a \in A, \sim(a H a)$;
- simetria: $\forall a,b \in A, a H b \iff b H a$;
- assimetria: $\forall a,b \in A, a H b \iff \sim(b H a)$;
- transitividade: $\forall a,b,c \in A, a H b \iff b H c \iff a H c$.

As situações de preferência são relações binárias. As quatro relações fundamentais e incompatíveis são (segundo [Roy e Bouyssou, 1993], p.33):

- indiferença (relação simétrica reflexiva), representada por **I**, p. ex. $a I b$ corresponde à existência de razões que justificam uma indiferença entre as acções a e b ;
- preferência estrita (relação assimétrica irreflexiva), representada por **P**, p. ex. $a P b$ corresponde à existência de razões que justificam o favorecimento de a em relação a b ;
- preferência fraca (relação assimétrica irreflexiva), representada por **Q**, p. ex. $a Q b$ corresponde a uma hesitação entre $a I b$ e $a P b$;
- incomparabilidade (relação simétrica irreflexiva), representada por **R**; p. ex. $a R b$ corresponde à ausência de razões que justifiquem uma das situações anteriores.

As estruturas de preferências sobre um conjunto A de acções potenciais são constituídas por um conjunto de relações binárias (H_1, \dots, H_n) , adequadas à representação das preferências dos actores e que satisfazem as exigências de exaustividade e exclusão mútua, i.e., $\forall a,b \in A \exists^1 i : a H_i b$. Apresentam-se de seguida como ilustração duas das mais importantes estruturas de preferência.

Um par de relações binárias (T, V) exaustivas e mutuamente exclusivas, tais que T é assimétrica e transitiva e V é simétrica e transitiva define uma estrutura de pré-ordem completa (ou pré-ordem total). A esta estrutura corresponde a noção intuitiva de classificação com possibilidade de *ex-aequo*. Um conjunto de três relações binárias (T, V, W) exaustivas e mutuamente exclusivas, tais que T é assimétrica, V é simétrica e reflexiva, W é simétrica irreflexiva e $(T \approx V)$ é transitiva define uma estrutura de pré-ordem parcial. Esta estrutura generaliza a pré-ordem completa incorporando a ideia de incomparabilidade. Podem ainda ser definidas estruturas de relações com valores, ou estruturas para comparação de desníveis (*écarts*) sobre $A \times A$ (ver [Vincke, 1989], Cap. 2, e [Roy e Bouyssou, 1993], sec. 1.4).

Objectivo : ver **ponto de vista**.

Ponto de vista : Um ponto de vista é a explicitação de um valor, representando um aspecto da decisão apercebido como importante para a construção de um modelo de avaliação (ver [Bana e Costa, 1992], [Bana e Costa e Ferreira, 1994]). Os valores de um actor também podem ser explicitados, para cada situação de decisão, à custa de objectivos, que são a expressão de algo que se deseja o mais possível ou o menos possível

[Keeney, 1992, pp.33-34]. Os objectivos e os pontos de vista podem ser interpretados como duas perspectivas distintas para explicitar valores, para um determinado contexto de decisão. O objectivo tem, sobre o ponto de vista, a informação adicional da direcção de preferência (maximização ou minimização). Por exemplo, a administração de determinada empresa pode ter como um valor fundamental a qualificação do seu pessoal. Um ponto de vista que explicita o valor numa situação de escolha de um novo colaborador é a qualificação presente e potencial do colaborador a contratar, enquanto o objectivo correspondente é maximizar essa qualificação. Os pontos de vista podem ser elementares (PVE) ou fundamentais (PVF). Os PVF (objectivos fundamentais) derivam de uma razão essencial para o interesse na decisão, e reflectem um valor fundamental isolável dos restantes. Os PVE não reflectem valores fundamentais nem isoláveis (geralmente), contribuindo para um ou vários PVF [Bana e Costa e Ferreira, 1994]. Segundo [Keeney, 1992, sec. 3.5], os objectivos fundamentais utilizados numa situação de decisão devem ser, no seu conjunto, essenciais, controláveis, completos, mensuráveis, operacionalizáveis, decomponíveis, não redundantes, concisos e compreensíveis.

Pré-ordem completa ou parcial : ver modelação de preferências.

Problemáticas de apoio à decisão : Deve-se a Bernard Roy a definição de quatro problemáticas⁷ de apoio à decisão (ver p. ex. Roy e Bouyssou, 1993], pp.69-70):

- problemática α (selecção): o objectivo do processo de decisão é a recomendação da escolha de uma única acção;
- problemática δ (descrição): o objectivo do processo de decisão é a recomendação de uma descrição das acções ou suas consequências através de uma linguagem apropriada;
- problemática γ (ordenação): o objectivo do processo de decisão é a recomendação de uma ordenação (seriação) das acções;
- problemática β (afecção): o objectivo do processo de decisão é a recomendação de uma afecção das acções em categorias (classes) pré-estabelecidas, ordenadas ou não.

Os objectivos de cada problemática nem sempre são atingidos. Por exemplo, a problemática de selecção resulta frequentemente num subconjunto de acções contendo mais do que uma acção. Esse subconjunto será em princípio tão restrito quanto possível e conterà a melhor acção a recomendar. Note-se que, no entanto, não se poderá concluir que o subconjunto contém as melhores acções a recomendar, isto é, a segunda melhor alternativa pode não estar incluída no subconjunto. Recentemente, [Bana e Costa, 1994] propôs um enriquecimento desta classificação, mas para os fins em vista neste trabalho o grau de detalhe da classificação de Roy é suficiente.

⁷ Adota-se o vocábulo francês "problématique", usado por B. Roy, apesar de este sugerir a existência de um "problema" de decisão (ver a este propósito [Keeney, 1992], que defende o uso de situação de decisão ou até oportunidade de decisão).

Pseudo-critério : Um pseudo-critério é uma função que operacionaliza um ponto de vista. A essa função g estão associados um limiar de indiferença e um limiar de preferência. Considerem-se duas alternativas a_1 e a_2 tais que $g(a_1) \geq g(a_2)$ e seja $\Delta = g(a_1) - g(a_2)$. Ao utilizar-se um pseudo-critério considera-se que as alternativas a_1 e a_2 são indiferentes segundo aquele ponto de vista se Δ for menor ou igual que o limiar de indiferença e só se conclui que a_1 é preferível a a_2 se Δ for maior que o limiar de preferência. O limiar de preferência tem de ser sempre superior ou igual ao limiar de indiferença.

Situação de preferência : ver **modelação de preferências**.

Taxa de substituição : A taxa de substituição define-se para modelos compensatórios, ou seja, modelos em que o piorar de um desempenho segundo um critério (ver **critério**) possa ser compensado pelo melhorar do desempenho segundo um critério diferente. À quantidade $t_{jr}(a)$ que se deve acrescentar⁸ ao desempenho de uma alternativa a segundo um critério de referência g_r , para compensar uma perda de uma unidade no desempenho de a segundo outro critério g_j , chama-se taxa de substituição do critério g_j em relação ao critério g_r em a . Deste modo, se houver duas alternativas, a e b , com o mesmo desempenho a todos os critérios, à excepção de dois (sejam estes g_1 e g_2), com $g_2(b) = g_2(a) - 1$ e $g_1(b) = g_1(a) + t_{21}(a)$ então pode concluir-se que as alternativas a e b são indiferentes.

Valor : A palavra "valor" é neste texto utilizada num sentido mais amplo que o de um princípio ético. Por valores de um actor entendem-se todos os princípios que este utiliza para avaliar as consequências actuais e potenciais de uma acção ou da inacção, de uma alternativa proposta ou de uma decisão [Keeney, 1992, p.6].

⁸ Esta definição pressupõe que o decisor pretende maximizar os critérios.

3 - PROCESSAMENTO PARALELO

3.1 Introdução

Este capítulo pretende constituir uma apresentação do ambiente computacional utilizado (sec. 3.6), enquadrado através de uma breve introdução ao processamento paralelo (sec. 3.1 a 3.5).

As aplicações informáticas têm incessantemente vindo a exigir dos computadores maior velocidade de processamento. Os computadores actuais, baseados em tecnologia de semi-condutores, padecem de limites de desempenho que estão a ser rapidamente aproximados. Dado que tecnologias alternativas (por exemplo ópticas) ainda são demasiado caras e carecem de mais testes, uma abordagem cada vez mais popular é a de utilizar mais do que um processador para executar os programas através de processamento paralelo. Para esta popularidade contribuem também o baixo preço dos processadores habitualmente utilizados e as economias de escala tornadas possíveis ao utilizar vários processadores a partilhar os restantes componentes do computador [Zenios, 1992].

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 como a IBM, Intel, Sun, Oracle ou Microsoft [Pountain e Bryan, 1992]. O potencial do processamento paralelo é suficiente para que [Zenios, 1992] considere que irá ser o paradigma dominante para computação em larga escala na próxima década, e talvez seguintes.

O processamento paralelo, contudo, envolve uma complexidade muito superior ao clássico compromisso entre portabilidade e eficiência do código, que se tornam objectivos ainda mais conflituosos. A diversidade de arquitecturas paralelas é muito grande e a portabilidade é um *desideratum* longe de ser alcançado.

A utilização de processamento paralelo implica a existência de um novo "paradigma de abordagem" [Cole, 1992]: partição do problema em actividades executáveis em paralelo, escalonamento das actividades e coordenação entre actividades. A impossibilidade de obter partições perfeitas, os custos de escalonamento e os custos de coordenação (partilha de dados e sincronização) limitam, para cada aplicação, o aproveitamento do poder computacional disponível numa máquina paralela. Por outro lado, o desenvolvimento de aplicações paralelas torna-se relativamente lento e sujeito a erros devido à complexidade envolvida.

Tais dificuldades não obstaram a que se desenvolvessem com sucesso muitas aplicações utilizando processamento paralelo. A maioria dessas aplicações lida com

problemas típicos de supercomputadores, como dinâmica de fluidos, análise molecular, aerodinâmica ou sistemas de tempo-real. Existem também aplicações a problemas de investigação operacional e ciências da gestão, como por exemplo programação linear e não-linear, programação dinâmica, simulação, optimização em redes e escalonamento de tarefas (referidas por [Adams, 1990]).

A secção seguinte apresenta uma classificação dos diversos tipos de computadores paralelos, estabelecendo uma distinção entre máquinas paralelas e outros sistemas onde existe paralelismo. Na secção 3.3 apresentam-se alguns dos mais importantes conceitos de processamento paralelo, no âmbito do paradigma MIMD. As secções 3.4 e 3.5 apresentam computadores MIMD, respectivamente de memória partilhada e de passagem de mensagens. Os primeiros constituem o tipo de computadores paralelos mais em voga na actualidade, enquanto os segundos são em geral mais escaláveis⁹ e incluem a máquina utilizada para este trabalho. O ambiente utilizado, nas suas vertentes de *hardware* (suporte físico) e *software* (programas), é descrito na secção 3.6, que conclui este capítulo.

3.2 Esquema de classificação

Não existem definições unanimemente aceites para processamento paralelo ou computador paralelo. Ao termo "processamento paralelo" está associada uma perspectiva do programador, enquanto o termo "computador paralelo" (ou arquitectura paralela) remete para uma perspectiva de arquitecto de hardware.

Como definição de trabalho considera-se processamento paralelo a execução simultânea de instruções (que podem ser idênticas) por múltiplos elementos processadores. Um computador paralelo será uma máquina com múltiplos elementos processadores, capaz de suportar processamento paralelo.

Dada a miríade de sistemas computacionais existentes ou propostos, alguns autores propõem taxinomias para facilitar a comunicação e definição de conceitos (ver p. ex. [Ryan, 1991]). Deve-se a [Flynn, 1966] (ver também [Flynn, 1972]) a taxinomia mais divulgada, cuja longevidade se deve ao facto de ser independente de qualquer tecnologia. Flynn divide os computadores segundo o número de diferentes fluxos de instruções e fluxos de dados que as máquinas podem processar ao mesmo tempo:

- computadores SISD (*Single Instruction stream, Single Data stream*) processam um fluxo de instruções e um fluxo de dados de cada vez, correspondendo portanto à arquitectura sequencial clássica de von Neumann (uniprocessador);

⁹ Em poucas palavras pode definir-se escalabilidade de um computador como a capacidade de oferecer maiores desempenhos à medida que se aumenta o número de processadores (ver sobre este assunto [Nussbaum e Agarwal, 1991]).

- computadores SIMD (*Single Instruction stream, Multiple Data stream*) processam de cada vez um só fluxo de instruções, mas podem operar simultaneamente sobre vários fluxos de dados que requerem as mesmas operações;
- computadores MISD (*Multiple Instruction stream, Single Data stream*) podem processar simultaneamente vários fluxos de instruções que operam sobre o mesmo fluxo de dados; na prática só existem computadores deste tipo para aplicações muito específicas, como o processamento de sinal;
- computadores MIMD (*Multiple Instruction stream, Multiple Data stream*) podem processar em paralelo múltiplos fluxos de instruções, cada um operando sobre um fluxo de dados distinto; estão nesta categoria as máquinas mais "genuinamente" paralelas.

A vantagem da independência da tecnologia, a que se deve o sucesso desta taxinomia, contribui, contudo, para que seja de difícil uso na distinção entre as muitas arquitecturas propostas posteriormente. Por exemplo, os computadores vectoriais, como por exemplo o CRAY-1, poderão ser considerados SISD porque um vector é um dado unitário do ponto de vista do hardware (veja-se [Riganati e Schneck, 1984]). No entanto, haverá certamente opiniões que defendam que um CRAY-1 é um SIMD.

Para ultrapassar esta limitação alguns autores (ver ainda [Ryan, 1991]) propuseram outras taxinomias que se referem explicitamente às tecnologias utilizadas. O esquema de classificação que orienta este capítulo, inspirado na taxinomia usada por [Duncan, 1990], é ilustrado na Figura 3.1. Neste esquema verifica-se que não se consideram como sendo computadores paralelos os sistemas com mecanismos de paralelismo de baixo nível, os computadores VLIW e os sistemas distribuídos.

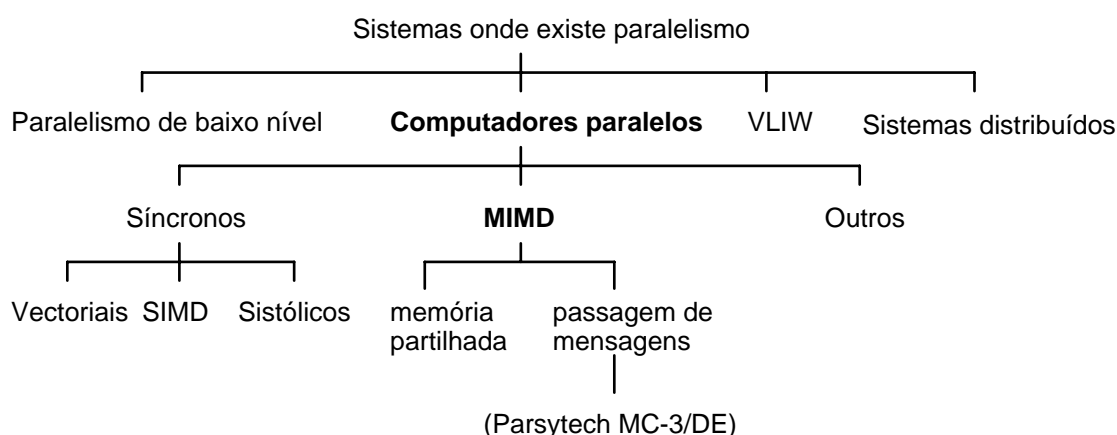


Figura 3.1 - Esquema de classificação

Apresentam-se de seguida algumas notas sobre os diversos ramos deste esquema de classificação, com excepção dos computadores MIMD. Estes serão apresentados em maior detalhe nas secções seguintes.

Os mecanismos de paralelismo de baixo nível visam obter maiores desempenhos para um processador, efectuando algumas "sub-instruções" em paralelo. Disso são exemplo computadores com co-processadores matemáticos, ou com co-processadores de entrada/saída, ou mesmo possuidores de processadores com unidades funcionais distintas¹⁰ e processadores com encadeamento (*pipelining*) interno¹¹. O processador, do ponto de vista do software, é um único elemento. Por outro lado, estes mecanismos estão de tal modo vulgarizados que seria pouco útil considerar como paralelo todo o computador que os possua.

Também não se consideram computadores paralelos os computadores de arquitectura VLIW (*Very Long Instruction Word*¹²). Esta arquitectura caracteriza-se por possuir uma palavra de instrução mais longa que o normal e por possuir várias unidades aritméticas e lógicas (ULA). A palavra de instrução é muito longa porque tem de conter instruções para todas as ULA da máquina. A eficiência desta arquitectura é no entanto limitada por dependências de dados e conflitos no acesso à memória entre instruções diferentes na mesma palavra de instrução. Estas limitações são aliviadas à custa da complexidade dos compiladores, que recorrem a técnicas variadas para as tentar ultrapassar. A eficiência do código dependerá da natureza da aplicação (quanto mais irregular for o código, melhor) e do desempenho do compilador. Um texto que apresenta estes conceitos pode encontrar-se em [Wayner, 1989].

Por fim, não se considera que um sistema distribuído seja um computador paralelo, mas sim um conjunto de computadores, cada um dos quais pode ser paralelo ou não. Podem, no entanto, ser programados como se fossem um computador MIMD de passagem de mensagens (ver sec. 3.6). A principal diferença residirá no poder computacional de cada elemento processador e na velocidade de comunicação entre esses elementos. Existem aplicações muito interessantes para estes sistemas, baseados no conceito de máquina paralela virtual. Este conceito consiste em encarar o sistema distribuído como uma única máquina virtual, aproveitando para isso a capacidade não utilizada de cada computador na rede. No âmbito deste conceito são populares os ambientes de computação PVM e HeNCE [Dongarra, 1993], de domínio público, bem como o Linda [Gelernter, 1988], cuja utilização não se limita a sistemas distribuídos, começando já a ser oferecidas versões para máquinas MIMD.

De entre computadores paralelos, os chamados síncronos possuem vários elementos processadores perfeitamente sincronizados através de um relógio central, de uma unidade de controlo central ou de unidades de controlo vectoriais [Duncan, 1990].

¹⁰ Processadores cuja unidade aritmética e lógica opera em paralelo com outras unidades funcionais, como por exemplo a unidade de cálculo em vírgula flutuante.

¹¹ A execução de uma instrução pode ser dividida em sub-instruções como "buscar instrução", "descodificar instrução", "buscar operandos", "alterar operandos" e "guardar operandos". Estas sub-instruções podem ser encadeadas como numa cadeia de montagem, de modo a que enquanto se processa uma fase de uma instrução se possa processar uma fase anterior da instrução seguinte.

¹² Tradução literal: Palavra de instrução muito longa.

Distinguem-se neste conjunto as arquitecturas SIMD, os computadores vectoriais e os vectores sistólicos. Estes últimos são os menos divulgados, sendo apenas aplicados a problemas muito específicos (p. ex. compressão de dados, ver [Pountain e Bryan, 1992], pag. 115). Nos vectores sistólicos, os dados são ritmicamente bombeados (por analogia com o bombear do sangue pelo coração) da memória para um vector de processadores especializados, cada um dos quais efectua uma operação distinta sobre cada dado antes de o enviar para o processador seguinte (ver [Kung, 1982]).

Entre os computadores paralelos síncronos, os mais conhecidos serão talvez os computadores vectoriais (p. ex. Cray, CDC, Fujitsu, Hitachi, NEC [Riganati e Schneck, 1984]), que alcançaram algum sucesso comercial. A principal característica destes computadores é a de permitirem operações elementares sobre vectores, conseguindo-se efectuar em simultâneo uma mesma operação sobre múltiplos elementos de um ou vários vectores. A primeira máquina deste tipo a ser amplamente divulgada foi a Cray-1, em 1976 (ver [Russel, 1978]), que possuía um só processador. Actualmente existem computadores vectoriais com mais que um processador, que arquitecturalmente podem ser considerados MIMD, mas que para o programador se assemelham a um SIMD, realizando o modelo SPMD (*Single Program Multiple Data* / Um Programa - Múltiplos Dados). Aliás, a designação de computador vectorial tenderá a cair em desuso à medida que os elementos processadores nos MIMD começam a incluir unidades vectoriais.

O conjunto dos computadores paralelos síncronos completa-se com os computadores SIMD. Os mais conhecidos são os computadores matriciais, entre os quais se conta o primeiro computador paralelo, o ILLIAC IV [Barnes et al., 1968], constituído por 64 processadores (dispostos numa rede matricial 8 x 8). Neste tipo de computadores cada instrução opera sobre todos os elementos de uma matriz de dados (um elemento por processador). Um exemplo notável deste tipo de máquina é a Connection Machine da TMC (ver [Hillis, 1985] ou [Tucker e Robertson, 1988]) que possui até 65536 elementos processadores muito simples, que podem ainda simular um número superior de processadores virtuais. Um tipo distinto de SIMD é a arquitectura com memória associativa (veja-se [Duncan, 1990], [Kohonen, 1987]). Nestas arquitecturas as instruções operam sobre todas os dados armazenados na chamada memória associativa cujo conteúdo satisfaça determinadas condições. São sobretudo usadas em aplicações orientadas para bases de dados.

Por fim, refira-se a existência de muitas outras arquitecturas, de difícil classificação, que podem ser consideradas paralelas. Entre essas arquitecturas destacam-se os computadores de fluxo de dados, as redes neuronais, as arquitecturas de redução, os vectores de propagação de onda e os computadores MIMD/SIMD. Descrições destas arquitecturas podem encontrar-se em [Dennis, 1980], para computadores de fluxo de dados, [Shumsheruddin, 1992], para redes neuronais, e [Duncan, 1990], para os restantes.

3.3 Processamento paralelo em arquitecturas MIMD

As arquitecturas paralelas MIMD permitem a execução simultânea de vários fluxos de instruções, que podem operar simultaneamente sobre dados distintos. Nesta secção introduzem-se alguns conceitos de programação paralela, no contexto do paradigma MIMD, distinguem-se multiprocessadores de memória partilhada de multiprocessadores de passagem de mensagens (também conhecidos por multicomputadores), e apresentam-se algumas medidas para o desempenho de aplicações paralelas.

Execução concorrente de fluxos de instruções

Uma aplicação, ou programa (num sentido lato), é composta por vários processos, por vezes executáveis em paralelo. Num computador sequencial, os processos partilham o único processador disponível, utilizando pseudo-parallelismo (também se utilizam as expressões *time-sharing*, concorrência e partilha no tempo). Num computador MIMD, os processos podem ser executados em paralelo em diferentes processadores, sem prejuízo de haver vários processos a partilhar cada um dos processadores. Alguns computadores ainda suportam a existência de vários fios de processamento por processo. Cada fio de processamento corresponde a um fluxo de instruções que é executado em pseudo-parallelismo com os restantes fios do processo, partilhando as áreas de dados e de código do processo, mas mantendo pilhas privadas.

A divisão do tempo de processamento pelos diversos processos e, eventualmente, pelos diversos fios de processamento (doravante usar-se-á a expressão "entidade de processamento", ou apenas "entidade", para designar indiferenciadamente um processo ou um fio) no estado activo constitui o problema do escalonamento. Uma entidade de processamento deixa de estar activa quando termina ou quando bloqueia à espera de algum acontecimento (p. ex. a libertação de um recurso). Assim que o acontecimento ocorra a entidade volta ao estado activo. Em alguns computadores, o escalonamento pode distinguir vários níveis de prioridade, em esquemas que podem ser preemptivos¹³ ou não-preemptivos. Nos esquemas com prioridades, a próxima entidade a ocupar o processador será escolhida entre as de maior nível de prioridade.

Comunicação entre entidades de processamento no mesmo processador

A comunicação entre entidades de processamento é necessária para que as estas se possam coordenar e cooperar entre si na resolução do problema computacional. Considera-se, de momento, que as entidades estão localizadas no mesmo processador, o

¹³ Por preempção entende-se a característica de um processo já iniciado poder ser interrompido, para que outro processo ocupe o processador. O processo interrompido é posteriormente continuado ou re-iniciado.

que sucede, por exemplo, na comunicação entre os diversos fios de processamento do mesmo processo. Nestes casos, a comunicação efectua-se através de memória partilhada, isto é, duas entidades de processamento podem comunicar através da escrita e leitura de uma área de memória que ambas podem endereçar. O emissor escreve numa área de memória comum o valor que deseja comunicar, e esse valor é "recebido" quando o receptor efectuar uma leitura na mesma área.

Frequentemente, a memória partilhada surge ao programador sob a forma de variáveis partilhadas, o que torna a programação mais simples. Existe, no entanto, necessidade de proteger o acesso às variáveis partilhadas. Por exemplo, não pode haver duas entidades a escrever simultaneamente valores diferentes na mesma variável partilhada. Considere-se o conjunto A de instruções de uma destas entidades que escreve um valor na variável partilhada, e seja A' um conjunto de instruções, que também escreve na mesma variável partilhada, pertencente ao código de outra entidade. Claramente, enquanto A estiver a ser executado, a execução de A' deve ser impedida, e vice-versa. Chama-se por isso a A e A' zonas de exclusão mútua ou zonas críticas do código.

Um mecanismo muito popular para proteger as zonas críticas, isto é, para impedir a violação da exclusão mútua, é o semáforo [Dijkstra, 1968]. O semáforo de Dijkstra consiste num mecanismo para incrementar (*Signal*) ou decrementar (*Wait*) uma variável de forma atómica¹⁴, causando o bloqueio das entidades que tentem decrementar essa variável quando esta contém o valor zero. Uma entidade bloqueada num semáforo só é desbloqueada quando a variável é incrementada, ou seja, quando outra entidade executar uma instrução *Signal*. A utilização de semáforos deve ser cuidadosa para que não haja impasses (*deadlocks*). [Ben-Ari, 1982] apresenta estes conceitos e seu uso detalhadamente.

Comunicação entre entidades de processamento em processadores distintos

Um dos mais importantes aspectos a considerar no paradigma MIMD é o da comunicação entre os diversos processadores. Distinguem-se duas estratégias de comunicação: através de escrita/leitura de variáveis partilhadas e através da transmissão/recepção de mensagens (passagem de mensagens).

Na comunicação por intermédio de variáveis partilhadas, as entidades de processamento podem endereçar uma área de memória comum. Conseguem assim comunicar de um modo extremamente flexível, podendo mesmo simular a comunicação por passagem de mensagens. A memória física endereçada pode ser centralizada ou distribuída pelos vários processadores. Em qualquer dos casos, convencionou-se chamar

¹⁴ Diz-se que uma operação é atómica (no sentido etimológico do termo) quando não é decomponível em sub-operações, o que implica que não pode ser interrompida: se for executada é-o sem interferência de outras operações.

aos computadores cujos processadores comunicam desta forma multiprocessadores de memória partilhada ou MIMD fortemente ligados (ver sec. 3.4).

O outro mecanismo muito utilizado para permitir a comunicação entre entidades de processamento em processadores distintos¹⁵ é a passagem de mensagens [Gentleman, 1981]. Quando uma determinada entidade pretende enviar informação a outra, envia uma mensagem (instrução de emissão) através de um canal de comunicação. A entidade receptora só receberá a mensagem quando executar uma instrução de recepção referente a esse canal. Se a entidade emissora ficar bloqueada até que o receptor execute a instrução de recepção, a comunicação diz-se síncrona e o canal diz-se sem memória. Caso contrário, diz-se assíncrona, e ao canal está associada memória para guardar a mensagem, já que a entidade emissora prossegue a sua execução. Chama-se às máquinas em que os processadores comunicam desta forma multiprocessadores de passagem de mensagens, multicomputadores, ou computadores MIMD fracamente ligados (ver sec. 3.5). Os processadores estão ligados por ligações físicas nas quais estão mapeados os canais de comunicação lógicos. Nas arquitecturas mais evoluídas o mapeamento pode comportar vários canais lógicos por ligação física.

Programação

A programação de computadores MIMD é relativamente complexa e dependente da máquina particular que se está a utilizar. Há que encontrar um compromisso para a divisão de um problema em unidades executáveis em paralelo, considerando o tempo de processamento e a carga adicional de comunicação e controlo. As ferramentas, tais como compiladores, oferecidas em cada ambiente de desenvolvimento, determinam a intervenção que o programador tem na exploração do paralelismo nas aplicações. Essa intervenção ocorre a determinado nível de abstracção, conforme se exemplifica de seguida (veja-se também [Cole, 1992]).

Nos paradigmas de mais elevado nível de abstracção, as tarefas de paralelização são deixadas à responsabilidade do compilador. Incluem-se neste paradigma a programação funcional, a programação lógica, o paralelismo de dados e, sobretudo, a paralelização automática de código por análise de ciclos e de dependências de instruções.

Num patamar intermédio encontram-se os modelos de computadores paralelos ideais (ver p. ex. [Chaudhuri, 1982], Cap. 2), como os modelos de memória partilhada ou de comunicação por mensagens. Cabe aos compiladores a tarefa de realizar estes modelos em arquitecturas reais.

Entre estes paradigmas a um nível mais baixo de abstracção podem incluir-se as redes de cálculo indiferenciado, RCI, (ver p. ex. aplicações em [Costa e Clímaco, 1993] e [Dias et al., 1993]). As RCI consistem num conjunto de processadores "escravos" a operar

¹⁵ As entidades executadas no mesmo processador também podem comunicar através de passagem de mensagens, mas nesse caso o mecanismo é simulado através de uma área de memória partilhada.

sob o controlo de um processador "mestre", com o qual comunicam por passagem de mensagens. Neste paradigma o programador é liberto da tarefa de encaminhar as mensagens, o que o torna relativamente portátil, mas à custa de um desempenho inferior (ver p. ex. experiência em [Dias et al., 1993]).

Ao mais baixo nível de abstracção situam-se os chamados paradigmas *ad-hoc*. Nestes casos, apenas se disponibiliza uma biblioteca de funções que permitem a realização de aplicações paralelas utilizando uma linguagem base vulgar em computadores sequenciais (p. ex. FORTRAN ou C). O programador tem de considerar pormenorizadamente a arquitectura da máquina para a qual se destina a aplicação, recebendo muito pouco auxílio do compilador.

Medidas de desempenho

Várias medidas têm sido propostas para avaliar o desempenho de algoritmos ou de aplicações. O estudo do desempenho de algoritmos a nível teórico, admite a execução de algoritmos em máquinas ideais, por exemplo com um número infinito de processadores. Enquanto nos algoritmos destinados a computadores sequenciais o estudo foca a complexidade temporal e de espaço de memória, nos algoritmos paralelos acresce o estudo da complexidade do número de processadores. A teoria de processamento paralelo tem obtido resultados para o estudo da complexidade destes algoritmos, destacando-se os limites de desempenho e a relação entre complexidade do tempo de computação e complexidade do número de processadores (ver p. ex. [Chaudhuri, 1982]). Neste texto, as medidas de desempenho referir-se-ão às aplicações, e não aos algoritmos. Estas medidas estão dependentes da arquitectura utilizada e chamam-se por vezes medidas estimadas.

Uma medida do desempenho de uma aplicação é o tempo de execução $T(n,p)$, que é função do tamanho n do problema e do número p de processadores utilizado. Esta medida não indica, porém, o mérito relativo da paralelização conseguida. Uma medida associada é o custo computacional $C(n,p)$, igual ao produto de $T(n,p)$ e p , e que não deve ser confundido com o custo real, que intervém noutra medida de comparação utilizada (que indica o custo/benefício), igual ao tempo $T(n,p)$ dividido pelo custo da máquina com p processadores.

Uma medida que indica o grau de paralelização obtido é o rácio de tempos de execução (*speedup*) real, $S(n,p)$, ou nominal, $S(n,p)^*$. O primeiro relaciona o tempo de execução de uma aplicação em paralelo com o tempo de execução do melhor algoritmo sequencial que executa a mesma aplicação, enquanto o segundo usa como medida de comparação o tempo de execução da aplicação paralela usando apenas um processador e mantendo toda a carga adicional exigida pelo paralelismo:

$$S(n,p) = \text{Menor tempo de execução com um programa sequencial} / T(n,p)$$

$$S(n,p)^* = T(n,1) / T(n,p)$$

Uma medida relacionada com o speedup é a eficiência real $E(n,p) = S(n,p) / p$, ou nominal $E(n,p)^* = S(n,p)^* / p$, que indica a utilização da capacidade computacional disponível nos p processadores.

Termina-se este capítulo com a apresentação da lei de Amdahl, sob a forma original e sob uma forma revista, de modo a extrair uma interessante conclusão. Considere-se a partição do tempo de execução de um programa num processador nas suas partes sequenciais e paralelizáveis: $T(n,1) = T_s(n,1) + T_p(n,1)$. Ao utilizar p processadores obtém-se, considerando fixo o tamanho n do problema,

$$S(n,p)^* = [T_s(n,1) + T_p(n,1)]/[T_s(n,p) + T_p(n,p)].$$

Considere-se que $f_s = T_s(n,1)$ e $f_p = T_p(n,1)$ e, por conveniência, $T(n,1) = 1$, isto é, a unidade de tempo é a execução da aplicação com tamanho n utilizando um só processador. f_s e f_p representam respectivamente a fracção sequencial e paralelizável do programa quando se utiliza um só processador. Considere-se também, optimisticamente, que o tempo de execução da fracção f_p utilizando p processadores é f_p/p . Na sua forma original [Amdahl, 1967] a lei de Amdahl é, para uma dimensão fixa n ,

$$S^*(p) = 1 / [f_s + f_p/p].$$

Esta lei foi considerada durante muito tempo como um limite aos benefícios da utilização de múltiplos processadores. Só mais tarde foi apresentada sob uma forma revista [Gustafson, 1988], que considera o seguinte:

- o tamanho do problema não é fixo;
- quando o número de processadores aumenta, o tamanho do problema também aumenta, mantendo-se fixa a quantidade de computação não paralelizável;
- $T(n',p) = 1$, isto é, a unidade de tempo é a execução da aplicação com tamanho n' utilizando p processadores, e pode ser decomposta numa fracção sequencial $f'_s = T_s(n',p)$ e numa fracção paralelizada $f'_p = T_p(n',p)$ (de que resulta $f'_s + f'_p = 1$);
- o tempo de execução da parte não paralelizável é igual quer se tenha um ou p processadores, ou seja, $T_s(n',1) = T_s(n',p) = f'_s$;
- o tempo de execução da parte paralelizável utilizando um processador é p vezes superior ao tempo utilizando p processadores, ou seja, $T_p(n',1) = p T_p(n',p) = p f'_p$.

Resulta então um speedup "escalado"

$$\begin{aligned} S^*(n',p) &= [T_s(n',1) + T_p(n',1)]/[T_s(n',p) + T_p(n',p)] \\ &= (f'_s + p f'_p) / (f'_s + f'_p) \\ &= f'_s + p f'_p \end{aligned}$$

Esta forma revista da lei de Amdahl não impõe qualquer limite superior ao speedup. Pressupõe que quando existem p processadores em vez de um único, o tamanho do problema aumenta de escala para aproveitar a maior capacidade computacional oferecida. Por outras palavras, recomenda que uma aplicação executada num computador sequencial poderá beneficiar de uma versão paralela para aumentar a dimensão do

problema, em vez de resolver mais rapidamente problemas da dimensão original. Esta conclusão de natureza qualitativa, apesar de se basear em pressupostos que constituem aproximações à realidade, sugere a utilização de processamento paralelo como um meio para redefinir as dimensões aceitáveis dos problemas computacionais.

Destaca-se por fim a existência de outra medida de desempenho, proposta recentemente, inspirada na interpretação de Gustafson: a fracção sequencial estimada [Karp e Flatt, 1990], que calcula a fracção não paralelizável de um programa a partir do speedup obtido por cronometragem.

3.4 Multiprocessadores de memória partilhada

Atribui-se o nome de multiprocessadores de memória partilhada (MMP) aos computadores MIMD cujos processadores comunicam através de memória partilhada. Também se utiliza a designação de MIMD fortemente ligados ou acoplados. O breve destaque que se dá a este tipo de arquitectura advém da sua popularidade actual, sendo vasta a lista de empresas de renome que tem produzido este tipo de máquinas, e que inclui a Cray, a IBM, a Alliant, a Sequent, a Convex e a Encore. De facto, esta arquitectura proporciona uma programação mais simples que a dos restantes MIMD, idêntica ao multiprocessamento em pseudo-paralelismo, na qual muitos programadores já são experientes. Quando as unidades elementares de execução são processos independentes, podem correr-se nos MMP aplicações escritas para máquinas sequenciais, sem qualquer alteração: os processos são distribuídos pelos diversos processadores em vez de partilharem um só. Neste âmbito, já existem versões para MMP do popular sistema operativo UNIX SVR4 (ver p. ex. [Nudelman, 1991]).

O problema que afecta este tipo de arquitecturas é o da contenção no acesso à memória, que se torna mais grave à medida que o número de processadores aumenta. Por este motivo, os MMP costumam ter um número modesto de processadores, na ordem da dezena. A redução da contenção no acesso à memória partilhada consegue-se à custa de memória cache. A memória cache (ver p. ex. [Vaughan-Nichols, 1991]) é um tipo de memória que pretende diminuir os acessos à memória normal, sendo mais rápida e de tamanho muito menor que esta última. Tenta-se por esse motivo guardar em memória cache os dados mais frequentemente acedidos. No entanto, a introdução de vários bancos de memória cache origina um problema de manutenção da coerência dos seus conteúdos, geralmente resolvido por protocolos de escuta em bus (*bus snooping*) ou baseados em directorias (ver p. ex. [Thakkar et al., 1990] e ainda [Vaughan-Nichols, 1991]). As diversas arquitecturas existentes apresentam compromissos entre o custo da máquina, o desempenho perdido por contenção no acesso à memória e o perdido pela manutenção da coerência dos diversos bancos de memória cache, e entre estes e a memória principal (ver p. ex. [Stenström, 1988], [Robinson, 1991]).

Podem dividir-se os MMP em três grandes conjuntos: memória partilhada centralizada, memória partilhada distribuída e só com memória cache (Figura 3.2a, 3.2b e 3.2c respectivamente). Estes últimos são ainda experimentais (ver por exemplo a Data Diffusion Machine em [Hagerstern et al., 1992]), não estando tão disseminados quanto os dois primeiros tipos. São um caso particular em que existe uma memória cache de segundo nível distribuída pelos diversos processadores, que constitui uma memória partilhada virtual.

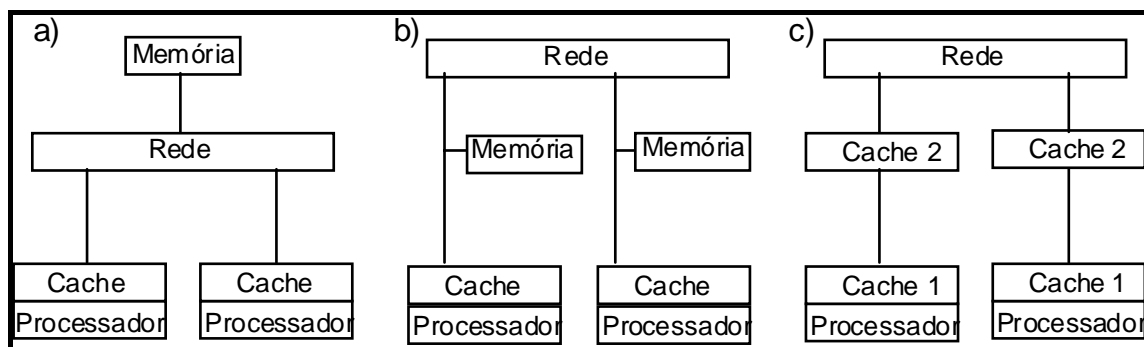


Figura 3.2 - Tipos de MMP

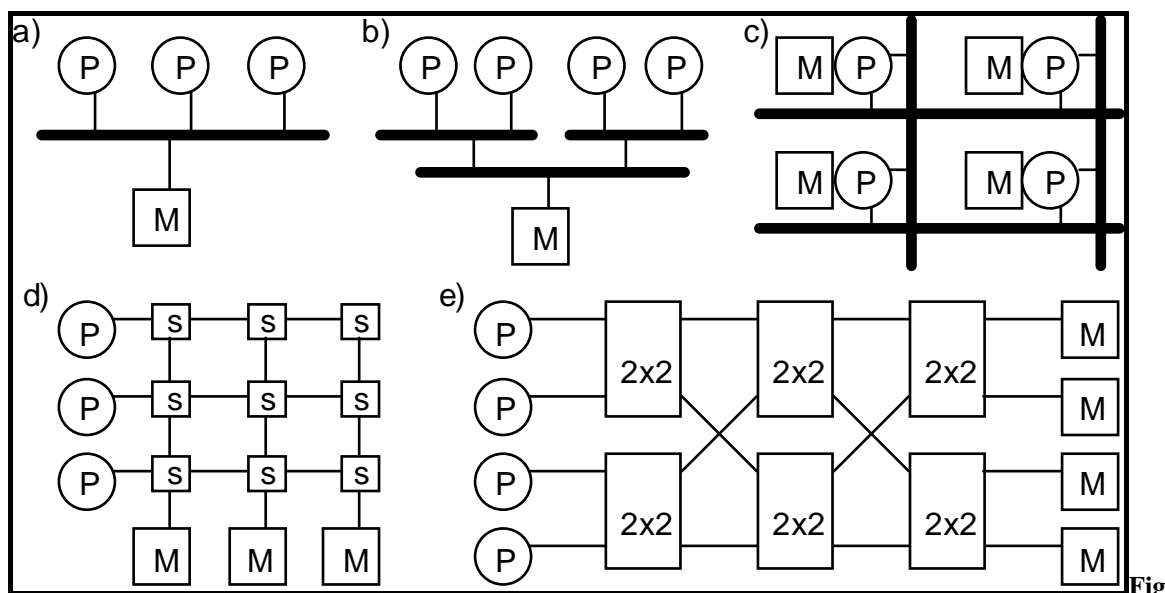


Figura 3.3 Tipos de interligação num MMP (P-processador, M-banco de memória)

Os multiprocessadores com memória partilhada central (global) ou com memória partilhada distribuída (local) são os mais comuns. Frequentemente, o programa "vê" uma memória partilhada global, mesmo que esta esteja distribuída. O acesso à memória partilhada pode efectuar-se de diversos modos, desde a utilização de buses às mais complexas redes de ligações. A utilização de um bus para acesso aos dados (Figura 3.3a) é a solução mais simples, e tem a vantagem de permitir técnicas de manutenção da coerência das caches por escuta do bus. No entanto, a contenção no acesso ao bus limita muito o número de processadores que podem coexistir nesta arquitectura. Uma variante

desta estratégia consiste em ter vários buses, dispostos numa hierarquia (Figura 3.3b) ou numa rede (Figura 3.3c). Num extremo oposto estão as redes de ligação completas $N \times N$ (*crossbar switches*). Estas redes (Figura 3.3d) podem ser vistas como uma central de comutação que pode efectuar simultaneamente N ligações correspondendo a uma aplicação bijectiva das N entradas (p. ex. processadores) nas N saídas (p. ex. bancos de memória). Permitem uma largura de banda muito grande, mas requerem um grande número de comutadores ($N \times N$) e são de construção onerosa. Numa posição intermédia estão as redes de ligação com múltiplos níveis. A Figura 3.3e mostra um exemplo de uma rede deste tipo, uma rede de Benes, que permite ligar quatro entradas a quatro saídas utilizando redes de ligação completas 2×2 . Outros exemplos podem ver-se, por exemplo, em [Lakshmivarahan e Dhall, 1990], sec. 2.3). Veja-se ainda [Bhuyan et al., 1989] para um estudo do desempenho das redes de ligações em MMP.

3.5 Multiprocessadores de passagem de mensagens (multicomputadores)

Os multicomputadores, ou computadores MIMD fracamente ligados (ou acoplados), caracterizam-se por não possuírem memória partilhada. Os diversos processadores comunicam através do envio explícito de mensagens através de ligações físicas. O nome multicomputador deriva do facto de cada elemento de processamento (nó) possuir memória local e poder, até certo ponto, ser considerado um computador. Cada processador só pode endereçar a sua memória local, que é privada.

A grande vantagem, ou até a própria razão de existência (segundo [Pountain e Bryan, 1992]), dos multicomputadores é a escalabilidade de hardware, isto é, a possibilidade de aumentar linearmente com o número de processadores o poder computacional oferecido. O principal inconveniente é, actualmente, a dificuldade de programar nestas arquitecturas e a consequente escassez de software comercial. A realização de programas para este tipo de MIMD envolve, para além da correcção lógica, a correcção em termos de interacção entre entidades de processamento (p. ex. cada mensagem enviada tem de ser recebida por outra entidade), a partição dos dados pelos diversos processadores e uma distribuição equilibrada da carga computacional (ver p. ex. [Stein, 1991])

Nos multicomputadores o desempenho é influenciado pelos processadores utilizados (geralmente microprocessadores produzidos em larga escala), da velocidade das ligações físicas inter-processador e, sobretudo, da topologia do multicomputador. A topologia é definida pelo modo como os diversos nós estão ligados entre si. Topologias comuns são a rede vectorial, o anel, a rede matricial (*grid* ou *mesh*), a árvore, o toro, ou o hipercubo (Figura 3.4).

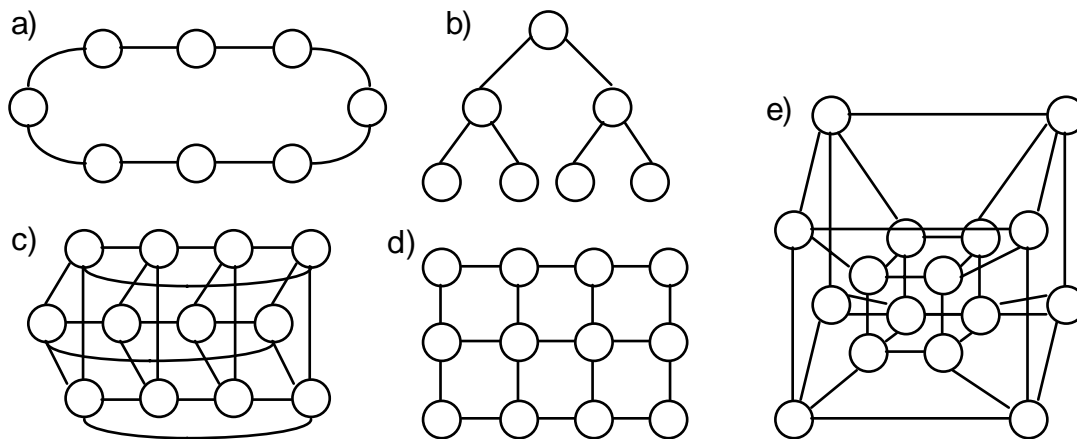


Figura 3.4 - Exemplos de topologias: a) anel, b) árvore, c) toro, d) rede matricial, e) hiper-cubo

Relacionado com a topologia surge o problema do encaminhamento das mensagens entre os diversos nós (ver p. ex. [Athas e Seitz, 1988]). Pretende-se minimizar o tempo de latência das mensagens, ou seja, o tempo que demoram a chegar ao destino. Este tempo inclui não só a transmissão da mensagem, mas também o tempo de estabelecimento do canal ou o tempo devido ao encaminhamento da mensagem. Frequentemente, existe um limite ao tamanho máximo de uma mensagem, pelo que as mensagens demasiado longas têm de ser partidas em mensagens mais pequenas. Dado que muitas vezes a mensagem passa por nós intermédios até chegar ao nó destino, é necessário existir um algoritmo de encaminhamento das mensagens do nó origem para o nó destino. O encaminhamento da mensagem poderá ser realizado por software ou, nas arquitecturas mais avançadas, por hardware. Segundo [Pountain, 1990], um algoritmo de encaminhamento deveria ser:

- completo, ou seja, garante da chegada de todas as mensagens;
- à prova de impasses (*deadlock free*);
- óptimo, no sentido em que cada mensagem segue o caminho mais curto;
- económico, na medida em que exija pouca informação de controlo em cada mensagem;
- escalável, ou seja, bom para qualquer dimensão da rede de nós;
- versátil, ou seja bom para qualquer topologia da rede de nós; e
- rápido a executar.

A esta lista poder-se-ia acrescentar a garantia de tempos de latência máximos razoáveis. De facto, a ocorrência de congestionamento em algumas ligações muito utilizadas, pode conduzir a tempos de latência muito superiores ao tempo de latência médio. Em aplicações de tempo-real (ver sobre este assunto [Stankovic, 1988] e [Stankovic, 1992]), para as quais o processamento paralelo é muito promissor, os tempos de latência têm limites superiores estritos e a possibilidade de congestionamento não pode ser negligenciada.

Pountain refere que os algoritmos utilizados na prática ainda não possuíam todas estas características, referindo alguns dos mais usados. Por exemplo, a técnica de colocar tabelas de encaminhamento em cada nó contendo informação sobre os caminhos para os restantes nós não é rápida a executar nem escalável. A técnica de erosão de bits, segundo a qual cada nó intermédio retira um bit de controlo e utiliza-o para decidir o caminho, não é versátil nem escalável. Uma outra técnica é o encaminhamento por intervalos, pela qual a cada ligação emanante de determinado nó está associado um intervalo $[i,j]$ de números inteiros, que indica ser essa a ligação que conduz aos nós i a j . O sucesso desta técnica depende, porém, do algoritmo de atribuição de intervalos às ligações, para que seja óptima ou versátil. Uma técnica curiosa é a do encaminhamento universal, que pretende minorar a congestão em ligações (ou nós) muito utilizadas(os), embora à custa de maiores tempos de latência. A mensagem é primeiro enviada para um nó escolhido ao acaso, e só depois é encaminhada desse nó para o nó destino.

A escolha de uma topologia adequada à aplicação que se pretende programar é fundamental. Alguns multicomputadores permitem que por vezes se altere a sua topologia física para conseguir uma melhor adaptação ao problema. Porém, o mais comum, por enquanto, é que essa alteração seja limitada ou impossível, pelo que se recorre a software (do sistema ou do programador da aplicação) para obter uma determinada topologia virtual na topologia física existente. A Figura 3.5 apresenta um exemplo de obtenção de um anel virtual quando a topologia física é uma rede matricial 3×4 .

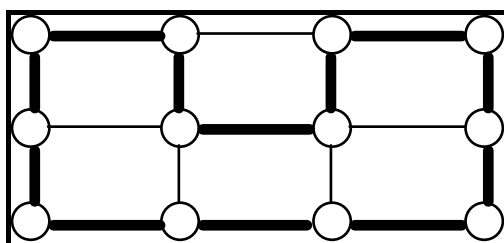


Figura 3.5 - Exemplo de topologia anel virtual (anel)

Uma das topologias mais estudadas e populares é o hipercubo (ver p. ex. [Lakshminarayanan e Dhall, 1990], sec. 2.2), dado que minimiza a distância máxima entre dois nós (diâmetro). É por isso frequentemente utilizada para aplicações em que o padrão de comunicação inclua muitas ligações de "longa distância". Também é popular porque permite encaminhamentos simples, e porque existem muitos resultados sobre o modo de "embeber" outras topologias no hipercubo. Padecem, porém, da desvantagem de serem em muitos casos difíceis de realizar num plano.

Um hipercubo de base b com N nós ($N \geq b \geq 2$) pode ser definido como um grafo $G=(V,A)$, em que V é o conjunto de nós (vértices) e A o conjunto de arcos (arestas). Seja k um número natural tal que $b^{k-1} < N \leq b^k$, então,
 $V = \{x : x \text{ é um inteiro de base } b \text{ com } k \text{ dígitos, i.e. } x = x_k x_{k-1} \dots x_1\}$ e

$A = \{(x,y) : x,y \in V \wedge \exists j \in \{1,\dots,k\} : x_j+y_j \equiv x_i=y_i \text{ para todo } i \neq j\}$, ou seja, dois nós estão ligados por um arco se e só se os seus identificadores de k dígitos diferirem em exactamente um dígito. No caso de $N=b^k$ o hipercubo diz-se completo. A distância máxima entre dois quaisquer nós é $\log_b N$, ou seja cresce logaritmicamente com o número de nós. A Figura 3.6 apresenta mais alguns exemplos de hipercubos, utilizando a notação (N,b,k) . As Figuras 3.6a) b) e c) exemplificam hipercubos completos.

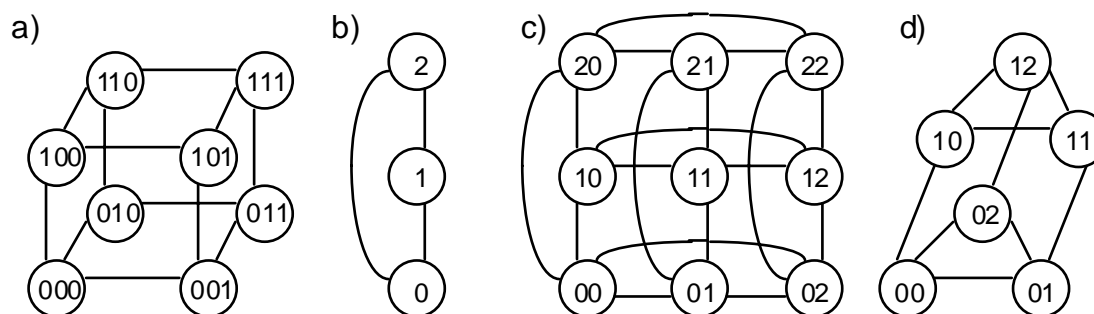


Figura 3.6 - Exemplos de hipercubos: a) (8,2,3), b) (3,3,1), c) (9,3,2) e d) (6,3,2)

Os hipercubos de base dois são os mais populares, provavelmente devido à lógica binária subjacente ao funcionamento dos computadores. Os primeiros multicomputadores, como o experimental Cosmic Cube [Seitz, 1985] ou os comerciais Intel iPSC e NCUBE, utilizam hipercubos de base dois.

Outra topologia que importa definir é a rede matricial (a duas dimensões), cujo interesse resulta de ser a topologia física da máquina utilizada neste trabalho. Ao contrário do hipercubo, esta topologia favorece a comunicação de "curta distância". Tem sobre o hipercubo a vantagem de ser mais facilmente realizável no plano. A rede matricial pode ainda ser partida em unidades que também são redes matriciais e suporta algoritmos de encaminhamento simples [Athas e Seitz, 1988]. Formalmente, pode definir-se uma rede matricial a duas dimensões $X \times Y$ como um grafo $G=(V,A)$, em que

$$V = \{(x,y) : x,y \in \mathbb{Z}_0 \wedge x < X \wedge y < Y\},$$

$$A = \{((a,b) , (c,d)) : (a,b) \in V \wedge (c,d) \in V \wedge [(a=c \wedge |b-d|=1) \vee (b=d \wedge |a-c|=1)]\}.$$

Existem $X \times Y$ nós. A distância máxima entre dois nós quaisquer é $(X-1)+(Y-1)$. No caso de $X=Y$ a distância máxima é $2(X-1)$, ou seja cresce com a raiz quadrada do número de nós.

3.6 Ambiente utilizado

O computador utilizado é um multicomputador Parsytech MultiCluster(MC)-3/DE. Os elementos processadores deste multicomputador são transputers Inmos T805. Esta secção descreve este ambiente, sob a perspectiva de hardware e software, remetendo frequentes vezes para conceitos apresentados nas secções 3.3 e 3.5. Começa-se por

descrever o tipo de processador utilizado, seguindo-se a descrição do multicomputador e das ferramentas de desenvolvimento.

Transputers

Os transputers constituem processadores microprogramados muito simples, especialmente vocacionados para processamento paralelo [Inmos, 1989], [Pountain, 1990]. O seu nome deriva das palavras *transistor* e *computer*: é um computador num só *microchip* (pastilha de silício), consistindo num bloco de construção. Possuem num só microchip a unidade de processamento central (CPU), alguma capacidade de memória interna, um interface para memória externa e quatro interfaces de ligação bidireccionais. A partir do transputer T800 também se encontra no microchip uma unidade de cálculo em vírgula flutuante (FPU) integrada (ver Figura 3.7). A CPU, a FPU e os interfaces de ligação podem funcionar em paralelo.

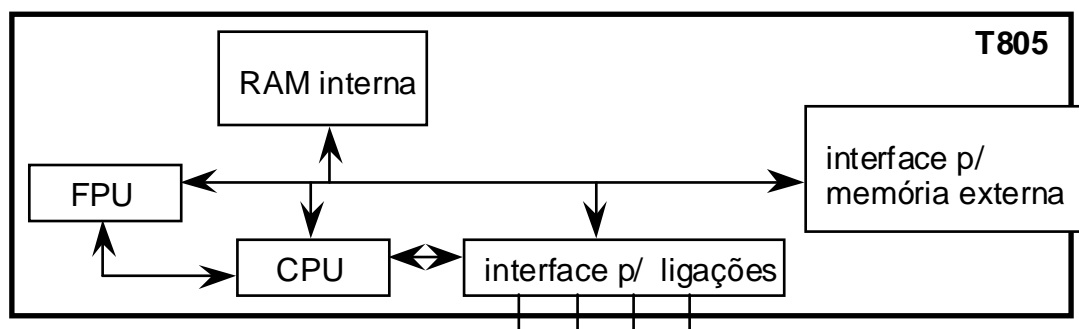


Figura 3.7 - Esquema de um Inmos Transputer T805

As ligações entre transputers são muito rápidas, apesar de serem ligações série. Para essa velocidade contribui o facto do protocolo de comunicação ser realizado pelo hardware. A comunicação entre transputers é síncrona, embora por software se possa emular um modo de comunicação assíncrono. Os interfaces de ligação permitem que cada transputer possa ser ligado a um máximo de quatro outros transputers, construindo-se facilmente redes de transputers em diversas topologias. O seu número é, contudo, uma limitação, porque quando há necessidade de mais de quatro canais lógicos (ou virtuais), as ligações físicas têm de ser partilhadas. De acordo com o anunciado, o transputer T9000 efectuará esta partilha por hardware, mas nos transputers anteriores só por software, com maior lentidão, se pode efectuar esta "multiplexagem".

O transputer oferece ainda possibilidade de executar vários processos em pseudo-paralelismo. O escalonamento dos processos é executado por hardware com tempos de mudança de contexto extremamente curtos (2 μ s), pelo que o pseudo-paralelismo acarreta custos muito baixos. Os processos possuem um de dois níveis de prioridade (prioritários e não-prioritários), não havendo preempção quando estão a ser executados processos prioritários.

O modelo de transputer utilizado é a versão T805 do T800, que opera com uma frequência de relógio de 30 MHz. É um processador de 32 bits que possui uma FPU de 64 bits e pode aceder à memória em palavras de 32 bits. A memória integrada no transputer é de 4 Kbytes, podendo este endereçar memória externa até 4 Gbytes. No que respeita à velocidade, este modelo de transputer pode executar até 30 MIPS e 4,3 MFLOPS, comunicando à velocidade de 20 Mbits/s por ligação. As principais limitações deste modelo em relação ao seu sucessor anunciado (T9000) serão a impossibilidade de partilhar ligações físicas por hardware e a ausência de encaminhamento por hardware.

O multicomputador Parsytech MC-3/DE

O Parsytech MC-3/DE é um computador paralelo MIMD, cujos elementos processadores são transputers T805 (no máximo 24), que comunicam por passagem de mensagens. A topologia deste multicomputador é uma rede matricial bi-dimensional com 16 T805 na configuração utilizada (Figura 3.8). As ligações entre os nós são bidireccionais e a comunicação processa-se de modo síncrono. Os processadores estão fisicamente dispostos por oito placas de dois transputers cada, montadas numa estrutura que assegura a sua alimentação e arrefecimento. Existem dois LED em cada placa que assinalam a actividade dos seus dois transputers.

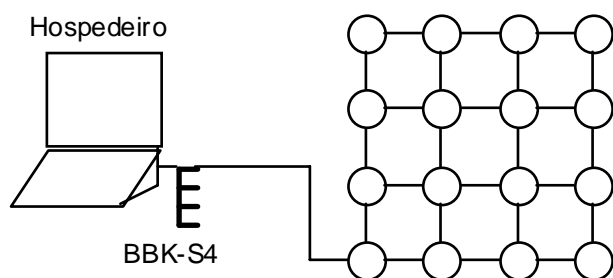


Figura 3.8 - Configuração do ambiente de desenvolvimento

A interacção entre o programador ou o utilizador com o computador paralelo efectua-se por intermédio de uma estação de trabalho (hospedeiro), na instância uma SUN a correr o sistema operativo UNIX. É através da SUN que o multicomputador pode aceder a um disco rígido, a um écran ou a outro dispositivo de entrada/saída. A SUN está ligada à estrutura que contém os transputers por intermédio de uma placa BBK-S4 que oferece quatro ligações físicas com velocidade idêntica à das ligações entre transputers (20 Mbits/s). Cada uma das ligações da BBK-S4 termina num dos interfaces de ligação de transputer disponíveis. Deste modo, permite-se a partição da rede de transputers em quatro redes que possam funcionar independentemente, utilizando-se cada ligação da BBK-S4 para aceder a uma rede ("máquina virtual") distinta (ver exemplo na Figura 3.9). A flexibilidade do MC-3/DE permite ainda, previsivelmente, utilizar placas com a

anunciada geração de transputers seguinte (T9000), eventualmente em conjunto com as placas existentes.

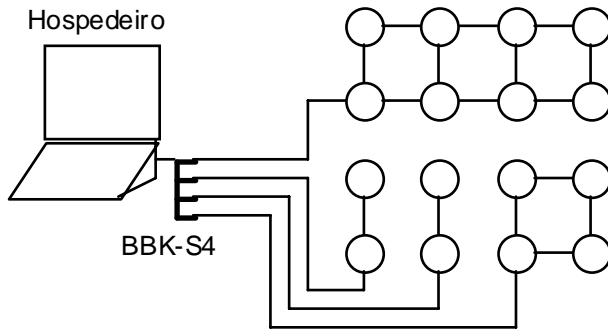


Figura 3.9 - Partições no MC-3/DE

O software

A gestão dos recursos do computador paralelo é efectuada pelo software PARIX, versão 1.2 [Parsytech, 1993], uma extensão paralela ao UNIX. O PARIX suporta a construção de aplicações paralelas através das suas bibliotecas de funções. Não proporciona, assim, mais do que um paradigma *ad-hoc* de baixo nível de abstracção.

A execução do software PARIX distribui-se pelo hospedeiro e pela rede de transputers, como indica o quadro seguinte:

| | Utilização | Administração |
|-------------------|-----------------------------|--|
| Hospedeiro | ambiente de desenvolvimento | gestão do ambiente multi-utilizador |
| MC-3/DE | ambiente de execução | gestão de partições e iniciações |

De entre as capacidades oferecidas no ambiente PARIX destacam-se as seguintes:

1. carregamento do mesmo programa em todos os transputers;
2. identificação de cada transputer na topologia virtual e na topologia física;
3. estabelecimento de topologias virtuais distintas da topologia física;
4. criação de fios de processamento;
5. criação de ligações virtuais entre quaisquer dois processos;
6. simulação de comunicação assíncrona (para além do natural usufruto da comunicação síncrona);
7. encaminhamento de mensagens (no caso de não se utilizar o T9000);
8. acesso a dois relógios, um com precisão de 64 μ s, outro com precisão 1 μ s;
9. utilização de chamadas a procedimentos remotos (RPCs) utilizando o hospedeiro.

O modelo de programação sugerido pelo PARIX consiste na coexistência dos vários processos da aplicação, um em cada processador, que podem ser executados em paralelo. Cada um destes processos poderá conter fios de processamento a partilhar o

processador em pseudo-parallelismo. Os diversos fios de processamento podem comunicar entre si, sincronizando-se através do envio de mensagens em modo síncrono (se estiverem em processadores diferentes) ou através de semáforos (se pertencerem ao mesmo processo).

Pelo ponto 1, é necessário escrever um programa que será executado em todos os transputers. Esse programa deverá conter fluxos de instruções distintos consoante a posição que cada transputer ocupa na topologia escolhida. Note-se que o ponto 2 indica que tal é possível: uma entidade de processamento pode conhecer a posição na rede do nó em que está a ser executada, bastando para isso efectuar uma chamada ao sistema. Pode por isso construir-se código que inclua saltos condicionais dependendo da localização do nó.

Os pontos 3 e 4 permitem o uso de topologias e ligações virtuais, bem como o uso de pseudo-parallelismo em cada transputer através da criação de fios de processamento. O uso de ligações virtuais acarreta, para a geração T800 de transputers, a partilha das ligações físicas por software, de um modo transparente ao utilizador. As topologias virtuais podem utilizar-se, quer após serem definidas à custa de ligações virtuais, quer escolhendo uma das topologias cuja construção está pré-definida em bibliotecas. Estas bibliotecas asseguram a optimização da distância máxima das ligações físicas correspondentes a uma ligação virtual. A principal vantagem do uso de topologias virtuais é a abstracção da topologia física por estas proporcionada, que resulta numa maior legibilidade e portabilidade do código.

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 (envio de mensagem em modo síncrono ou recepção de mensagem) que cause um bloqueio (enquanto a outra parte não estiver pronta a comunicar), então só ficará bloqueado um fio de processamento e não a totalidade de um processo. Os restantes fios poderão permanecer activos e ser executados. Note-se que, no caso de não se recorrer a fios de processamento, a possibilidade de receber comunicação por mais que um canal poderia ser muito dificultada. Os fios de processamento podem ser encarados como subrotinas que operam concorrentemente.

A possibilidade de estabelecer um canal virtual entre dois processos localizados em processadores não vizinhos (ponto 5), libertando o programador da tarefa de encaminhar as mensagens (ponto 7), bem como a possibilidade de simular comunicação assíncrona (ponto 6), permite estilos de programação muito variados e, mais importante, facilita a reutilização de código. Todavia, no trabalho desenvolvido não se sentiu

necessidade de explorar estas capacidades, preferindo-se controlar mais directamente a comunicação (especialmente o encaminhamento) entre os processos.

A capacidade referida no ponto 8 é extremamente útil para a obtenção das medidas de desempenho referidas na secção 3.3. Neste trabalho utilizou-se apenas o relógio menos preciso, cuja precisão já é muito boa para os fins pretendidos e tem a vantagem de voltar ao zero muito mais lentamente (só após vários dias). A possibilidade de efectuar RPCs (chamadas a procedimentos remotos) oferece a qualquer processo a ser executado num dos transputers os serviços tipicamente utilizados no ambiente UNIX, como o acesso a discos rígidos ou a máquinas remotas.

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 [ACE, 1992a], [ACE, 1992b]. Inevitavelmente, o programa inicial conterà erros cuja detecção é muito morosa, através de ensaios de tentativa e erro. O ciclo é reiniciado sempre que se encontre um erro, até à obtenção de um programa correcto.

4 - PARALELIZAÇÃO DO MÉTODO PROMETHEE

4.1 Introdução

O método PROMETHEE¹⁶ [Brans, 1982; Brans e Vincke, 1985; Brans et al., 1986] pertence à família de métodos conducentes a uma relação de prevalência de síntese (ver sec. 2.5). Na execução do método PROMETHEE os computadores sequenciais conseguem satisfazer plenamente as necessidades do utilizador, quando o problema computacional de apoio à decisão tem um "tamanho razoável". A resolução do problema computacional num computador paralelo permitirá potencialmente aumentar o tamanho do problema, mantendo tempos de resposta aceitáveis para o utilizador.

Em geral, o problema computacional pode tornar-se grande quer no número de critérios quer no número de alternativas consideradas. No caso do método PROMETHEE, pode encontrar-se referência a uma ferramenta, de nome BANKADVISER [Mareshal e Brans, 1991], que foi utilizada numa situação de decisão com 557 alternativas. Dado que essa aplicação está baseada num PC (computador pessoal sequencial), não foi considerado praticável comparar todas as alternativas entre si (Mareshal e Brans sugerem um máximo de 50 alternativas para que tal fosse possível), tal como preconiza o método PROMETHEE. Foi adoptada nesse caso a estratégia de usar um banco contendo 25 alternativas "de referência" com as quais as restantes foram comparadas. Este exemplo ilustra como a introdução do processamento paralelo pode alargar os horizontes daquilo que é possível efectuar: se o programa fosse suficientemente rápido, todas as alternativas poderiam ter sido comparadas entre si, tal como preconiza o método PROMETHEE.

No que se refere à existência de um grande número de critérios, pode-se, por exemplo, imaginar um sistema de apoio à decisão em grupo em que o conjunto dos critérios seja a reunião de todos os critérios de cada decisor. Em [Keeney, 1992] apresentam-se outras situações que podem originar um número considerável de critérios: transporte de resíduos nucleares [Keeney, 1992, sec. 11.2] (31 objectivos), programa de pesquisa sobre mudanças climáticas [Keeney, 1992, sec. 11.3] (79 objectivos) e gestão da empresa British Columbia Hydro [Keeney, 1992, Cap. 12] (22 atributos).

Este capítulo descreve uma experiência de implementação do método PROMETHEE, nas suas variantes PROMETHEE I e PROMETHEE II, usando a arquitectura paralela descrita na secção 3.6. Apresenta-se o método, os programas desenvolvidos, as experiências efectuadas e os resultados obtidos. Pretendeu-se apenas

¹⁶ PReference Ranking Organization METHOD for Enrichment Evaluations.

estudar o comportamento dos vários programas desenvolvidos, sem ter em vista uma utilização concreta para os mesmos.

4.2 O método PROMETHEE

Considere-se que, concluída uma fase de estruturação da situação de decisão, se definiu o conjunto A de m alternativas a avaliar e um conjunto C de n critérios $f_1(\cdot)$, ..., $f_n(\cdot)$. Os critérios são funções que atribuem um valor real a cada alternativa. Represente-se por $f_j(a)$ o desempenho da alternativa a segundo o critério j. Estes desempenhos são os elementos de uma matriz de decisão, que se considera definida. Como exemplo, considera-se a seguinte matriz de decisão (m=4 e n=3):

| | crit1 | crit2 | crit3 |
|------|-------|-------|-------|
| alt1 | 2 | 79 | 7 |
| alt2 | 3 | 99 | 1 |
| alt3 | 2 | 96 | 4 |
| alt4 | 3 | 90 | 6 |

O método PROMETHEE desenvolve-se em duas fases: construção da relação de prevalência e exploração da relação de prevalência. Segundo [Brans e Vincke, 1985] os métodos de prevalência caracterizam-se por enriquecer a relação de dominância, que se baseia num princípio de unanimidade, aceitando uma relação mais fraca, baseada num princípio de maioria¹⁷. Troca-se deste modo uma maior segurança nas avaliações por um aumento do número de pares de alternativas comparáveis.

Na primeira fase, o método PROMETHEE compara todos os pares de alternativas, agregando os desempenhos segundo os vários critérios numa relação de prevalência difusa, a que também se chamou na secção 2.5 relação de prevalência do tipo 2. Numa segunda fase, explora essa relação com vista a obter uma ordenação das alternativas com possíveis *ex-aequo*, i.e. uma pré-ordem total, através da variante PROMETHEE II, ou com vista a obter uma pré-ordem com possibilidade de incomparabilidade entre alternativas, i.e. uma pré-ordem parcial, através da variante PROMETHEE I.

A fase de construção da relação de prevalência, segundo Brans, requer que se associe a cada ponto de vista aquilo a que chama um critério generalizado $p_j(x)$. Recorde-se que normalmente se chama critério (ou, mais rigorosamente, verdadeiro-critério) a uma função $f_j(a)$, cujo domínio é A, tal que, dadas duas alternativas $a_1, a_2 \in A$ para as quais se calcula $x = f_j(a_1) - f_j(a_2)$, se pode concluir que:

¹⁷ Segundo [Roy, 1990], a relação binária obtida pelo PROMETHEE não é estritamente uma relação de prevalência. Neste texto utiliza-se a expressão "relação de prevalência" num sentido lato, tal como [Brans e Vincke, 1985], [Vincke, 1989] e [Pomerol e Barba-Romero, 1993] (para citar alguns exemplos).

- a_1 e a_2 são indiferentes, segundo o ponto de vista operacionalizado por f_j , se $x = 0$;
- a_1 é preferível a a_2 , segundo o ponto de vista operacionalizado por f_j , se $x > 0$;
- a_2 é preferível a a_1 , segundo o ponto de vista operacionalizado por f_j , se $x < 0$;

A noção de critério generalizado permite modelar, para cada critério, a atitude do decisor perante a diferença de desempenhos x , através de uma função $p_j(x)$, cujo domínio é o conjunto dos números reais. Esta generalização permite associar a uma diferença de desempenhos $x = f_j(a_1) - f_j(a_2)$ um índice que indica o grau de preferência da alternativa a_1 em relação a uma alternativa a_2 segundo o j -ésimo critério. Brans propõe seis tipos diferentes de critério generalizado (Figura 4.1), em que se verifica que $p_j(x)$ varia entre 0 e 1 e é nulo quando x é negativo. Passa-se a descrever esses seis tipos, omitindo o índice j em $p_j(x)$ e considerando duas alternativas genéricas $a_1, a_2 \in A$ para as quais se calculou $x = f_j(a_1) - f_j(a_2)$.

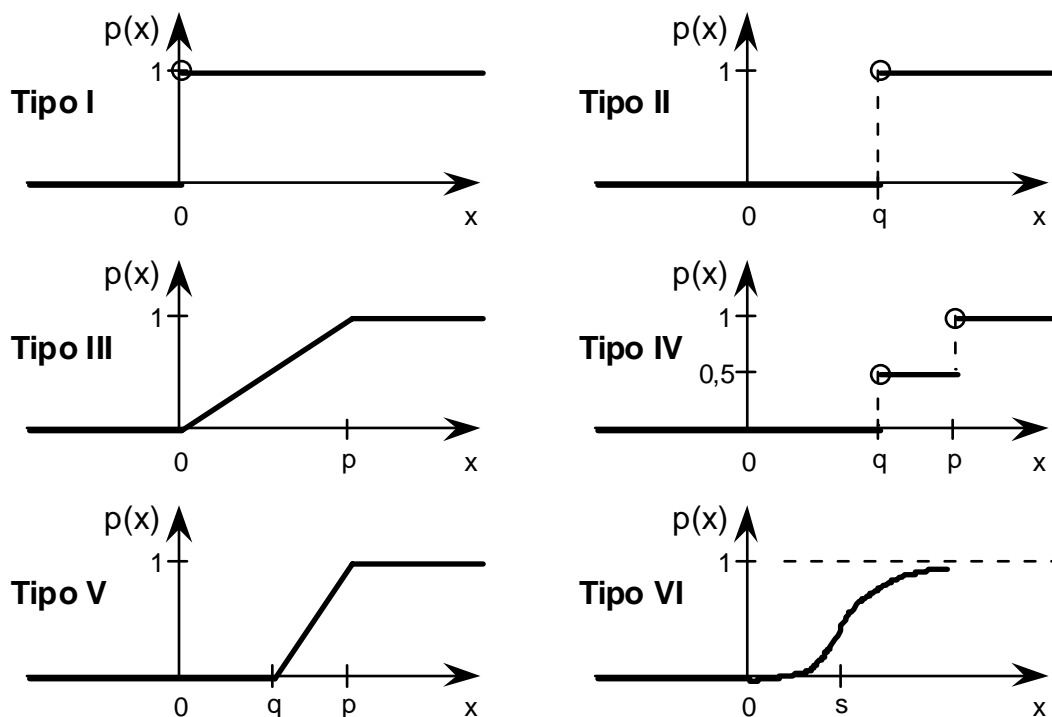


Figura 4.1 - Tipos de critério generalizado

Tipo I

Neste caso as alternativas só são consideradas indiferentes se $x=0$. A expressão analítica de $p(x)$ é:

$$p(x) = \begin{cases} 0 & , \text{se } x \leq 0 \\ 1 & , \text{se } x > 0 \end{cases}$$

Tipo II

Esta generalização do conceito de critério permite considerar indiferentes duas alternativas, desde que a diferença dos seus desempenhos seja igual ou inferior a um limiar de indiferença q , definido pelo decisor para o critério considerado.

A expressão analítica de $p(x)$ é:

$$p(x) = \begin{cases} 0 & , \text{se } x \leq q \\ 1 & , \text{se } x > q \end{cases}$$

Tipo III

Novamente, duas alternativas só são consideradas equivalentes se $x=0$. Caso contrário, podem verificar-se duas situações distintas. Se a diferença entre os desempenhos segundo o critério considerado for superior a um limiar p de preferência estrita, definido pelo utilizador, então a alternativa com melhor desempenho é estritamente preferível à outra, pelo que $p(x)$ é igual ao valor máximo. Se não, considera-se que o índice que indica o grau de preferência pela alternativa com melhor desempenho cresce linearmente com a diferença de desempenhos.

A expressão analítica de $p(x)$ é:

$$p(x) = \begin{cases} 0 & , \text{se } x \leq 0 \\ x/p & , \text{se } 0 < x < p \\ 1 & , \text{se } x \geq p \end{cases}$$

Tipo IV

De acordo com este critério, duas alternativas são indiferentes se x não ultrapassar um limiar de indiferença q (definido pelo decisor), e aquela com melhor desempenho é estritamente preferida se x for superior a um limiar de preferência estrita p (também definido pelo decisor). Este tipo pode ser comparável ao pseudo-critério tal como definido no Cap. 2. Porém, segundo [Brans e Vincke, 1985], considera-se que quando x é superior a q mas inferior (ou igual) a p , tal não corresponde a uma hesitação entre indiferença e preferência estrita, mas a um índice de preferência intermédio de 0,5.

A expressão analítica de $p(x)$ é:

$$p(x) = \begin{cases} 0 & , \text{se } x \leq q \\ 0,5 & , \text{se } q < x \leq p \\ 1 & , \text{se } x > p \end{cases}$$

Tipo V

Este tipo de critério é semelhante ao anterior, à excepção da situação em que a diferença de desempenhos x é superior ao limiar de indiferença q e inferior ao limiar de preferência estrita p . Nesse caso, em vez de se obter um índice de preferência constante, resulta um índice $p(x)$ que cresce nesse intervalo linearmente com a diferença de desempenhos.

A expressão analítica de $p(x)$ é:

$$p(x) = \begin{cases} 0 & , \text{se } x \leq q \\ \frac{x-q}{p-q} & , \text{se } q < x \leq p \\ 1 & , \text{se } x > p \end{cases}$$

Tipo VI

O último tipo de critério proposto é inspirado numa curva de Gauss e possui um só parâmetro s , a fixar pelo decisor, que coincide com o ponto de inflexão da curva. Segundo [Brans et al., 1986], este tipo de critério possui a vantagem de possuir uma expressão analítica diferenciável, o que pode contribuir para uma maior estabilidade dos resultados perante variações introduzidas nos dados de entrada.

A expressão analítica de $p(x)$ é:

$$p(x) = \begin{cases} 0 & , \text{se } x \leq 0 \\ 1 - e^{-x^2/(2s^2)} & , \text{se } x > 0 \end{cases}$$

De acordo com [Brans e Vincke, 1985], poder-se-ão criar novos tipos de critério generalizado, embora se considere que os seis tipos propostos são suficientes. Também se pode tornar os tipos existentes mais flexíveis através da introdução de limiares variáveis $q(\cdot)$, $p(\cdot)$ ou $s(\cdot)$.

A comparação de todos os pares de alternativas utilizando os critérios generalizados resulta num índice de preferência $p_j[f_j(a)-f_j(b)]$ por cada par ordenado de alternativas $(a,b) \in A \times A$ e por cada critério j . Obtém-se uma relação de prevalência difusa de síntese ao agregar estes índices de preferência, que são monocritério, num índice de preferência multicritério. O modelo de agregação utilizado consiste numa média ponderada dos índices monocritério para calcular o *índice de preferência multicritério*:

$$\pi(a,b) = \sum_{j=1}^n w_j p_j[f_j(a) - f_j(b)]$$

O decisor atribui importâncias diferentes a critérios diferentes através da especificação dos coeficientes de ponderação (pesos) w_j que intervêm nesta soma ponderada. Atente-se, no entanto, que o significado destes pesos é diferente do apresentado para os coeficientes de ponderação na secção 2.4. Os pesos deverão ser superiores a zero e a sua soma deverá ser um. Deste modo, os índices de preferência multicritério $\pi(a,b)$ variam entre 0 e 1 e definem uma relação de prevalência difusa. Note-se que, ao contrário do descrito na secção 2.5 (que se aplica ao método ELECTRE III, precursor dos métodos de prevalência difusos), este índice não pretende reflectir a credibilidade da afirmação "**a** prevalece sobre **b**". Procura, em vez disso, atribuir um grau à preferência por **a** em detrimento de **b** ou, segundo [Roy e Bouyssou, 1993, p.280], reflectir a credibilidade da afirmação "**a** é estritamente preferível a **b**". Note-se ainda que o índice de preferência multicritério se define para pares ordenados de alternativas: dadas

as alternativas **a** e **b** define-se $\pi(a,b)$ e $\pi(b,a)$. Para além disso, em geral não se pode inferir nada sobre $\pi(a,b)$ conhecendo-se apenas $\pi(b,a)$. A relação que se obtém pode representar-se através de um grafo orientado com valores nos arcos.

Procedendo à aplicação do método PROMETHEE ao exemplo cuja matriz de decisão foi apresentada acima, suponha-se que aos três pontos de vista utilizados estão associados os seguintes critérios generalizados:

| CRITÉRIOS | peso | tipo | par1 | par2 | |
|-----------|------|------|------|------|---|
| crit1 | 0.25 | 1 | | | |
| crit2 | 0.5 | 6 | 10 | | {par1 representa s} |
| crit3 | 0.25 | 4 | 2 | 4 | {par1 representa q e par2 representa p} |

Os índices de preferência multicritério que se obteriam, que definiriam uma relação de prevalência difusa, seriam os elementos da seguinte matriz:

| | alt1 | alt2 | alt3 | alt4 |
|------|-------|-------|-------|-------|
| alt1 | | 0.250 | 0.125 | 0.000 |
| alt2 | 0.682 | | 0.272 | 0.167 |
| alt3 | 0.382 | 0.125 | | 0.082 |
| alt4 | 0.477 | 0.250 | 0.250 | |

O grafo correspondente é apresentado na Figura 4.2.

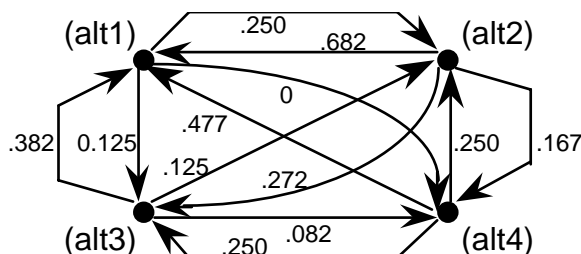


Figura 4.2 - Exemplo de relação definida pelos índices de preferência multicritério

Para a fase de exploração da relação obtida Brans propõe duas abordagens, a do PROMETHEE I e a do PROMETHEE II. Tanto uma quanto a outra exploram a relação de prevalência de um modo muito simples, baseado na "força" e "fraqueza" relativas de cada alternativa perante todas as outras. Pode estabelecer-se uma analogia entre a força de uma alternativa e a soma do número de pontos que um desportista marca perante todos os seus adversários e, de acordo com a mesma analogia, interpretar a fraqueza de uma alternativa como a soma dos pontos que os adversários marcam no confronto com o mesmo desportista. Formalmente, calcula-se a força de cada alternativa através de¹⁸

$$\Phi^+(a) = \sum_{i=1}^m \pi(a,i) \text{ (fluxo que sai do nó do grafo),}$$

¹⁸ Note-se que, qualquer que seja a alternativa **a**, $\pi(a,a) = 0$.

e calcula-se a fraqueza de cada alternativa através de

$$\Phi^{-}(a) = \sum_{i=1}^m \pi(i,a) \text{ (fluxo que entra no nó do grafo).}$$

O PROMETHEE I permite obter uma pré-ordem parcial (P, I, R) a partir de duas pré-ordens completas distintas, (P^{+}, I^{+}) e (P^{-}, I^{-}) . A primeira permite ordenar as alternativas segundo a sua força, enquanto a segunda permite ordenar as alternativas segundo a sua fraqueza. Estas pré-ordens completas definem-se do seguinte modo:

a P^{+} b sse $\Phi^{+}(a) > \Phi^{+}(b)$;

a I^{+} b sse $\Phi^{+}(a) = \Phi^{+}(b)$;

a P^{-} b sse $\Phi^{-}(a) < \Phi^{-}(b)$;

a I^{-} b sse $\Phi^{-}(a) = \Phi^{-}(b)$.

A partir da informação contida nestas duas pré-ordens completas, o PROMETHEE I obtém uma pré-ordem parcial (P,I,R):

- **a** é preferível a **b** se for melhor segundo as duas pré-ordens completas, ou se for melhor numa delas e estiver *ex-aequo* com **b** na outra, ou seja,

a P b se (a P^{+} b e a P^{-} b) ou (a P^{+} b e a I^{-} b) ou (a I^{+} b e a P^{-} b);

- **a** e **b** são indiferentes uma à outra se estiverem *ex-aequo* segundo as duas pré-ordens completas, ou seja,

a I b se a I^{+} b e a I^{-} b;

- **a** e **b** são incomparáveis se uma for melhor que a outra numa das pré-ordens e pior na outra pré-ordem, ou seja,

a R b se não se verificar nem a P b nem a I b.

A pré-ordem parcial que resultaria para o exemplo da Figura 4.2 é apresentada na Figura 4.3, onde as setas indicam preferência e a ausência de setas indica incomparabilidade (neste caso não se verificou nenhuma situação de indiferença).

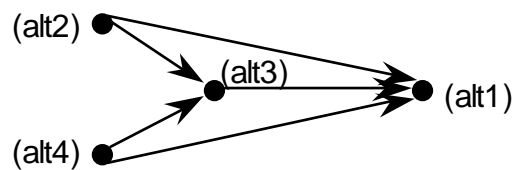


Figura 4.2 - Pré-ordem parcial (exemplo)

O PROMETHEE II permite obter uma pré-ordem completa, ordenando as alternativas pelo chamado fluxo “líquido” $\Phi(a) = \Phi^{+}(a) - \Phi^{-}(a)$:

a P b sse $\Phi(a) > \Phi(b)$ e a I b sse $\Phi(a) = \Phi(b)$.

Utilizando os dados do exemplo que se tem seguido obter-se-ia a seguinte ordenação das alternativas:

Posição 1: alt4 ($\Phi=0.728$)

Posição 2: alt2 ($\Phi=0.496$)
Posição 3: alt3 ($\Phi=-0.058$)
Posição 4: alt1 ($\Phi=-1.166$)

Compete ao decisor optar pelo uso de um, ou de ambos os métodos. Segundo [Brans e Vincke, 1985], o PROMETHEE II consegue ordenar completamente as alternativas, mas à custa de uma relação mais pobre e menos informativa face ao balanço entre a força e a fraqueza de cada alternativa.

4.3 Um algoritmo para a execução do método PROMETHEE

Esta secção apresenta um algoritmo para a execução do método PROMETHEE, elaborado de acordo com a descrição apresentada na secção anterior. O algoritmo prevê a possibilidade de explorar a relação de prevalência através do PROMETHEE I, do PROMETHEE II, ou de ambos. Utiliza-se uma linguagem algorítmica, baseada na língua portuguesa, para a especificação mais detalhada do algoritmo.

Dados de entrada

Número de critérios n (número inteiro positivo);

Número de alternativas m (número inteiro positivo);

Matriz de desempenho D ($m \times n$), cujos elementos d_{ij} (números reais) indicam o desempenho da alternativa i segundo o critério que operacionaliza o ponto de vista j , ou seja, $d_{ij} = f_j(i)$;

Caracterização dos critérios generalizados C (dado estruturado: contém o peso do critério, o tipo de critério e os parâmetros que lhe estão associados);

Algoritmo simplificado

1. Cálculo dos índices de preferência multicritério para cada par ordenado de alternativas
Para cada par ordenado de alternativas (a,b) , utiliza a informação contida em C e D para calcular uma parcela $w_j p_j [f_j(a)-f_j(b)]$ por critério, e calcula a sua soma $\pi(a,b)$.
2. Cálculo da força e da fraqueza de cada alternativa
Calcula, para cada alternativa a , $\Phi^+(a)$ e $\Phi^-(a)$.
3. Exploração através do PROMETHEE I (obtem pré-ordem parcial (P,I,R))
Para cada par ordenado (a,b) de alternativas, decide se $a P b$, $a I b$, ou $a R b$, com base em $\Phi^+(a)$, $\Phi^-(a)$, $\Phi^+(b)$ e $\Phi^-(b)$.
4. Exploração através do PROMETHEE II (obtem pré-ordem completa (P,I))
Para cada alternativa a , calcula $\Phi(a)$; ordena depois as alternativas por ordem decrescente de $\Phi(a)$.

Algoritmo detalhado

1. Cálculo dos índices de preferência multicritério para cada par ordenado de alternativas

Para cada alternativa a ($a \diamond 2, \dots, m$) faz

Para cada alternativa b menor que a ($b \diamond 1, \dots, a-1$) faz

$\pi(a,b) \diamond 0$ e $\pi(b,a) \diamond 0$

Para cada critério j ($j \diamond 1, \dots, n$) faz

Usa C e D para calcular $t \diamond w_j p_j [f_j(a) - f_j(b)]$

$\pi(a,b) \diamond \pi(a,b) + t$

Usa C e D para calcular $t \diamond w_j p_j [f_j(b) - f_j(a)]$

$\pi(b,a) \diamond \pi(b,a) + t$

fim_faz

fim_faz

fim_faz

2. Cálculo da força e da fraqueza de cada alternativa

Para cada alternativa a ($a \diamond 1, \dots, m$) faz

$\Phi^+(a) \diamond 0$ e $\Phi^-(a) \diamond 0$

Para cada alternativa b diferente de a ($b \diamond 1, \dots, m ; b \neq a$) faz

$\Phi^+(a) \diamond \Phi^+(a) + \pi(a,b)$

$\Phi^-(a) \diamond \Phi^-(a) + \pi(b,a)$

fim_faz

fim_faz

3. Exploração através do PROMETHEE I (obtem pré-ordem parcial (P,I,R))

Para cada alternativa a ($a \diamond 1, \dots, m$) faz

Para cada alternativa b diferente de a ($b \diamond 1, \dots, m ; b \neq a$) faz

Se $[\Phi^+(a) > \Phi^+(b) \text{ e } \Phi^-(a) < \Phi^-(b)]$ ou

$[\Phi^+(a) > \Phi^+(b) \text{ e } \Phi^-(a) = \Phi^-(b)]$ ou

$[\Phi^+(a) = \Phi^+(b) \text{ e } \Phi^-(a) < \Phi^-(b)]$ então resulta a P b

Se_não, Se $\Phi^+(a) = \Phi^+(b) \text{ e } \Phi^-(a) = \Phi^-(b)$ então resulta a I b

Se_não resulta a R b

fim_se

fim_faz

fim_faz

4. Exploração através do PROMETHEE II (obtem pré-ordem completa (P,I))

Para cada alternativa a ($a \in \{1, \dots, m\}$) faz

$$\Phi(a) \leftarrow \Phi^+(a) - \Phi^-(a)$$

fim_faz

Ordena as alternativas por ordem decrescente de $\Phi(a)$

4.4 Possibilidades de paralelização no método PROMETHEE

Se se encarar um algoritmo como uma sequência de tarefas a executar, então ao serem identificadas algumas tarefas cuja execução pode ser realizada em paralelo está-se a identificar uma possibilidade de paralelização. A construção de uma aplicação para executar o método PROMETHEE no computador paralelo utilizado (descrito na secção 3.6) deve ser precedida por uma identificação e análise destas possibilidades.

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, b, c e d , a avaliação do par (a,b) é independente da avaliação do par (c,d) .

No PROMETHEE obtém-se uma relação de prevalência difusa, definida pelos índices de preferência multicritério $\pi(a,b)$. O seu cálculo pode ser efectuado de modo independente para cada par ordenado (a,b) , sendo o número destes pares igual ao quadrado do número de alternativas m . Esta independência no cálculo dos índices multicritério $\pi(a,b)$ constitui por isso uma possibilidade de paralelismo não negligenciável.

A paralelização da fase seguinte (exploração da relação obtida) já não constituirá uma possibilidade de paralelismo idêntica para todos os métodos de prevalência. No caso do PROMETHEE I todas as alternativas são novamente comparadas entre si aos pares, resultando dessas comparações a pré-ordem parcial (P, I, R) , já descrita. Uma vez mais o paralelismo torna-se possível, de um modo análogo ao do cálculo dos índices de preferência multicritério, dada a independência das comparações entre pares de alternativas diversos. Contudo, há que calcular previamente a força $\Phi^+(\cdot)$ e a fraqueza $\Phi^-(\cdot)$ de cada alternativa, através de adições simples. Estas adições também podem ser realizadas em paralelo, embora seja difícil saber antecipadamente se a carga computacional em causa justificará os custos da comunicação entre os processadores. O caso do PROMETHEE II é diferente, uma vez que a operação básica é uma ordenação, que uma vez mais se pode tentar paralelizar. O cálculo prévio do fluxo "líquido" (adições/subtracções simples) pode novamente ser efectuado em paralelo.

4.5 Programas construídos

O estudo efectuado sobre a paralelização do método PROMETHEE seguiu uma estratégia de experimentação, com o intuito de observar quais das soluções de paralelização adoptadas proporcionariam melhores resultados, e em que circunstâncias. Para isso construiu-se um conjunto de programas, que se descreve nesta secção, e executou-se um plano de experiências, que se descreverá na secção seguinte.

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 (que são mais complexos), e por outro lado ser uma referência para o tempo de resposta às solicitações do utilizador, para o cálculo das medidas de desempenho.

Posteriormente iniciou-se uma paralelização das fases do método PROMETHEE em versões sucessivamente mais complexas. Por fim, e com base em resultados preliminares, simplificaram-se alguns programas paralelos, retirando-lhes funcionalidade para, em contrapartida, obter maiores velocidades de execução. Segue-se uma descrição de cada um dos programas. Dada a sua extensão, omite-se a apresentação do código dos programas (em linguagem C) ou de algoritmos detalhados.

Programa sequencial (nome abreviado: "série")

O corpo principal do programa sequencial construído segue o algoritmo mais detalhado apresentado na secção 4.3. O programa apresenta algumas funcionalidades não explicitadas nesse algoritmo, como a possibilidade de proceder a uma análise de sensibilidade após a obtenção de uma primeira solução.

A interacção com o utilizador processa-se de um modo muito simples, não mais elaborado do que o necessário para os objectivos deste estudo. As situações de decisão são especificadas através de um ficheiro de texto, cuja elaboração obedece a determinadas regras, antes da execução do programa. Um ficheiro-tipo é o correspondente ao exemplo utilizado para ilustrar o método PROMETHEE na secção 4.2, apresentado na página seguinte:

| PROBLEMA exemplo | | | | |
|------------------|------|-------|-------|-----------|
| CRITÉRIOS | peso | tipo | par1 | par2 |
| n=3 | | | | |
| crit1 | 0.25 | 1 | | |
| crit2 | 0.5 | 6 | 10 | |
| crit3 | 0.25 | 4 | 2 | 4 |
| ALTERNATIVAS | | crit1 | crit2 | crit3 ... |
| m=4 | | | | |
| alt1 | | 2 | 79 | 7 |
| alt2 | | 3 | 99 | 1 |
| alt3 | | 2 | 96 | 4 |
| alt4 | | 3 | 90 | 6 |

Quando o programa é executado, através da escrita do seu nome na linha de comando do sistema operativo do computador hospedeiro, surgem ao utilizador as seguintes opções:

- "Ler problema", permite ler para as estruturas de dados do programa o conteúdo de um ficheiro de texto análogo ao representado acima;
- "Mostrar problema corrente", permite visualizar os dados relativos a uma situação de decisão contidos nas estruturas de dados do programa;
- "Resolver problema corrente", permite executar o método PROMETHEE (I, II ou ambos) utilizando os dados da situação de decisão corrente e posteriormente efectuar uma análise de sensibilidade à caracterização dos critérios (tipo, peso, limiares); a análise de sensibilidade pode afectar vários destes parâmetros de cada vez e envolver um ou mais critérios, existindo duas rotinas distintas (Cálculo Normal e Cálculo Abreviado) para o cálculo dos índices de preferência multicritério na fase de análise de sensibilidade, consoante seja necessário ou não recalculer os índices para mais que um critério;
- "Guardar problema corrente", permite guardar a situação corrente, considerando as alterações introduzidas durante as análises de sensibilidade, num ficheiro de texto compatível com o de entrada de dados e com um nome à escolha;
- "Sair do programa".

Os diversos programas paralelos (pp1 a pp3) são, no que respeita à interacção com o utilizador, em tudo semelhantes ao programa sequencial. A única diferença observável será a velocidade de execução, que varia de programa para programa. Deste modo, tudo o que se escreveu acima permanece válido para os programas paralelos.

Primeiro programa paralelo (nome abreviado: "pp1")

O primeiro programa paralelo (pp1) foi construído a partir do programa sequencial, paralelizando apenas a obtenção dos índices de preferência multicritério $\pi(a,b)$, isto é, o primeiro ponto do algoritmo apresentado em 4.3.

O programa utiliza a topologia física da rede de transputers, através da qual envia os dados e recolhe os resultados. Em cada transputer é executado um processo idêntico,

que contém instruções que são condicionalmente executadas dependendo da posição do nó na rede. O processo que estiver a ser executado no nó de interface com o hospedeiro interage directamente com o utilizador e lê o ficheiro de texto, cujos dados guarda e retransmite aos seus vizinhos. Os restantes processos, no seu início, limitam-se a receber e retransmitir os dados, de acordo com o padrão ilustrado na Figura 4.3, que possibilita um algoritmo de encaminhamento muito simples e a minimização da distância máxima a percorrer por uma mensagem.

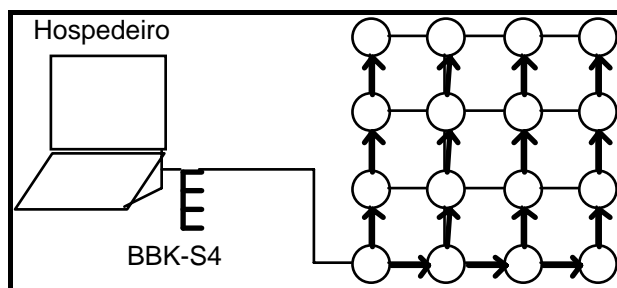


Figura 4.3 - Difusão dos dados do nó interface para os restantes

A fase de obtenção da relação de prevalência de síntese consiste em calcular $\pi(a,b)$ para todo o par ordenado de alternativas $(a,b) \in A \times A$, em que $A \times A$ representa o conjunto $\{(a,b): a \in A, b \in A\}$. A paralelização desta fase consistiu no cálculo simultâneo de vários pares ordenados, atribuindo-se um subconjunto de $A \times A$ a cada processador. A partição de $A \times A$ efectua-se de modo a atribuir cargas semelhantes aos diferentes processadores, como se ilustra no quadro seguinte para um exemplo com 100 alternativas (1 a 100) e uma topologia de rede matricial 4 x 4.

| | | | |
|---|--|--|---|
| PROCESSADOR 12 a: 76 a 100 x b: 1 a 25 | PROCESSADOR 13 a: 76 a 100 x b: 26 a 50 | PROCESSADOR 14 a: 76 a 100 x b: 51 a 75 | PROCESSADOR 15 a: 76 a 100 x b: 76 a 100 |
| PROCESSADOR 8 a: 51 a 75 x b: 1 a 25 | PROCESSADOR 9 a: 51 a 75 x b: 26 a 50 | PROCESSADOR 10 a: 51 a 75 x b: 51 a 75 | PROCESSADOR 11 a: 51 a 75 x b: 76 a 100 |
| PROCESSADOR 4 a: 26 a 50 x b: 1 a 25 | PROCESSADOR 5 a: 26 a 50 x b: 26 a 50 | PROCESSADOR 6 a: 26 a 50 x b: 51 a 75 | PROCESSADOR 7 a: 26 a 50 x b: 76 a 100 |
| PROCESSADOR 0* a: 1 a 25 x b: 1 a 25 | PROCESSADOR 1 a: 1 a 25 x b: 26 a 50 | PROCESSADOR 2 a: 1 a 25 x b: 51 a 75 | PROCESSADOR 3 a: 1 a 25 x b: 76 a 100 |

* Processador de interface com o hospedeiro

Neste exemplo, de entre os 10000 elementos do conjunto $\{(a,b): a \in A, b \in A\}$, seria necessário calcular o índice de preferência multicritério para 9900, dado que 100 seriam de cálculo trivial por corresponderem ao conjunto $\{(a,a): a \in A\}$. A cada processador caberia apenas o cálculo de 25 x 25 pares ordenados na pior das hipóteses, de

acordo com o quadro acima. Note-se que o número de elementos de $A \times A$ não tem de ser múltiplo do número de processadores: as expressões seguintes (em linguagem C) tentam assegurar sempre uma partição o mais equitativa possível.

```

lsupx = (n_alt/dim_x)*(gridx+1);
lsupx += (gridx+1>n_alt%dim_x)?n_alt%dim_x:gridx+1;
lsupy = (n_alt/dim_y)*(gridy+1);
lsupy += (gridy+1>n_alt%dim_y)?n_alt%dim_y:gridy+1;
linfx = (n_alt/dim_x)*gridx + 1;
linfx += (gridx>n_alt%dim_x)?n_alt%dim_x:gridx;
linfy = (n_alt/dim_y)*gridy + 1;
linfy += (gridy>n_alt%dim_y)?n_alt%dim_y:gridy;

```

Em que:

- a rede matricial tem dim_x colunas e dim_y linhas;
- cada processador possui uma coordenada $(gridx, gridy)$, correspondendo ao processador 0 a coordenada $(0,0)$;
- n_alt representa o número de alternativas;
- o processador de coordenada $(gridx, gridy)$ possuirá um subconjunto de alternativas correspondente a $\{linfx, \dots, lsupx\} \times \{linfy, \dots, lsupy\}$.

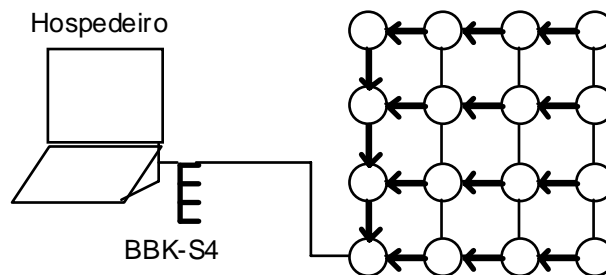


Figura 4.4 - "Recolha" dos resultados pelo nó interface

Cada processador i calcula então $\pi(a,b)$ para todos os pares (a,b) contidos num conjunto que se designará por P_i . A partição efectuada verifica $A \times A = \{a_1, \dots, a_m\} \times \{a_1, \dots, a_m\} = P_0 \approx P_1 \approx \dots \approx P_{p-1}$. Considere-se, sem perda de generalidade, um qualquer processador i para o qual $P_i = \{a_r, a_{r+1}, \dots, a_s\} \times \{a_t, a_{t+1}, \dots, a_u\}$, com $1 \leq r,s,t,u \leq m$. À medida que o processador i calcula $\pi(a,b)$ para cada par ordenado $(a,b) \in P_i$, envia o resultado para o nó interface (isto para os casos em que $a \neq b$, dado que quando $a = b$ o nó interface já sabe que o índice de preferência multicritério é nulo). O facto de se efectuar comunicação logo que cada resultado é calculado aproveita a capacidade que os transputers possuem de comunicar e efectuar cálculos em paralelo. O programa calcula assim o próximo resultado enquanto se processa o protocolo de comunicação.

Cada processo contém fios de processamento específicos para encaminhar estes resultados de acordo com o padrão ilustrado na Figura 4.4. Existe um fio por cada ligação pela qual chegam mensagens e cada um executa um algoritmo muito simples:

1. Aguarda mensagem vinda da ligação X

2. Se o processo não se situa no nó interface (processador zero)
Então Envia mensagem recebida pela ligação Y
Se_não Actualiza estrutura de resultados e assinala
mais uma chegada.

Naturalmente, o número de fios de processamento e as ligações correspondentes a X e Y dependerão da posição do nó na rede. Contudo, não há necessidade de alterar o código quando se muda a dimensão da topologia em rede matricial utilizada.

Esta fase termina assim que o nó interface (nó 0) receba todos os resultados provenientes dos restantes nós e termine o cálculo relativo aos pares ordenados de P_0 . Procede-se então à fase de exploração da relação obtida através do PROMETHEE I e/ou PROMETHEE II, conforme o desejo do utilizador. Durante a fase de exploração o processador interface executa de forma sequencial os passos 2 a 4 do algoritmo apresentado na secção 4.3, enquanto os restantes processadores permanecem inactivos.

No caso de o utilizador proceder a uma análise de sensibilidade tudo se passa como no cálculo da solução inicial, mas com uma importante diferença: agora já não é necessário divulgar aos processadores todos os dados relativos ao problema, que já são conhecidos, bastando comunicar as alterações introduzidas pelo utilizador. No caso de as alterações afectarem apenas um dos critérios cada processador executa uma rotina de cálculo mais simples (cálculo abreviado). Caso contrário, utilizam a rotina de cálculo normal.

O programa pp1 é pouco ambicioso, dado que apenas paraleliza uma parte do algoritmo do método PROMETHEE (que é, no entanto, a parte mais pesada em termos computacionais conforme se verá adiante). Na melhor das hipóteses, ao utilizar p processadores esta parte seria executada p vezes mais rapidamente. No entanto, esta hipótese é muito irrealista, sobretudo quando se calcula a solução inicial, dada a enorme quantidade de comunicação presente na rede (tanto para envio dos dados como para recolha dos resultados). A fase de exploração da relação de prevalência é executada pelo nó de interface à mesma velocidade do programa sequencial. Escolheu-se o nó de interface para executar esta fase, sendo assim um nó pivot em todo o processo, dado que, estando directamente ligado ao hospedeiro, é aquele que mais rapidamente comunica com o exterior.

Segundo programa paralelo (nome abreviado: "pp2")

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 do PROMETHEE I - pontos 2 e 3 do algoritmo apresentado em 4.3 - mantendo a execução em modo paralelo do ponto 1 e a execução em modo sequencial do ponto 4.

Considere-se a partição, que se mantém em vigor, efectuada aquando do cálculo em paralelo dos índices de preferência multicritério. Considere-se, sem perda de generalidade, um qualquer processador i , ao qual coube um conjunto de pares ordenados $P_i = \{a_r, a_{r+1}, \dots, a_s\} \times \{a_t, a_{t+1}, \dots, a_u\}$, com $1 \leq r, s, t, u \leq m$. Na fase de exploração da relação de prevalência através do PROMETHEE I o processador i deve decidir, para todos os pares ordenados $(a, b) \in P_i$, se $a P b$, $a I b$ ou $a R b$, a partir de $\Phi^+(a)$, $\Phi^-(a)$, $\Phi^+(b)$ e $\Phi^-(b)$. Será para isso previamente necessário que esse processador obtenha a força e a fraqueza de todas as alternativas $x \in \{a_r, a_{r+1}, \dots, a_s\} \approx \{a_t, a_{t+1}, \dots, a_u\}$. Descreve-se a seguir como são efectuados os cálculos, considerando o processador i , apresentando-se depois um exemplo.

No programa pp2, força e fraqueza de cada alternativa são obtidas em modo paralelo. Após calcular os índices de preferência multicritério $\pi(a, b)$ para todos os pares ordenados $(a, b) \in P_i$, o processador calculará uma força parcial e uma fraqueza parcial tendo em conta os dados que já calculou. Deste modo, continuando a considerar a partição $P_i = \{a_r, a_{r+1}, \dots, a_s\} \times \{a_t, a_{t+1}, \dots, a_u\}$, obtém

$$\Phi_i^+(a_j) = \pi(a_j, a_t) + \pi(a_j, a_{t+1}) + \dots + \pi(a_j, a_u), \text{ para } j = r, r+1, \dots, s, \text{ e}$$

$$\Phi_i^-(a_k) = \pi(a_r, a_k) + \pi(a_{r+1}, a_k) + \dots + \pi(a_s, a_k), \text{ para } k = t, t+1, \dots, u.$$

À medida que vão sendo calculadas, as forças parciais $\Phi_i^+(a_j)$ ($j = r, r+1, \dots, s$) são enviadas para todos os processadores na mesma linha e as fraquezas parciais $\Phi_i^-(a_k)$ ($k = t, t+1, \dots, u$) são enviadas para todos os processadores na mesma coluna. Em paralelo, o processador recebe dos outros processadores na mesma linha as forças parciais $\Phi_l^+(a_j)$ ($j = r, r+1, \dots, s$) e recebe dos outros processadores na mesma coluna as fraquezas parciais $\Phi_c^-(a_k)$ ($k = t, t+1, \dots, u$). Finalmente, o processador i soma os parciais recebidos aos parciais que calculou, conseguindo deste modo obter parte das forças e fraquezas pretendidas:

$$\Phi^+(a_j) = \sum \Phi_l^+(a_j), \text{ para } j = r, r+1, \dots, s; \text{ e}$$

$$\Phi^-(a_k) = \sum \Phi_c^-(a_k), \text{ para } k = t, t+1, \dots, u.$$

No entanto, é necessário conhecer $\Phi^+(x)$ e $\Phi^-(x)$ para todo $x \in \{a_r, a_{r+1}, \dots, a_s\} \approx \{a_t, a_{t+1}, \dots, a_u\}$, ou seja, pode ainda faltar conhecer $\Phi^+(a_k)$ para $k = t, t+1, \dots, u$ e $\Phi^-(a_j)$ para $j = r, r+1, \dots, s$. Por esse motivo, no programa pp2 cada processador envia $\Phi^+(y)$, para todo $y \in \{a_r, a_{r+1}, \dots, a_s\} \leftrightarrow \{a_t, a_{t+1}, \dots, a_u\}$, aos restantes processadores na mesma coluna e $\Phi^-(y)$, para todo $y \in \{a_r, a_{r+1}, \dots, a_s\} \leftrightarrow \{a_t, a_{t+1}, \dots, a_u\}$, aos processadores na mesma linha. Note-se que só haverá alguns processadores para os quais $\{a_r, a_{r+1}, \dots, a_s\} \leftrightarrow \{a_t, a_{t+1}, \dots, a_u\}$ não é um conjunto vazio.

O quadro seguinte pretende ilustrar este processo, considerando o exemplo do quadro anterior, do ponto de vista do processador nº 6.

| | | | |
|---|--|--|---|
| PROCESSADOR 12 $\Phi_{12}^+ : j = 76, \dots, 100$ $\Phi_{12}^- : k = 1, \dots, 25$ | PROCESSADOR 13 $\Phi_{13}^+ : j = 76, \dots, 100$ $\Phi_{13}^- : k = 26, \dots, 50$ | PROCESSADOR 14 $\Phi_{14}^+ : j = 76, \dots, 100$ $\Phi_{14}^- : k = 51, \dots, 75$ | PROCESSADOR 15 $\Phi_{15}^+ : j = 76, \dots, 100$ $\Phi_{15}^- : k = 76, \dots, 100$ |
| PROCESSADOR 8 $\Phi_8^+ : j = 51, \dots, 75$ $\Phi_8^- : k = 1, \dots, 25$ | PROCESSADOR 9 $\Phi_9^+ : j = 51, \dots, 75$ $\Phi_9^- : k = 26, \dots, 50$ | PROCESSADOR 10 $\Phi_{10}^+ : j = 51, \dots, 75$ $\Phi_{10}^- : k = 51, \dots, 75$ | PROCESSADOR 11 $\Phi_{11}^+ : j = 51, \dots, 75$ $\Phi_{11}^- : k = 76, \dots, 100$ |
| PROCESSADOR 4 $\Phi_4^+ : j = 26, \dots, 50$ $\Phi_4^- : k = 1, \dots, 25$ | PROCESSADOR 5 $\Phi_5^+ : j = 26, \dots, 50$ $\Phi_5^- : k = 26, \dots, 50$ | PROCESSADOR 6 $\Phi_6^+ : j = 26, \dots, 50$ $\Phi_6^- : k = 51, \dots, 75$ | PROCESSADOR 7 $\Phi_7^+ : j = 26, \dots, 50$ $\Phi_7^- : k = 76, \dots, 100$ |
| PROCESSADOR 0 $\Phi_0^+ : j = 1, \dots, 25$ $\Phi_0^- : k = 1, \dots, 25$ | PROCESSADOR 1 $\Phi_1^+ : j = 1, \dots, 25$ $\Phi_1^- : k = 26, \dots, 60$ | PROCESSADOR 2 $\Phi_2^+ : j = 1, \dots, 25$ $\Phi_2^- : k = 51, \dots, 75$ | PROCESSADOR 3 $\Phi_3^+ : j = 1, \dots, 25$ $\Phi_3^- : k = 76, \dots, 100$ |

Para calcular se $a \in P$, $b \in I$ ou $a \in R$, para cada par ordenado (a,b) , com $a=26, \dots, 50$ e $b=51, \dots, 75$, o processador 6 tem de obter $\Phi^+(a_x)$ e $\Phi^-(a_x)$, para todo o $x \in \{26, \dots, 50\} \approx \{51, \dots, 75\}$. De acordo com o modo de proceder atrás descrito, o nó 6 obterá:
 $\Phi^+(a_j) = \Phi_4^+(a_j) + \Phi_5^+(a_j) + \Phi_6^+(a_j) + \Phi_7^+(a_j)$, para $j = 26, \dots, 50$; e
 $\Phi^-(a_k) = \Phi_2^-(a_k) + \Phi_6^-(a_k) + \Phi_{10}^-(a_k) + \Phi_{14}^-(a_k)$, para $k = 51, \dots, 75$.

Do mesmo modo, os restantes processadores obteriam as forças $\Phi^+(a_j)$ e fraquezas $\Phi^-(a_k)$ para algumas alternativas. É necessário, porém, que alguns processadores conheçam mais algumas forças e fraquezas. O processador 6, por exemplo, ainda não conhece $\Phi^+(a_k)$ para $k = 51, \dots, 75$ e $\Phi^-(a_j)$ para $j = 26, \dots, 50$. Os valores $\Phi^+(a_k)$ para $k = 51, \dots, 75$ serão comunicados por um dos processadores que os tenha calculado. Pode verificar-se no quadro anterior que qualquer dos processadores na linha 8-9-10-11 poderia enviar os dados necessários. Contudo, para impedir redundância na comunicação só um destes enviará esses dados, a saber, aquele que conhecer a fraqueza da alternativa cuja força vai enviar (processador 10). Segundo outra perspectiva, se a partição $P_i = \{a_r, a_{r+1}, \dots, a_s\} \times \{a_t, a_{t+1}, \dots, a_u\}$ do processador i for tal que $I_i = \{a_r, a_{r+1}, \dots, a_s\} \leftrightarrow \{a_t, a_{t+1}, \dots, a_u\}$ seja não vazio, então cabe a esse processador comunicar $\Phi^+(y)$ para todo o $y \in I_i$ aos processadores na mesma coluna e $\Phi^-(y)$ para todo o $y \in I_i$ aos processadores na mesma linha. Neste exemplo, estarão nestas condições os processadores na diagonal 0-5-10-15. Deste modo, o processador 6 receberia do processador 10 os valores $\Phi^+(a_k)$ para $k = 51, \dots, 75$ e receberia do processador 5 os valores $\Phi^-(a_j)$ para $j = 26, \dots, 50$.

Finalmente, todos os processadores ficam aptos a determinar a porção da pré-ordem parcial correspondente aos pares ordenados que lhe estão atribuídos. Cabe a cada processador enviar ao nó interface (processador 0) dois tipos de resultado, o que é feito de acordo com o padrão ilustrado na Figura 4.4. Um tipo de resultados é $\Phi^+(a)$ e $\Phi^-(a)$, de

que o nó interface necessita para calcular o fluxo líquido $\Phi(a)$ e proceder à obtenção em modo sequencial da pré-ordem completa do PROMETHEE II. O outro tipo de resultados que o nó interface necessita receber, para mostrar ao utilizador, é se $a P b$, $a I b$ ou $a R b$ para cada par ordenado $(a,b) \in A \times A$. Cada processador i calcula esta relação para $(a,b) \in P_i$, mas não envia necessariamente esse resultado, de modo a evitar redundância: veja-se por exemplo no quadro anterior que tanto o processador 3 como o processador 12 podem saber se $a_{20} P a_{80}$, $a_{80} P a_{20}$, $a_{20} I a_{80}$ ou $a_{20} R a_{80}$ e enviar o resultado ao nó de interface. A regra que os processadores seguem neste programa é a seguinte: dadas duas alternativas a_j e a_k , envia resultado $a_j H a_k$ (em que H é uma relação binária pertencente ao conjunto $\{P,I,R\}$) se e só se $H = P$ ou $(H = I \text{ e } j < k)$. Por outras palavras, após calcular $a_j H a_k$ um processador comporta-se de acordo com o seguinte:

- se $a_j P a_k$ então envia este resultado ao nó de interface;
- se $a_j I a_k$ então só envia este resultado ao nó de interface se $j < k$;
- se $a_j R a_k$ então não envia este resultado ao nó de interface.

Os custos de evitar a comunicação de resultados redundantes consistem na necessidade de fazer cada processador enviar ao processador de interface uma mensagem especial a indicar que não há mais resultados a enviar. No entanto, este acréscimo, de uma mensagem por processador, parece ser pouco importante face à comunicação que se evita.

O programa pp2 é mais complexo do que o pp1, dado que paraleliza todo o método PROMETHEE I. Só a fase de exploração da relação de prevalência pelo PROMETHEE II é executada pelo nó de interface, à mesma velocidade do programa sequencial.

Terceiro programa paralelo (nome abreviado: "pp3")

A terceira e última estratégia adoptada (pp3) foi a de paralelizar adicionalmente 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. O programa pp3 efectua portanto em modo paralelo todos os pontos (1 a 4) do algoritmo apresentado em 4.3.

A paralelização do ponto 4 do algoritmo inicia-se com o cálculo em paralelo do fluxo líquido $\Phi(a) = \Phi^+(a) - \Phi^-(a)$ de todas as alternativas de A . De acordo com o descrito para o programa pp2, que se aplica também para o pp3, a força $\Phi^+(a)$ e fraqueza $\Phi^-(a)$ de todas as alternativas são calculadas em paralelo. Através do texto e dos quadros

apresentados para o programa pp2, pode-se inferir que, qualquer que seja a linha ou a coluna de processadores, nesta encontram-se $\Phi^+(a)$ e $\Phi^-(a)$ para todo o $a \in A$. Para a exploração através do PROMETHEE II basta por isso escolher uma linha ou coluna de processadores, tendo-se escolhido a linha do processador 0, apenas por estar próxima do interface. Segundo o exemplo apresentado para o programa pp2 essa linha seria a seguinte:

| PROCESSADOR 0 | PROCESSADOR 1 | PROCESSADOR 2 | PROCESSADOR 3 |
|------------------------------|------------------------------|------------------------------|------------------------------|
| ($\Phi: j = 1, \dots, 25$) | ($\Phi: j = 1, \dots, 25$) | ($\Phi: j = 1, \dots, 25$) | ($\Phi: j = 1, \dots, 25$) |
| $\Phi: k = 1, \dots, 25$ | $\Phi: k = 26, \dots, 60$ | $\Phi: k = 51, \dots, 75$ | $\Phi: k = 76, \dots, 100$ |

Mantendo-se válida a partição, $P_i = \{a_r, a_{r+1}, \dots, a_s\} \times \{a_t, a_{t+1}, \dots, a_u\}$ para um dado processador i na linha escolhida, esse processador efectua a subtracção $\Phi^+(a_k) - \Phi^-(a_k)$ para todo o $k = t, t+1, \dots, u$. De seguida, ordena as alternativas a_t, a_{t+1}, \dots, a_u , por ordem decrescente do seu fluxo líquido, e envia-as por ordem ao processador vizinho na direcção do processador 0. Obtém-se assim no nó de interface (processador 0) uma ordenação de todas as alternativas, com possíveis *ex-aequo*, que define a pré-ordem completa do PROMETHEE II.

O programa pp3 é o mais complexo de todos, dado que paraleliza todo o método PROMETHEE (I e II).

Variantes dos programas construídos

Os programas paralelos pp1, pp2 e pp3 envolvem grande quantidade de comunicação entre processadores, especialmente a dirigida para o nó de interface, o que lhes degrada o desempenho. Por esse motivo, construíram-se algumas variantes para alguns destes programas que, à custa de uma maior especialização, obtêm uma redução significativa da comunicação entre processadores. Por maior especialização entende-se a incapacidade de obter algumas saídas (resultados) do método PROMETHEE, em oposição aos programas anteriores, generalistas, que permitem obter todos os resultados do método: índices de preferência multicritério, pré-ordem parcial do PROMETHEE I e pré-ordem completa do PROMETHEE II.

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 $\pi(a,b)$ (para $(a,b) \in A \times A$). Os índices de preferência são intrinsecamente técnicos, constituindo muitas vezes um resultado intermédio que o decisor não pretende analisar. No caso da já referida aplicação BANKADVISER, o decisor é apenas confrontado com os resultados da exploração pelo PROMETHEE II, isto é, com uma ordenação das alternativas sem incomparabilidades. Pode-se mesmo antever que, quando o número de alternativas for

muito grande (várias centenas), tanto os índices de preferência como os resultados obtidos pelo PROMETHEE I serão possivelmente muito pouco inteligíveis.

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 nó de interface os índices de preferência multicritério. Por esse motivo, a comunicação com o processador 0 no final do ponto 1 do algoritmo apresentado na secção 4.3 é eliminada, e o utilizador não pode conhecer estes índices: apenas pode conhecer os resultados da exploração da relação por estes definida. Não se elaborou um programa sequencial com estas características, dado que não haveria nenhuma alteração nos tempos de execução. Também não se construiu um programa pp1\R, porque neste programa o nó de interface tem de receber os índices de preferência para proceder à exploração pelo PROMETHEE I e/ou II.

Noutras situações, é possível que o utilizador, para além de não necessitar de conhecer os índices de preferência, apenas deseje conhecer os resultados da exploração da relação de prevalência fornecidos por uma das versões do PROMETHEE. Pretenderá, portanto, apenas conhecer os resultados do PROMETHEE I ou apenas os resultados do PROMETHEE II. No primeiro destes casos, os programas pp2\R e pp3\R não necessitam de ser alterados, dado que não seriam muito mais rápidos se não pudessem calcular a pré-ordem completa do PROMETHEE II. No segundo caso isso já não é verdade: o facto de o utilizador "dispensar" o programa do cálculo da pré-ordem parcial do PROMETHEE I faz desaparecer a necessidade de determinar essa relação e de a comunicar ao processador 0. Pelos motivos apresentados anteriormente, só os programas paralelos 2 e 3 poderão beneficiar de tal especialização. Esses programas, designados por pp2\RI e pp3\RI, só oferecem a possibilidade de fornecer os resultados pela exploração da relação de prevalência pelo PROMETHEE II. Em contrapartida, no programa pp2\RI é apenas necessário que cada processador i comunique ao interface os números $\Phi_i^+(\cdot)$ e $\Phi_i^-(\cdot)$ que pode calcular, sendo a pré-ordem completa definida pelo nó de interface. No programa pp3\RI a pré-ordem completa é definida em paralelo, de acordo com o descrito para o programa pp3, dispensando-se toda a comunicação com o interface excepto a do envio das alternativas por ordem decrescente do seu fluxo líquido $\Phi(\cdot)$.

Apresentam-se de seguida dois quadros que permitem uma visualização global dos programas construídos. O primeiro quadro indica os pontos do algoritmo apresentado na secção 4.3 que são executados em modo paralelo, em cada programa. O segundo quadro indica as saídas (resultados disponíveis) oferecidas pelos programas paralelos 2 e 3.

| | série | pp1 | pp2 | pp3 |
|---|------------|----------|----------|----------|
| 1. Cálculo dos índices de preferência multicritério | sequencial | paralelo | paralelo | paralelo |

| | | | | |
|---|------------|------------|------------|----------|
| 2. Cálculo da força e da fraqueza de cada alternativa | sequencial | sequencial | paralelo | paralelo |
| 3. Exploração através do PROMETHEE I | sequencial | sequencial | paralelo | paralelo |
| 4. Exploração através do PROMETHEE II | sequencial | sequencial | sequencial | paralelo |

| | sem sufixo | sufixo \R | sufixo \RI |
|--------------------------------------|------------|-----------|------------|
| Índices de preferência multicritério | Sim | Não | Não |
| Exploração através do PROMETHEE I | Sim | Sim | Não |
| Exploração através do PROMETHEE II | Sim | Sim | Sim |

4.6 Experiências efectuadas

Foi planeado um conjunto de experiências que permitisse aferir as vantagens das paralelizações efectuadas, em diversas situações. As situações que podem influenciar o desempenho dos programas são o tamanho do problema (que no caso do método PROMETHEE depende do número de alternativas, do número de critérios e do seu tipo), o número de processadores utilizado e a possibilidade de utilizar versões especializadas dos programas. Os aspectos focados por este estudo experimental foram:

- Tipo dos critérios do problema;
- Número de critérios do problema;
- Número de alternativas do problema;
- Número de processadores utilizado;
- Saídas pretendidas pelo agente de decisão (relação de prevalência, pré-ordem parcial do PROMETHEE I, pré-ordem completa do PROMETHEE II).

São muitas as variáveis a controlar, pelo que o estudo efectuado não pretendeu ser exaustivo. No que respeita à variação do tipo de critérios geraram-se dois bancos de problemas. O banco G_s possui problemas cujo tipo de critério varia entre 1 e 6, enquanto no banco G o tipo de critério varia entre 4 e 6 (que são os critérios mais complexos). O tipo dos critérios foi uniformemente distribuído dentro da gama 1-6 e 4-6, respectivamente, para os problemas do banco G_s e G.

Os bancos G_s e G são compostos por 20 problemas cada, correspondentes à combinação de um número de critérios igual a 5, 10, 50 ou 100, com um número de alternativas igual a 5, 10, 50, 100 e 500. Utilizou-se somente um problema para cada dimensão, contendo dados fictícios.

Conduziram-se três conjuntos de experiências:

- 1) Todos os problemas contidos nos bancos G_s e G foram executados ao nível máximo de exigência (relação de prevalência, PROMETHEE I e PROMETHEE II) pelos

programas série (utilizando apenas o processador 0), pp1, pp2 e pp3 (utilizando todos os 16 processadores).

- 2) Experimentou-se de seguida executar todos os problemas do banco G ao nível máximo de exigência (relação de prevalência, PROMETHEE I e PROMETHEE II) variando o número de processadores utilizado pelos programas paralelos. Compararam-se os desempenhos em quatro topologias: rede 4 x 4 (16 processadores), rede 4 x 2 (8 processadores, dos quais 4 na linha do nó de interface), rede 2 x 4 (8 processadores, dos quais 2 na linha do nó de interface) e rede 2 x 2 (4 processadores).
- 3) Por fim, experimentou-se executar todos os problemas do banco G a diferentes níveis de exigência. Foram usadas as variantes dos programas paralelos 2 e 3, e experimentou-se apenas com a topologia 4 x 4.

Para todos estes casos, os tempos de execução medidos referem-se à situação em que se usa o cálculo normal e à situação em que se usa o cálculo abreviado dos índices $\pi(a,b)$, isto é, a situação em que na análise de sensibilidade apenas se altera o tipo ou parâmetro de um único critério, sem lhe alterar o peso.

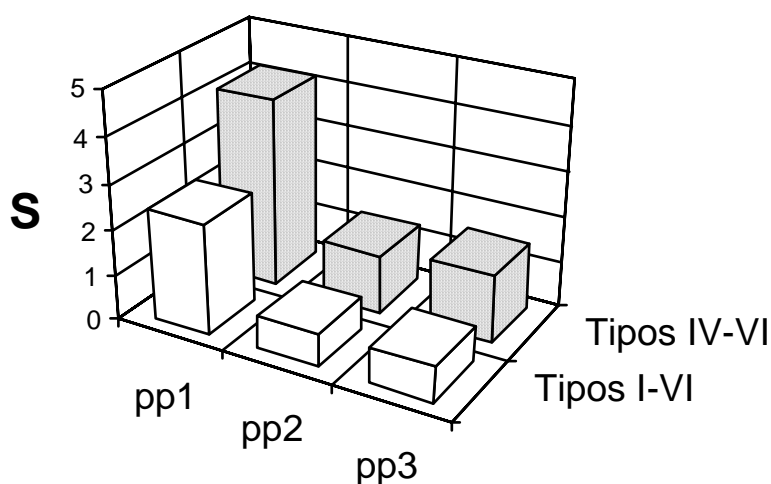
Os tempos de execução foram medidos utilizando o relógio de tempo real do PARIX, que permite medir tempos com um erro máximo de 64 μ s. A medida de desempenho (ver sec. 3.3) escolhida para avaliar o mérito relativo de cada paralelização foi o speedup real $S(n,p)$, ou seja a razão entre os tempos de execução usando o programa sequencial e os tempos de execução usando um programa paralelo. Usar-se-á por comodidade a expressão "razão S" (ou "speedup"), estando o tamanho do programa e o número de processadores implícitos no contexto da expressão.

4.7 Resultados obtidos

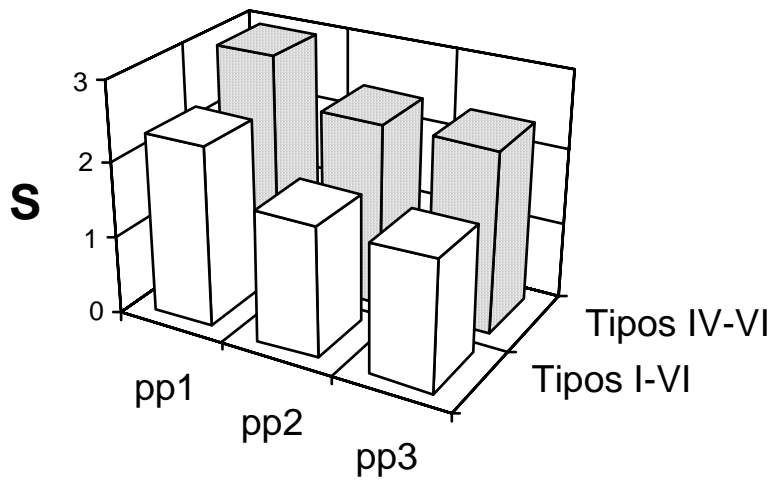
Conjunto 1) de experiências: utilização dos 16 processadores para calcular os índices de preferência, o PROMETHEE I e o PROMETHEE II

As Figuras 4.5a) a 4.5d) apresentam a razão S , calculada a partir dos tempos de execução medidos, para os quatro problemas “extremos” de cada banco, quando se utilizam 16 processadores para executar os programas pp1, pp2 e pp3.

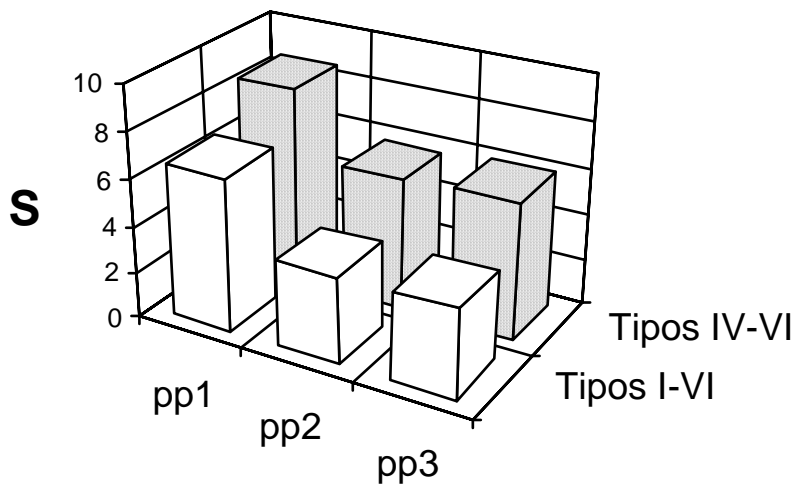
Verificou-se que o banco de problemas G (com critérios de tipo 4 a 6) é aquele que demorou mais tempo a executar e é também aquele que proporcionou maiores contrapartidas pelo uso do paralelismo. Tal não surpreende, uma vez que, mantendo-se constante a comunicação entre processadores, a razão S referida acima será tanto maior quanto a quantidade de cálculos a efectuar em cada processador. Como semelhantes conclusões são aplicáveis aos restantes casos a experimentar, optou-se por usar apenas o banco de problemas G. Esta opção justifica-se por ser o banco que contém os problemas de execução mais demorada (veja-se o Quadro 2 do Apêndice A) e pelo facto de os critérios de tipo 4 a 6 generalizarem os restantes. Existem mesmo aplicações onde apenas estes tipos de critério são referidos e utilizados (cf. [Mladineo et al. 1987]). Pelo mesmo motivo, a razão S obtida quando se usou o cálculo abreviado na análise de sensibilidade foi inferior à obtida com o cálculo normal, referindo-se os gráficos apresentados ao caso mais complexo, aquele que se refere ao caso em que um peso ou mais do que um critério são alterados numa iteração da análise de sensibilidade. Apresentam-se estes resultados no Apêndice A, Quadros 3 a 5.



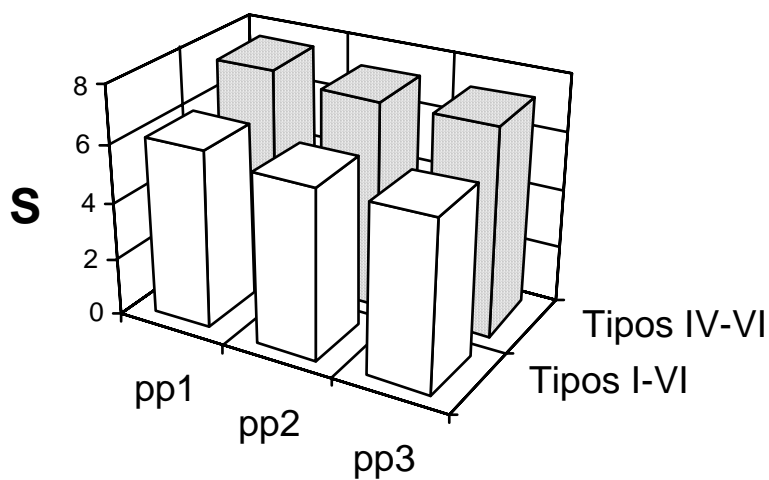
Figuras 4.5a) - Problema com 5 critérios e 5 alternativas



Figuras 4.5b) - Problema com 5 critérios e 500 alternativas

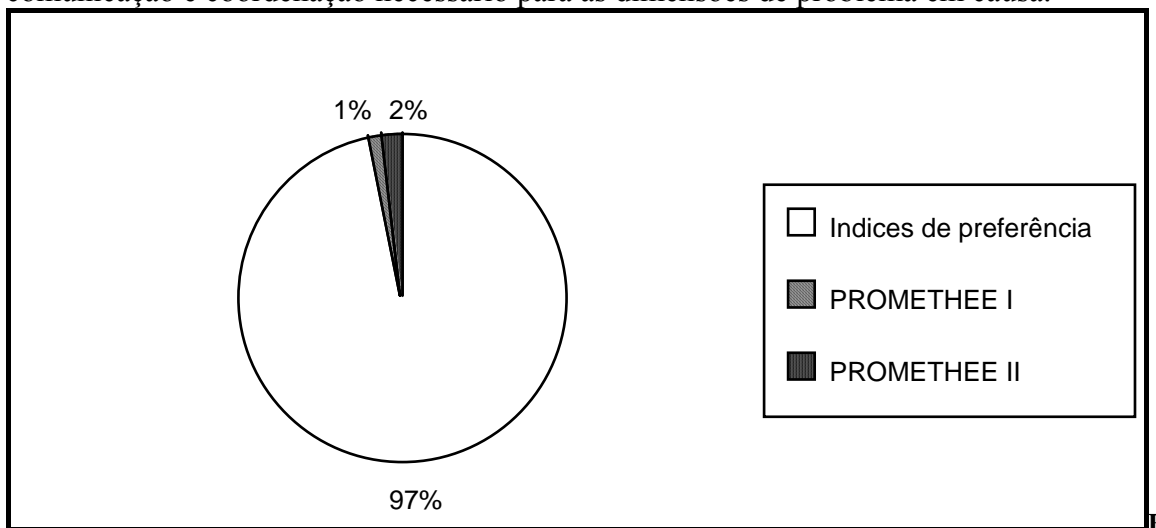


Figuras 4.5c) - Problema com 100 critérios e 5 alternativas



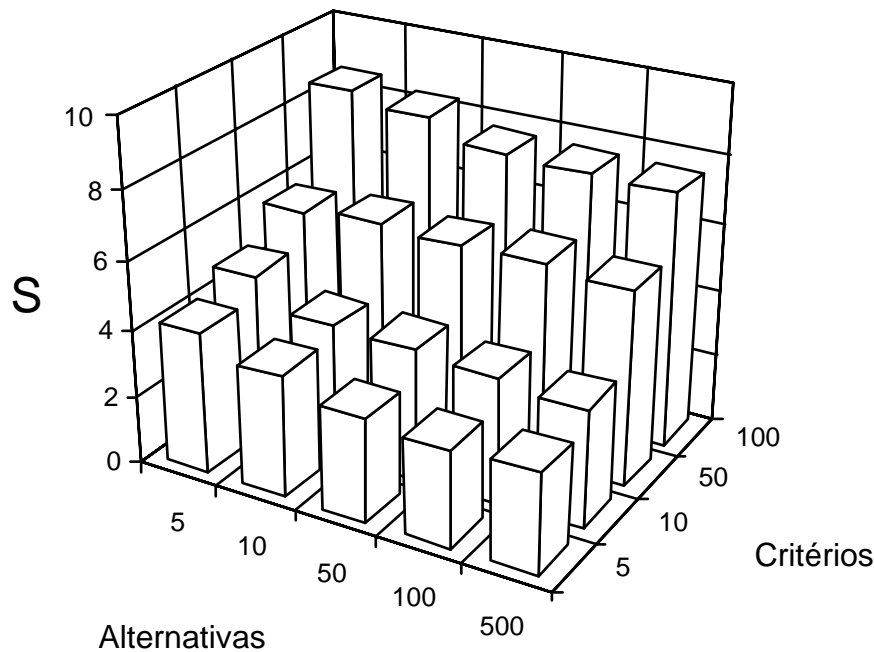
Figuras 4.5d) - Problema com 100 critérios e 500 alternativas

O programa pp1 (o mais simples) obteve melhores desempenhos que os programas pp2 e pp3, que se mostraram equivalentes. Este resultado, à partida inesperado, é justificável pelo facto do cálculo dos índices de preferência ser absolutamente predominante no tempo de execução dos problemas. De facto, a execução dos diversos problemas usando um só processador revelou que o tempo utilizado no cálculo dos índices de preferência $\pi(a,b)$ é bastante elevado, nunca inferior a 96.8% do tempo necessário para obter uma solução completa (que inclui exploração pelo PROMETHEE I e PROMETHEE II), mínimo que ocorre para o caso com 100 critérios e 5 alternativas (cf. Figura 4.6 e Quadro 1 do Apêndice A). Os programas pp2 e pp3 paralelizam as restantes fases, mas o ganho resultante do paralelismo não parece compensar o acréscimo de comunicação e coordenação necessário para as dimensões de problema em causa.



Figuras 4.6 - Decomposição do tempo de execução para o programa sequencial (problema com 100 critérios x 5 alternativas)

A razão S no programa mais rápido (pp1) aumentou continuamente com o número de critérios, novamente devido ao aumento da razão computação/(comunicação + controlo). A variação desta medida (veja-se Figura 4.7), revela uma influência importante do número de critérios e uma degradação do desempenho quando cresce o número de alternativas, atribuível à comunicação dos índices de preferência multicritério (um índice por cada par ordenado de alternativas) em direcção ao nó de interface (conforme a Figura 4.4).



Figuras 4.7 - Evolução da razão S com o tamanho do problema (programa pp1)

O máximo atingido pela medida S foi de 8.54 na avaliação de 5 alternativas por 100 critérios. No entanto, para este caso o programa sequencial obtém a solução em menos de um segundo. Relativamente mais interessante é o valor de S de 7.52 para o moroso caso com 100 critérios e 500 alternativas. De facto, o uso do programa sequencial facilmente se torna desconfortável para grandes dimensões do problema, pelos tempo de espera que se tornam necessários (cf. Quadro 2 do Apêndice A). Executar um problema com 500 alternativas por apenas 5 critérios pode demorar cerca de 5 minutos. Os limites da paciência são testados quando em vez de 5 se trabalha com 100 critérios, o que pode acarretar uma espera superior a 1 hora e um quarto para avaliar 500 alternativas. Por isso, apesar de a razão S estar muito abaixo de 16 (o número de processadores utilizado), o seu significado em termos de melhoria do tempo de execução não deve ser menosprezado, dada a notória impraticabilidade de utilizar o programa sequencial para problemas de grande dimensão.

Conjunto 2) de experiências: variação do número de processadores para calcular a relação de prevalência, o PROMETHEE I e o PROMETHEE II

Neste segundo estudo trabalhou-se com 4, 8 e 16 processadores e apenas com o banco de problemas G (é razoável supor que os resultados serão qualitativamente idênticos para o banco Gs). Os algoritmos usados são escaláveis, na medida em que poderiam ter sido usadas outras dimensões da topologia de rede matricial. A única limitação imposta pelo programa é que o número de alternativas seja superior ao mínimo

entre o número de linhas e número de colunas de processadores utilizadas. Dado que os algoritmos não usam a rede de transputers de modo simétrico, estudou-se a execução dos problemas usando 8 processadores dispostos de maneira diferente: 4 linhas por 2 colunas e 2 linhas por 4 colunas.

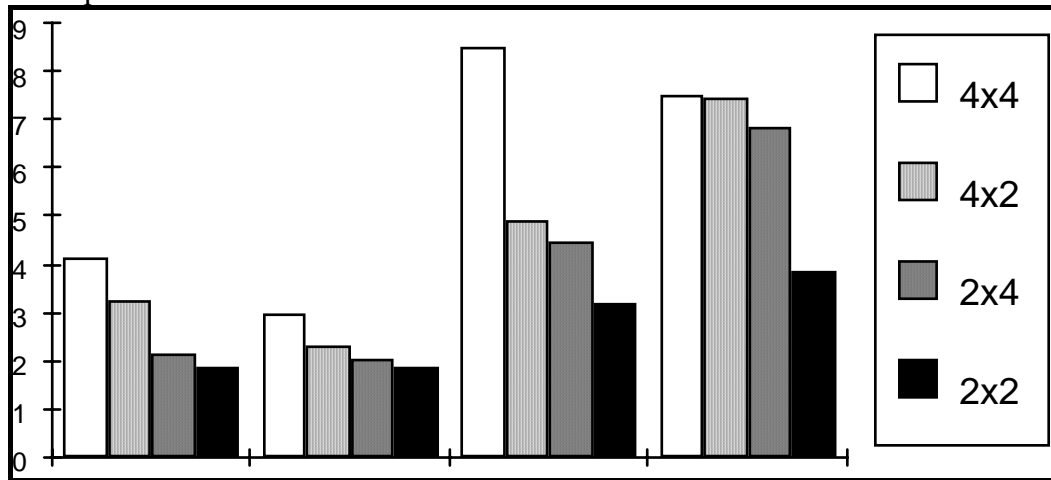


Figura 4.8 - Variação da razão S para o programa pp1 em função da rede de processadores utilizada (ver legenda) e da dimensão do problema (da esquerda para a direita: 5 critérios e 5 alternativas, 5 critérios e 500 alternativas, 100 critérios e 5 alternativas e 100 critérios e 500 alternativas).

A evolução da razão S com as dimensões do problema é análoga à verificada para as experiências usando 16 processadores, ilustrada na Figura 4.7 para o programa pp1. Novamente o programa pp1 é o mais rápido e não se encontram diferenças significativas entre o desempenho dos programas pp2 e pp3. Os resultados são apresentados no Apêndice A, nos quadros 3 a 5 (rede matricial 4 x 4) e 6 a 8 (restantes topologias) e na Figura 4.8, que se refere ao programa pp1 e a quatro problemas de dimensão extrema.

As experiências usando 8 processadores permitem concluir que a rede 4 x 2 proporciona maiores desempenhos que a rede 2 x 4. Tal resulta provavelmente do modo assimétrico como os dados são recolhidos, e que é ilustrado na Figura 4.9. Nesta figura pode observar-se que, considerando somente a ligação mais crítica, assinalada com uma seta, há maior tráfego (e, conseqüentemente, maior congestionamento) na rede 2 x 4 do que na 4 x 2.

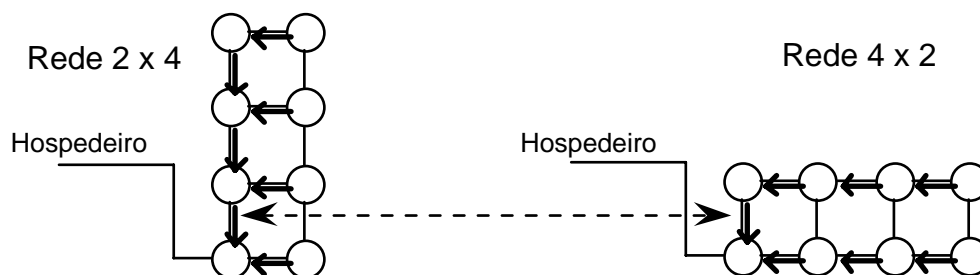


Figura 4.9 - Comparação da recolha dos resultados nas duas redes com oito nós.

Por último, assinala-se que a eficiência (definida na secção 3.3) diminui quando o número de processadores aumenta. Este comportamento é típico de programas paralelos, quando o aumento da capacidade computacional oferecida não compensa o aumento de tráfego resultante da existência de mais processadores a comunicar entre si.

Conjunto 3) de experiências: comparação de programas com diferentes funcionalidades

Das experiências anteriores é lícito concluir que, caso o agente de decisão esteja interessado em conhecer a relação de prevalência difusa, através dos índices $\pi(a,b)$, então o programa paralelo pp1 é o mais indicado. Apesar de ser o programa paralelo mais simples, é aquele que permite obter os resultados com maior rapidez em todas as situações experimentadas. No entanto, é possível que o decisor não deseje conhecer esta relação, mas somente o resultado da sua exploração pelos métodos PROMETHEE I e/ou PROMETHEE II. Pode então utilizar-se um dos programas paralelos que possuem o *handicap* de não poderem fornecer alguns dos resultados do método PROMETHEE.

Para o caso de não se necessitar conhecer a relação de prevalência foram construídas as versões limitadas pp2\R e pp3\R dos programas paralelos pp2 e pp3, respectivamente. Nas experiências efectuadas (16 processadores, banco G de problemas) os programas pp3\R e pp2\R foram os mais rápidos, seguidos do pp1 (cf. Apêndice A, Quadro 9). O valor máximo atingido pela razão S é 14.07, atingido para o problema com 500 alternativas a avaliar segundo 100 critérios. A Figura 4.10 ilustra os resultados referentes aos quatro problemas de dimensão extrema.

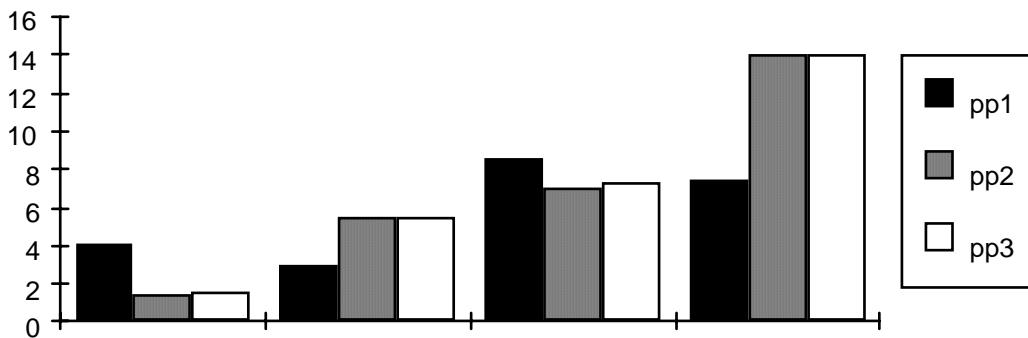
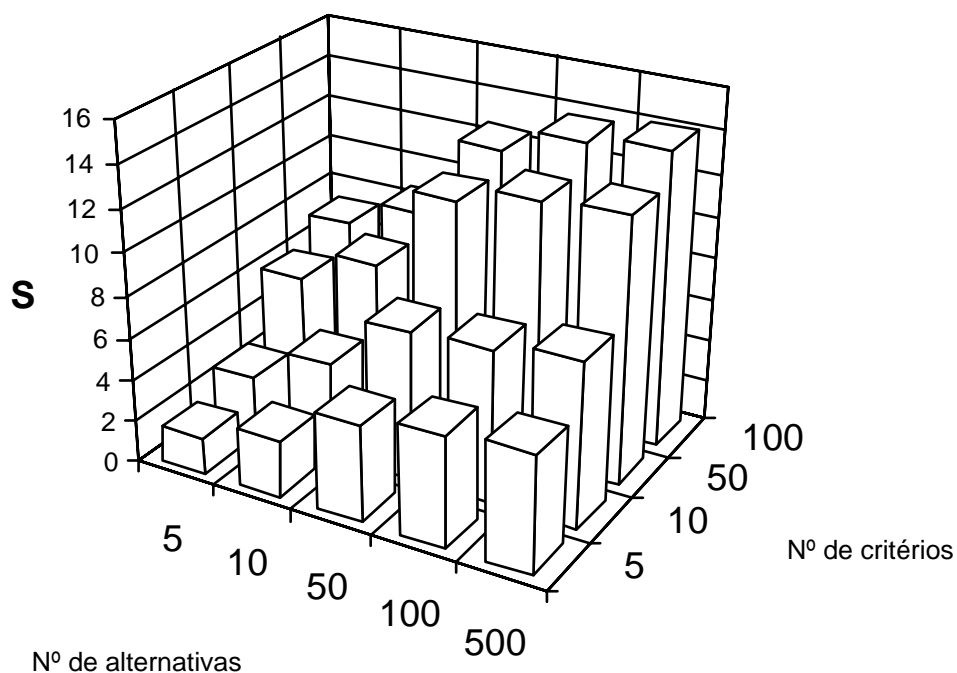


Figura 4.10 - Variação da razão S quando se abdica dos índices de preferência, em função do programa utilizado (ver legenda, na qual se omitem os sufixos) e da dimensão do problema (da esquerda para a direita: 5 critérios e 5 alternativas, 5 critérios e 500 alternativas, 100 critérios e 5 alternativas e 100 critérios e 500 alternativas).

Nos programas pp2\R e pp3\R a razão S aumenta com o número de critérios e com o número de alternativas (veja-se Figura 4.11). Este facto deve-se à ausência da necessidade de comunicar os índices de preferência $\pi(a,b)$, cujo número cresce com o quadrado do número de alternativas, tornando-se agora o aumento da computação o factor predominante no desempenho da aplicação paralela.



Figuras 4.11 - Evolução da razão S com o tamanho do problema (programa pp3\R)

Para o caso de se desejar obter apenas a pré-ordem parcial do PROMETHEE I as conclusões são idênticas, com a excepção de os programas pp2\R e pp3\R se revelarem ainda mais equivalentes (cf. Apêndice A, Quadro 10).

Por fim, considerou-se a hipótese de só se desejar conhecer a pré-ordem completa do PROMETHEE II, abdicando-se de conhecer quer os índices de preferência quer a pré-ordem parcial obténível pelo PROMETHEE I. Conforme se referiu na secção 4.5, existem situações em que é natural que o decisor apenas deseje conhecer este resultado. Torna-se então possível utilizar as versões limitadas pp2\RI e pp3\RI, respectivamente baseadas nos programas pp2 e pp3, e repetir a experiência anterior.

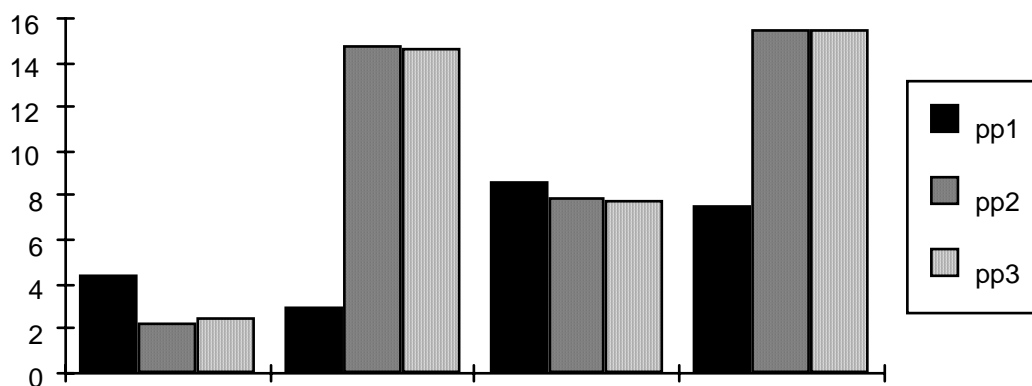


Figura 4.12 - Variação da razão S quando se abdica dos índices de preferência e da pré-ordem parcial do PROMETHEE I, em função do programa utilizado (ver legenda, em que se omite os sufixos \RI) e da dimensão do problema (da esquerda para a direita: 5 critérios e 5 alternativas, 5 critérios e 500 alternativas, 100 critérios e 5 alternativas e 100 critérios e 500 alternativas).

Os resultados da experiência, apresentados no Quadro 11 do Apêndice A, indicam que a razão S é maior do que quando só se abdica de obter os índices de preferência, ou quando se abdica de obter esses índices e a pré-ordem parcial do PROMETHEE I. Digno de ser assinalado é o facto de se atingir um valor máximo de 15,53 para a razão S, bastante próximo do número de processadores utilizado. A variação da razão S com o número de critérios e com o número de alternativas é análoga à apresentada na Figura 4.11, atingindo-se novamente o valor máximo para o caso de 500 alternativas e 100 critérios. Note-se que, no entanto, a partir de 50 ou 100 alternativas (em comparação com 500 alternativas) o crescimento é pouco notório. O programa pp2\RI é o que obtém, ainda que de modo ligeiro, os resultados mais interessantes (cf. Apêndice e Figura 4.12). A paralelização da obtenção da pré-ordem completa não se revelou eficiente.

4.8 Algumas conclusões

Os resultados obtidos mostram que diminuição dos tempos de execução dos problemas é maior quando os critérios são dos tipos que implicam maiores tempos de computação (tipos 4-6 / banco G), quando o número de critérios aumenta, quando o número de alternativas aumenta, quando o decisor abdica de conhecer alguns resultados, e quando a análise de sensibilidade afecta mais do que um critério (como no caso de alteração dos pesos de critérios).

O aproveitamento dos recursos computacionais dos 16 processadores foi relativamente baixo enquanto se pretendeu obter os índices de preferência $\pi(a,b)$. Disso resultou que paralelização conseguida obteve os melhores resultados quando o decisor sacrifica a possibilidade de conhecer esses índices. Nesse caso, a diminuição do tempo de resposta às solicitações do utilizador do programa poderá ser bastante apreciável, tornando os problemas de grande dimensão mais fáceis de estudar.

As necessidades do decisor ditam deste modo o melhor programa a utilizar. Sacrificando funcionalidade obtém-se maior aproveitamento da capacidade computacional e menores tempos de resposta às solicitações do utilizador. As implicações para o apoio à decisão são importantes, sobretudo no caso de se poder utilizar o programa pp2\RI. Considere-se o caso extremo em que se pretende avaliar 500 alternativas por 100 critérios. A obtenção de uma pré-ordem completa utilizando a rotina de cálculo normal, que é utilizada no cálculo da solução inicial, na análise de sensibilidade aos pesos dos critérios e, em geral, quando se altera mais do que um critério, demora cerca de 4644 segundos (aproximadamente 1h 17m). Caso se utilize o programa pp2\RI o mesmo resultado surge ao utilizador 15,53 vezes mais depressa, isto é, em cerca de 5 minutos. Se for feita uma alteração num parâmetro, excluindo o peso, de apenas um critério, os programas usam uma rotina de cálculo abreviado. Resultaria então para o programa sequencial um tempo

de resposta de cerca de 5 minutos, e para o programa pp2\RI um tempo 15,1 vezes menor, ou seja, cerca de 21 segundos.

Por fim, atente-se que nas iterações mais simples de análise de sensibilidade, em que se utiliza a rotina de cálculo abreviado, a razão S também atinge valores elevados. Todavia, tal só se verifica quando se pretende apenas obter a pré-ordem completa do PROMETHEE II e para os problemas de maior dimensão em termos de número de alternativas (cf. Quadro 11 do Apêndice A).

5 - PARALELIZAÇÃO DO MÉTODO ELECTRE III

5.1 Introdução

A família de métodos ELECTRE¹⁹ pertence ao conjunto dos métodos conducentes a uma relação de prevalência de síntese (ver sec. 2.5). Entre estes métodos, um dos mais conhecidos é o ELECTRE III [Roy, 1978];[Skalka et al., 1986], dedicado à problemática de ordenação das alternativas, e pioneiro na construção e exploração de uma relação de prevalência difusa.

Segundo [Vincke, 1989, p.96] e [Brans et al., 1986], o método ELECTRE III apresenta como maior obstáculo à sua utilização a necessidade de fixar parâmetros sem qualquer significado económico ou físico aparente. Por este motivo, é ainda mais necessário proceder a análises de sensibilidade para aumentar a confiança do decisor nos resultados do método, face à escolha de valores para parâmetros cujas implicações desconhece. Este facto constitui, de acordo com a argumentação apresentada no Capítulo 1, uma importante motivação para elaborar programas paralelos para executar o ELECTRE III.

Tal como os restantes métodos de apoio multicritério à decisão, o algoritmo do método ELECTRE III pode ser executado num computador sequencial vulgar, desde que o tamanho do problema seja "razoável". No capítulo anterior mostrou-se que a utilização de um computador paralelo permitiu executar mais rapidamente problemas que de outro modo seriam penosamente lentos, permitindo de algum modo redefinir em que consiste um tamanho "razoável". Neste capítulo descreve-se um estudo análogo relativamente ao método ELECTRE III, pretendendo-se averiguar o comportamento de vários programas paralelos à medida que o número de critérios e o número de alternativas aumentam. Os motivos que podem conduzir a um problema de grandes dimensões são os mesmos que se identificaram para o método PROMETHEE.

Este capítulo encontra-se estruturado do mesmo modo que o anterior, seguindo-se a esta introdução uma apresentação do método ELECTRE III e do seu algoritmo. Surge depois uma descrição dos programas elaborados e um relato das experiências realizadas para avaliar o desempenho desses programas, no ambiente computacional descrito na secção 3.6. O capítulo conclui-se com uma análise sumária dos resultados obtidos, que são discutidos à medida que vão sendo apresentados. Mais uma vez, pretendeu-se apenas estudar o comportamento dos programas desenvolvidos, sem ter em vista a aplicação dos mesmos a uma situação concreta.

¹⁹ ELimination Et Choix Traduisant la REalité.

5.2 O método ELECTRE III

O método ELECTRE III agrega os desempenhos sob diversos critérios para recomendar uma ordenação das alternativas, mais concretamente uma pré-ordem parcial, dado que admite incomparabilidade. A aplicação do ELECTRE III divide-se em duas fases, como é habitual nos métodos de prevalência. Numa primeira fase comparam-se todos os pares de alternativas, considerando o desempenho nos múltiplos critérios, de modo a obter uma relação de prevalência difusa (ou seja, do tipo 2). Posteriormente, procede-se à exploração da relação obtida, com o fim de extrair uma pré-ordem parcial das alternativas.

Considere-se que, concluída uma fase de estruturação da situação de decisão, se definiu o conjunto A de m alternativas a avaliar e uma família F de n critérios $g_1(\cdot), \dots, g_n(\cdot)$. Represente-se por $g_j(a)$ o desempenho da alternativa a segundo o critério j . Estes desempenhos são os elementos de uma matriz de decisão, que se supõe definida. Como exemplo, considere-se a seguinte matriz de decisão ($m=4$ e $n=3$), idêntica à utilizada no capítulo anterior:

| | crit1 | crit2 | crit3 |
|------|-------|-------|-------|
| alt1 | 2 | 79 | 7 |
| alt2 | 3 | 99 | 1 |
| alt3 | 2 | 96 | 4 |
| alt4 | 3 | 90 | 6 |

A primeira fase do método ELECTRE III permite a modelação dos diversos pontos de vista através de pseudo-critérios (ver sec. 2.6). Deste modo, perante um critério g_j e duas alternativas a_1 e a_2 , as situações de preferência possíveis são as apresentadas na Figura 5.1. A introdução de pseudo-critérios, que é opcional, implica a fixação para cada critério²⁰ g_j de um limiar de indiferença (q_j) e de um limiar de preferência estrita (p_j). Para além disso, é possível especificar para cada critério g_j um limiar de veto (v_j), cujo papel se descreverá mais adiante.

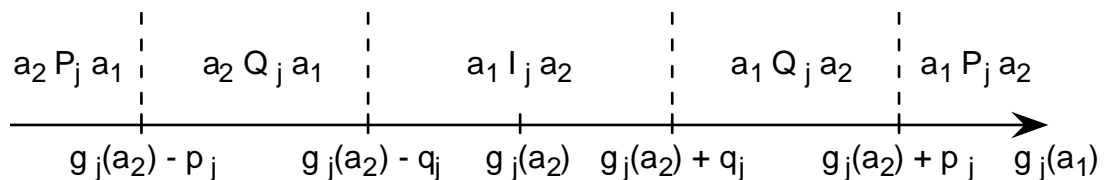


Figura 5.1 - Situações de preferência em pseudo-critérios: P indica preferência estrita, Q indica preferência fraca, I indica indiferença.

²⁰ Por comodidade utilizar-se-á doravante apenas o termo "critério", apesar de se considerar que se estão a utilizar pseudo-critérios.

Segundo Roy, o objectivo desta primeira fase é o estabelecimento de uma relação de prevalência difusa, definida através do cálculo, para cada par ordenado de alternativas $(a_1, a_2) \in A \times A$, de um número que expresse a credibilidade da hipótese de prevalência "a alternativa a_1 prevalece sobre a alternativa a_2 ", ou seja, "a alternativa a_1 é pelo menos tão boa como a alternativa a_2 ". Para o cálculo desse número consideram-se os critérios concordantes e os critérios discordantes com esta afirmação. Os primeiros constituem a chamada "coligação concordante" e segundos formam a "coligação discordante".

A coligação concordante é formada pelos critérios g_j que verifiquem (ainda em relação a a_1 e a_2) uma das seguintes condições: $a_1 P_j a_2$, $a_1 Q_j a_2$, $a_1 I_j a_2$, ou $a_2 Q_j a_1$. A verificação de cada uma destas condições concorda mais, ou menos, fortemente com a hipótese de prevalência, pelo que se define para cada par ordenado $(a_1, a_2) \in A \times A$ e para cada critério um número que expressa de modo difuso a concordância do critério. Este índice atinge o valor máximo (a unidade) quando $a_1 I_j a_2$ ou $a_1 P_j a_2$, atinge o valor zero se o critério não fizer parte da coligação concordante (i.e. $a_2 P_j a_1$), e varia de modo linear entre estas situações. A forma analítica do índice de concordância é a seguinte (ver gráfico correspondente na Figura 5.2):

Índice de concordância do critério g_j com a hipótese " a_1 prevalece sobre a_2 "

$$c_j(a_1, a_2) = \begin{cases} 1 & , \text{se } g_j(a_1) \geq g_j(a_2) - q_j \\ 0 & , \text{se } g_j(a_1) < g_j(a_2) - p_j \\ \frac{g_j(a_1) + p_j - g_j(a_2)}{p_j - q_j} & , \text{ caso contrário} \end{cases}$$

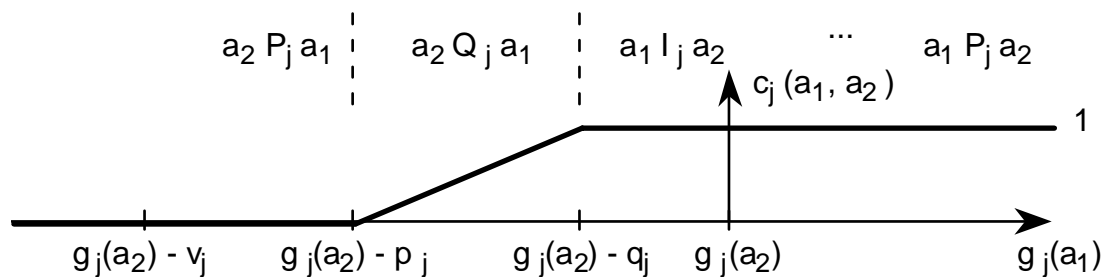


Figura 5.2 - Índice de concordância para um critério

A força da coligação concordante na validação da hipótese de prevalência é uma média ponderada dos índices de concordância monocritério, na qual intervêm coeficientes de ponderação ("pesos") que estabelecem a importância relativa de cada critério. Note-se, porém, que estes "pesos" pouco têm a ver com os referidos na secção 2.4.

Designando por k_j a importância relativa de cada critério g_j , define-se o *índice de concordância multicritério* do seguinte modo:

$$c(a_1, a_2) = \sum_{j=1}^n k_j \times c_j(a_1, a_2)$$

A coligação discordante é formada pelos critérios g_j que não pertencem à coligação concordante, ou seja, aqueles para os quais se verifique $a_2 P_j a_1$. Os critérios que verificam esta condição podem discordar mais, ou menos, fortemente com a hipótese de prevalência, pelo que se define, para cada par ordenado $(a_1, a_2) \in A \times A$ e para cada critério, um número que expressa de modo difuso a discordância do critério. Este índice atinge o valor máximo (a unidade) quando a diferença de desempenhos entre a_1 e a_2 ultrapassa um limiar de veto v_j , atinge o valor zero se o critério fizer parte da coligação concordante, e varia de modo linear entre estas situações extremas. A forma analítica do índice de discordância é a seguinte (ver gráfico correspondente na Figura 5.3):

Índice de discordância do critério g_j com a hipótese " a_1 prevalece sobre a_2 "

$$d_j(a_1, a_2) = \begin{cases} 0 & , \text{se } g_j(a_2) - g_j(a_1) \leq p_j \\ \frac{g_j(a_2) - g_j(a_1) - p_j}{v_j - p_j} & , \text{se } p_j < g_j(a_2) - g_j(a_1) \leq v_j \\ 1 & , \text{se } g_j(a_2) - g_j(a_1) > v_j \end{cases}$$

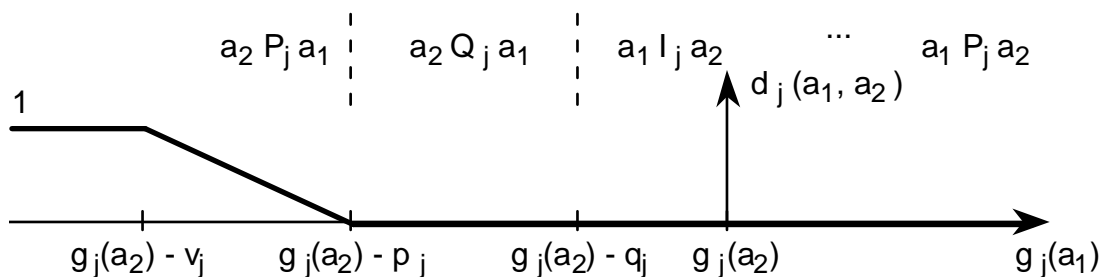


Figura 5.3 - Índice de discordância para um critério

O nome de limiar de veto atribuído a v_j deve-se ao facto de um critério poder vetar a hipótese de prevalência quando a diferença de desempenhos entre duas alternativas ultrapassar esse limiar. Esta possibilidade de veto, os diversos índices de discordância (monocritério) e o índice de concordância multicritério, são combinados de forma a obter um índice de credibilidade para a hipótese de prevalência. Entre muitas expressões que poderiam ter sido propostas, B. Roy escolheu a apresentada a seguir, que varia entre zero e um, e possui a propriedade de só considerar a discordância de cada critério quando esta é superior à concordância multicritério. O índice de credibilidade da afirmação " a_1 prevalece sobre a_2 " (hipótese de prevalência) é, segundo B. Roy, calculado da seguinte forma:

$$\sigma(a_1, a_2) = c(a_1, a_2) \prod_{\substack{j \in \{1, \dots, n\}: \\ d_j(a_1, a_2) > c(a_1, a_2)}} \frac{1 - d_j(a_1, a_2)}{1 - c(a_1, a_2)}$$

O conjunto destes índices de credibilidade define a relação de prevalência difusa que a primeira fase do método pretende obter. Note-se que a credibilidade é nula sempre que a condição de veto se verifique para pelo menos um critério.

Para o caso do exemplo para o qual se apresentou a matriz de decisão, suponha-se que os pontos de vista tinham sido modelados através de três pseudo-critérios definidos como segue, com limiares fixos:

| Critério | Direcção | Peso | L.indiferença | L.preferência | L.veto |
|----------|-----------|------|---------------|---------------|--------|
| crit 1 | maximizar | 0,3 | 0,3 | 0,9 | 2 |
| crit 2 | maximizar | 0,3 | 5 | 10 | 20 |
| crit 3 | maximizar | 0,4 | 1 | 3 | 10 |

O cálculo dos índices de credibilidade resultaria então na relação de prevalência difusa ilustrada na Figura 5.4.

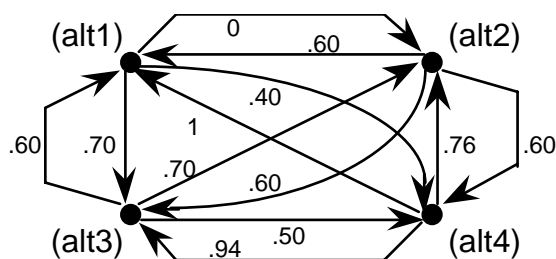


Figura 5.4 - Exemplo da relação de prevalência definida pelos índices de credibilidade

A segunda fase do método ELECTRE III visa obter uma pré-ordem parcial através da exploração da relação de prevalência difusa resultante da primeira fase. No método ELECTRE III a exploração consiste em obter duas pré-ordens completas, possivelmente distintas, chamadas destilação descendente e destilação ascendente, que serão combinadas posteriormente numa pré-ordem parcial.

Em cada destilação a relação de prevalência difusa é transformada em relações de prevalência não difusas (do tipo 1), através de níveis de corte apropriados. Por exemplo, aplicando um nível de corte de 0,75 à relação ilustrada na Figura 5.4 resultaria a relação não difusa ilustrada na Figura 5.5.

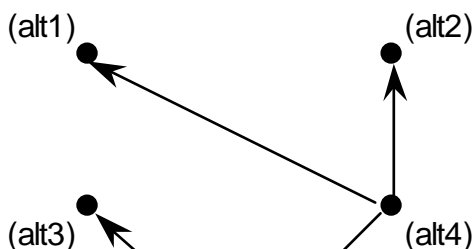


Figura 5.5 - Exemplo da relação de prevalência definida por um nível de corte

Para cada relação de tipo 1 definida por um nível de corte, cada destilação distingue as alternativas pela sua “qualificação”, entendendo-se por qualificação de uma alternativa a diferença entre o número de alternativas sobre as quais a alternativa considerada prevalece e o número de alternativas que prevalecem sobre a alternativa considerada. O método pressupõe que a qualificação de cada alternativa é uma indicação da sua desejabilidade pelo decisor.

A destilação descendente escolhe, em cada iteração, a alternativa ou as alternativas com maior qualificação, de modo a ordenar as alternativas por uma ordem decrescente de desejabilidade. A destilação ascendente processa-se de modo contrário, escolhendo a alternativa ou as alternativas com menor qualificação, de modo a ordenar as alternativas por uma ordem crescente de desejabilidade. Ambas as destilações tentam minorar a existência de *ex-aequo* entre alternativas, recorrendo a relações de prevalência não difusas sucessivamente mais fracas, resultantes da aplicação de níveis de corte cada vez menores.

O algoritmo das destilações faz intervir uma função de discriminação $s(\cdot)$, função a definir pelo utilizador, que serve dois propósitos distintos. O primeiro é determinar de quanto se deve baixar o limiar de corte quando o actual não é suficientemente discriminatório. O segundo consiste em empobrecer a relação de prevalência não difusa obtida pelo limiar de corte, estabelecendo um limiar que tem de ser ultrapassado pela diferença entre $\sigma(a,b)$ e $\sigma(b,a)$ (com $a, b \in A$) para que se possa afirmar que a prevalece sobre b nessa relação. O processo de destilação é complexo, apresentando-se por isso um algoritmo para melhor o descrever:

Destilação ascendente (descendente)

Seja $B \diamond A$ o conjunto de alternativas a ordenar

Enquanto B não estiver vazio faz

$\lambda \diamond$ máximo índice de credibilidade $\sigma(a,b)$, considerando todos os pares $(a,b) \in B \times B$ **[subproblema 1]**

$D \diamond B$

Enquanto $\lambda \neq 0$ e o número de alternativas em D for diferente de 1 faz

$\lambda \diamond$ máximo $\sigma(a,b)$, considerando todos os pares $(a,b) \in D$ tais que $\sigma(a,b) < \lambda - s(\lambda)$ **[subproblema 2]**

/* A qualificação $q(a)$ de cada alternativa $a \in D$,

baseada na relação de prevalência não difusa

$a S^\lambda b \iff \sigma(a,b) > \lambda$ e $\sigma(a,b) > \sigma(b,a) + s[\sigma(a,b)]$, é: /*

$q(a) \diamond \#\{b \in D: a S^\lambda b\} - \#\{b \in D: b S^\lambda a\}$

[subproblema 3]

$q^* \diamond$ máximo (mínimo) $q(a)$, para $a \in D$ **[subproblema 4]**

Retém em D todas as alternativas que possuam qualificação q^*

fim_faz

Classifica a(s) alternativa(s) pertencente(s) a D na classe presente

Seja $B \diamond B - D$ o conjunto das alternativas ainda por ordenar

/* reitera ciclo para encontrar as alternativas a classificar na classe seguinte */

fim_faz

No fim dos procedimentos de destilação, as duas pré-ordens completas resultantes podem ser agregadas numa pré-ordem parcial (P, I, R), como segue:

- a é preferível a b (a P b) sse a está melhor posicionado que b segundo uma das destilações e não está pior posicionado segundo a outra destilação;
- a é indiferente a b (a I b) sse a e b estiverem posicionados ex-aequo em ambas as destilações;
- a e b são incomparáveis (a R b) sse a estiver melhor posicionada que b numa das destilações e o contrário acontecer na outra destilação.

Prosseguindo a aplicação do método ELECTRE III ao exemplo que tem sido apresentado obter-se-ia os seguintes resultados das destilações:

| Posição | Descendente | Ascendente |
|----------|-------------|--------------|
| 1 | alt4 | alt2 e alt4 |
| 2 | alt2 | |
| 3 | alt1 e alt3 | alt 1 e alt3 |
| 4 (pior) | | |

Estas destilações poderiam então ser combinadas na pré-ordem parcial ilustrada na Figura 5.6, onde as setas de um sentido indicam preferência, as setas de dois sentidos indicam indiferença e a ausência de setas indica incomparabilidade. Esta pré-ordem pode ser apresentada de um modo mais simplificado, em que surgiriam apenas relações transitivas. Contudo, os programas construídos no âmbito deste trabalho não o fazem, dado que em [Skalka et al., 1986], a referência seguida, não se especifica um algoritmo para o fazer.

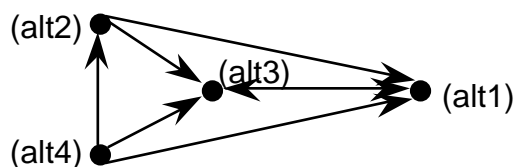


Figura 5.6 - Exemplo da pré-ordem parcial obtida por combinação das destilações

Conclui-se esta secção com um esquema do método ELECTRE III (Figura 5.7), adaptado de [Skalka et al., 1986].

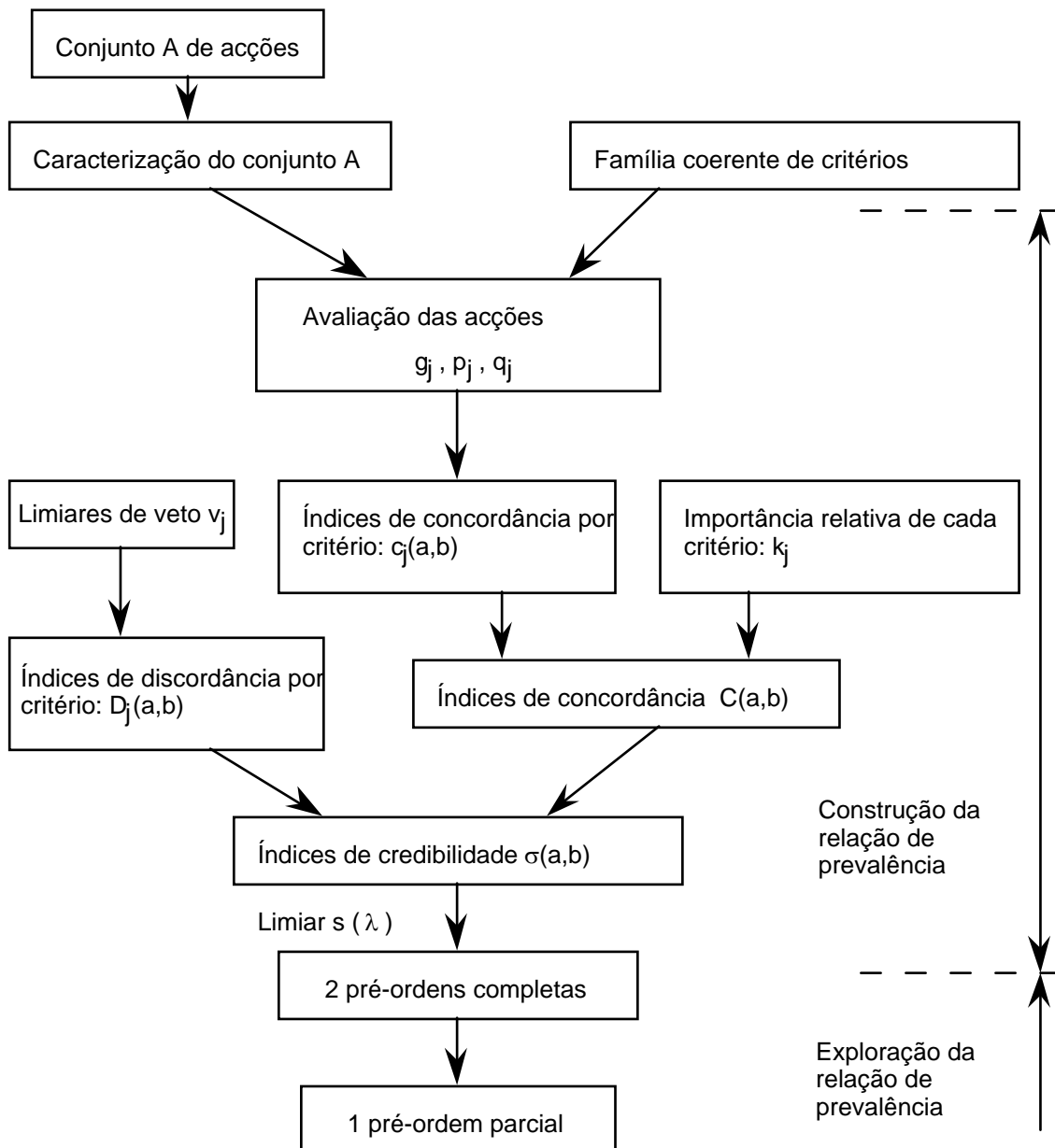


Figura 5.7 - O método ELECTRE III

5.3 Um algoritmo para a execução do método ELECTRE III

Tal como se fez para o método PROMETHEE, apresenta-se um algoritmo para a execução do método ELECTRE III, baseado no que foi descrito na secção anterior. Utiliza-se uma linguagem algorítmica, baseada na língua portuguesa, para a especificação do algoritmo.

Dados de entrada

- Número de critérios n (número inteiro positivo);
- Número de alternativas m (número inteiro positivo);

Matriz de desempenho D (m x n), cujos elementos d_{ij} (números reais) indicam o desempenho da alternativa i segundo o critério que operacionaliza o ponto de vista j, ou seja, $d_{ij} = g_j(i)$;

Caracterização dos pseudo-critérios C (dado estruturado: contém o seu peso, os limiares de indiferença, preferência e veto, e o modo como os limiares são utilizados);

Caracterização da função de discriminação $s(\cdot)$ (dado estruturado: contém a definição de uma função)

Algoritmo simplificado

1. Cálculo dos índices de credibilidade para cada par ordenado de alternativas

Para cada par ordenado de alternativas (a,b), utiliza a informação contida em C e D para calcular os índices de concordância monocritério, os índices de concordância multicritério, os índices de discordância, e calcula $\sigma(a,b)$.

2. Obtenção da destilação descendente (obtem pré-ordem completa (P,I)_d)

Utiliza índices de credibilidade $\sigma(a,b)$ e informação sobre o limiar $s(\cdot)$ para ordenar as alternativas segundo uma destilação descendente.

3. Obtenção da destilação ascendente (obtem pré-ordem completa (P,I)_a)

Utiliza índices de credibilidade $\sigma(a,b)$ e informação sobre o limiar $s(\cdot)$ para ordenar as alternativas segundo uma destilação ascendente.

4. Combinação das duas destilações (obtem pré-ordem parcial (P,I,R))

Utiliza os resultados das destilações para construir uma pré-ordem parcial.

Algoritmo detalhado

1. Cálculo dos índices de credibilidade para cada par ordenado de alternativas

Para cada alternativa a (a=2,...,m) faz

Para cada alternativa b menor que a (b=1,...,a-1) faz

$c(a,b) \diamond 0$ e $c(b,a) \diamond 0$

Para todos os critérios (j=1,...,n) faz

Usa C e D para calcular $t \diamond k_j c_j(a,b)$

$c(a,b) \diamond c(a,b) + t$

Usa C e D para calcular $t \diamond k_j c_j(b,a)$

$c(b,a) \diamond c(b,a) + t$

Usa C e D para calcular $d_j(a,b)$

Usa C e D para calcular $d_j(b,a)$

fim_faz

Usa $c(a,b)$ e $d_j(a,b)$ (j=1,...,n) para calcular $\sigma(a,b)$

Usa $c(b,a)$ e $d_j(b,a)$ (j=1,...,n) para calcular $\sigma(b,a)$

fim_faz

fim_faz

2. Obtenção da destilação descendente (obtem pré-ordem completa $(P,I)_d$)

classe \diamond 1 /* melhor posição*/

B \diamond A /* Conjunto das alternativas */

Enquanto B não estiver vazio faz

λ \diamond máximo índice de credibilidade $\sigma(a,b)$, considerando todos os pares
 $(a,b) \in B \times B$ [**subproblema 1**]

D \diamond B

Enquanto $\lambda \neq 0$ e o número de alternativas em D for diferente de 1 faz

λ \diamond máximo $\sigma(a,b)$, considerando todos os pares $(a,b) \in D$ tais
que $\sigma(a,b) < \lambda - s(\lambda)$ [**subproblema 2**]

$q(a) \diamond \#\{b \in D: a S^\lambda b\} - \#\{b \in D: b S^\lambda a\}$

[**subproblema 3**]

$q^* \diamond$ máximo $q(a)$, para $a \in D$ [**subproblema 4**]

Retém em D apenas as alternativas que possuam qualificação q^*

fim_faz

Para todas as alternativas $a \in D$ faz

pos_des(a) \diamond classe /* posição de a na destilação descendente */

fim_faz

Seja B \diamond B - D o conjunto das alternativas ainda por ordenar

classe \diamond classe + 1;

fim_faz

2. Obtenção da destilação ascendente (obtem pré-ordem completa $(P,I)_a$)

classe \diamond PIOR /* pior posição*/

B \diamond A /* Conjunto das alternativas */

Enquanto B não estiver vazio faz

λ \diamond máximo índice de credibilidade $\sigma(a,b)$, considerando todos os pares
 $(a,b) \in B \times B$ [**subproblema 1**]

D \diamond B

Enquanto $\lambda \neq 0$ e o número de alternativas em D for diferente de 1 faz

λ \diamond máximo $\sigma(a,b)$, considerando todos os pares $(a,b) \in D$ tais
que $\sigma(a,b) < \lambda - s(\lambda)$ [**subproblema 2**]

$q(a) \diamond \#\{b \in D: a S^\lambda b\} - \#\{b \in D: b S^\lambda a\}$

[**subproblema 3**]

$q^* \diamond$ mínimo $q(a)$, para $a \in D$ [**subproblema 4**]

Retém em D apenas as alternativas que possuam qualificação q^*

fim_faz

Para todas as alternativas $a \in D$ faz

pos_asc(a) \diamond classe /* posição de a na destilação ascendente */

fim_faz

Seja $B \diamond B - D$ o conjunto das alternativas ainda por ordenar
classe \diamond classe - 1;

fim_faz

4. Combinação das duas destilações (obtem pré-ordem parcial (P,I,R))

Para cada alternativa a ($a=1,\dots,m$) faz

Para cada alternativa b diferente de a ($b \diamond 1,\dots,m ; b \neq a$) faz

Se $[\text{pos_des}(a) < \text{pos_des}(b) \text{ e } \text{pos_asc}(a) < \text{pos_asc}(b)]$ ou
 $[\text{pos_des}(a) < \text{pos_des}(b) \text{ e } \text{pos_asc}(a) = \text{pos_asc}(b)]$ ou
 $[\text{pos_des}(a) = \text{pos_des}(b) \text{ e } \text{pos_asc}(a) < \text{pos_asc}(b)]$

então resulta a P b

Se_não,

Se $[\text{pos_des}(a) = \text{pos_des}(b) \text{ e } \text{pos_asc}(a) = \text{pos_asc}(b)]$
então resulta a I b

Se_não resulta a R b

fim_se

fim_faz

fim_faz.

5.4 Possibilidades de paralelização no método ELECTRE III

No início de um processo de paralelização de uma aplicação deve-se imaginar um algoritmo que contenha tarefas que possam ser executadas de modo independente. Os ganhos de desempenho que se podem esperar serão tanto maiores quanto melhor for o compromisso entre a quantidade de computação (pretende-se maximizar a ocupação útil dos processadores) e a quantidade de comunicação (que se pretende minimizar).

Tal como se concluiu para o método PROMETHEE, a obtenção da relação de prevalência difusa (passo 1 do algoritmo apresentado na secção anterior) pode ser efectuada em modo paralelo. De facto, o cálculo de todos os índices, culminando no cálculo do índice de credibilidade para cada par ordenado, é independente dos cálculos a efectuar para os restantes pares ordenados. Estes cálculos podem portanto ocorrer em simultâneo.

No que respeita à exploração da relação de prevalência obtida, as possibilidades parecem ser menos atraentes. Por um lado, as destilações ascendente (passo 2 do algoritmo) e descendente (passo 3) podem ser efectuadas simultaneamente. No entanto, esta possibilidade é limitada quando se possui mais que dois processadores. É por isso necessário tentar encontrar paralelismo *em* cada uma das destilações, visto que o paralelismo existente *entre* as destilações é insuficiente para ser explorado no multicomputador descrito na secção 3.6.

O estudo do algoritmo de cada destilação revela que este não parece ser paralelizável de modo eficiente. De facto, a quase totalidade da computação necessária reside nos ciclos assinalados como "subproblemas" (1 a 4), cuja execução não pode ser iniciada sem se conhecer o resultado do subproblema anterior. Ora isto impõe uma sequência na execução dessas instruções, ou seja, um algoritmo sequencial. A este carácter sequencial do algoritmo de destilação acresce a sua forte dependência dos dados utilizados, o que reveste o seu comportamento e a sua duração de alguma imprevisibilidade.

No entanto, os subproblemas 1 a 4 são paralelizáveis individualmente, o que leva a identificar uma estratégia em que o algoritmo é controlado por um único processador "pivot" e os subproblemas são resolvidos utilizando os restantes processadores. É, no entanto, uma estratégia menos promissora do que a referente à obtenção dos índices de credibilidade. Uma resolução para este óbice passaria provavelmente por uma redefinição do algoritmo de destilação, mantendo a sua filosofia, mas introduzindo alterações a um nível operacional. Contudo, tal consistiria numa alteração ao algoritmo de B. Roy, pelo que não se prosseguiu essa via.

Por fim, pode identificar-se a possibilidade de, uma vez conhecidas as pré-ordens completas resultantes das destilações, obter a pré-ordem parcial (P,I,R): o passo 4 do algoritmo pode ser paralelizado, na medida em que existe uma vez mais independência no estabelecimento de relações binárias entre pares distintos de alternativas.

5.5 Programas construídos

Nesta secção apresentam-se vários programas construídos para avaliar as potencialidades das possibilidades de paralelização identificadas na secção anterior. Apresenta-se em primeiro lugar um programa sequencial para execução do método ELECTRE III, que serviu para validação de resultados e para referência de tempos de execução no cálculo de medidas de desempenho. Na descrição deste programa inclui-se o modo como o utilizador pode interagir com o programa, que é comum aos restantes programas, que são paralelos. Novamente, omite-se a apresentação de algoritmos detalhados ou do código do programa dada a extensão adicional que tal implicaria.

Programa sequencial (nome abreviado "seq.")

O programa construído segue o algoritmo detalhado apresentado na secção 5.3. No entanto, o programa possui algumas funcionalidades não explicitadas no algoritmo. Por exemplo, o programa suporta análises de sensibilidade à caracterização e peso dos critérios, e também à caracterização do limiar $s(.)$. Para além disso, o programa permite potencialidades de modelação ao nível do exposto em [Skalka et al., 1986]. A esse respeito, destaca-se a modelação dos limiares de indiferença e preferência como funções

do tipo $Y = \alpha + \beta X$, às quais está associado um sentido: um sentido directo (inverso) indica que se pretende considerar o limiar como uma função do pior (melhor) desempenho. Suponha-se, por exemplo, que se está a calcular o índice de concordância para um par de alternativas (a,b) e um critério g_j tais que $g_j(b) > g_j(a)$. Segundo a expressão apresentada na secção 5.2, o índice será igual a zero se $g_j(b) - g_j(a) > p_j$ (considera-se que quanto maior for o desempenho mais preferível será a alternativa). De acordo com Skalka et al., substitui-se essa condição, que pressupõe um limiar p_j fixo, por $g_j(b) - g_j(a) > p_j(\cdot)$, com $p_j(\cdot) = \alpha + \beta g_j(a)$, se o sentido for directo, ou $p_j(\cdot) = \alpha + \beta g_j(b)$ se o sentido for inverso. No que respeita ao limiar de veto admitem-se quatro formas de modelação, tal como indicado por Skalka et al. Apresentam-se de seguida as expressões destas quatro formas, supondo que ou $g_j(a)$ é o pior desempenho e o sentido é directo, ou $g_j(a)$ é o melhor desempenho e o sentido é inverso:

- Tipo 0 : $v_j[g_j(a)] = \alpha + \beta g_j(a)$;
- Tipo 1 : $v_j[g_j(a)] = p_j[g_j(a)] + \alpha / k_j$;
- Tipo 2 : $v_j[g_j(a)] = p_j[g_j(a)] + \alpha g_j(a) / k_j$;
- Tipo 3 : $v_j[g_j(a)] = p_j[g_j(a)] + \alpha g_j(a) / k_j + \beta / k_j$.

É portanto necessário que se especifiquem para cada critério o sentido, os parâmetros para o limiar de indiferença (α_I e β_I), para o limiar de preferência (α_P e β_P) e para o limiar de veto (tipo, α_V e, eventualmente, β_V). Permite-se ainda a escolha entre “maximização” e “minimização” dos desempenhos segundo cada critério.

Todos estes dados são introduzidos num ficheiro de texto como o seguinte, correspondente ao exemplo apresentado na secção 5.2, previamente à execução do programa, sem prejuízo de poderem ser alterados durante a execução da aplicação, na fase de análise de sensibilidade.

| PROBLEMA exemplo | | | | | | | | | | |
|------------------|------|---------|---------|-------------|-------|-------------|------|------|------|------|
| CRITÉRIOS | | | | indiferença | | preferência | | veto | | |
| n=3,p=1 | peso | max/min | dir/inv | alfa | beta | alfa | beta | tipo | alfa | beta |
| crit1 | 3 | MAX | DIRECTO | 0.3 | 0 | 0 | 0 | 2 | 0 | |
| crit2 | 3 | MAX | DIRECTO | 5 | 0 | 10 | 0 | 0 | 20 | 0 |
| crit3 | 4 | MAX | INVERSO | 1 | 0 | 3 | 0 | 0 | 10 | 0 |
| ALTERNATIVAS | | | | Crit1 | Crit2 | Crit3 | | | | |
| m=4 | | | | | | | | | | |
| alt1 | | 2 | | 79 | | 7 | | | | |
| alt2 | | 3 | | 99 | | 1 | | | | |
| alt3 | | 2 | | 96 | | 4 | | | | |
| alt4 | | 3 | | 90 | | 6 | | | | |

É ainda necessário especificar a forma do limiar $s(\cdot)$, utilizado pelos algoritmos de destilação. Seguiu-se novamente Skalka et al., em que se coloca $s(\lambda) = \alpha + \beta \lambda$. Não é necessário especificar um valor para os parâmetros, assumindo-se por omissão os valores recomendados de $\alpha = 0,3$ e $\beta = -0,15$. O utilizador poderá modificar estes parâmetros no decurso da execução do programa.

Quando o programa é executado, através da escrita do seu nome na linha de comando do sistema operativo do computador hospedeiro, surgem ao utilizador as seguintes opções:

- "Ler problema", permite ler para as estruturas de dados do programa o conteúdo de um ficheiro de texto análogo ao representado acima;
- "Mostrar problema corrente", permite visualizar os dados relativos a uma situação de decisão contidos nas estruturas de dados do programa;
- "Resolver problema corrente", permite executar o método ELECTRE III utilizando os dados da situação de decisão corrente e, posteriormente, efectuar uma análise de sensibilidade à caracterização dos critérios (peso, limiares) e à caracterização do limiar $s(\cdot)$; a análise de sensibilidade pode afectar vários destes parâmetros de cada vez e envolver um ou mais critérios; no caso de se alterar apenas a forma de $s(\cdot)$ então só a fase de exploração é repetida, uma vez que a relação de prevalência permanece inalterada;
- "Alterar parâmetros definidos por omissão", permite alterar os parâmetros α e β definidos por omissão para definir o limiar $s(\cdot)$;
- "Guardar problema corrente", permite guardar a situação corrente, considerando as alterações introduzidas durante as análises de sensibilidade, excepto as referentes ao limiar $s(\cdot)$, num ficheiro de texto compatível com o de entrada de dados e com um nome a escolher pelo utilizador;
- "Sair do programa".

Os diversos programas paralelos são, no que respeita à interacção com o utilizador, em tudo semelhantes ao programa sequencial. A única diferença observável será a da velocidade de execução, que varia de programa para programa.

Primeiro programa paralelo (nome abreviado "pp1")

O primeiro programa paralelo (pp1) foi construído a partir do programa sequencial, paralelizando apenas a primeira fase do método ELECTRE III, ou seja, a fase de obtenção dos índices de credibilidade (ponto 1 do algoritmo apresentado em 5.3).

Tal como sucedia nos programas paralelos construídos para o método PROMETHEE, o programa utiliza a topologia física da rede de transputers, através da qual envia os dados e recolhe os resultados. Em cada transputer é executado um processo idêntico, que contém instruções que são condicionalmente executadas dependendo da posição do nó na rede. O processo que estiver a ser executado no nó de interface com o hospedeiro interage directamente com o utilizador e lê o ficheiro de texto, cujos dados guarda e retransmite aos seus vizinhos. Os restantes processos, no seu início, limitam-se a receber e retransmitir os dados, de acordo com o padrão ilustrado na Figura 5.8 (que é uma repetição da Figura 4.3).

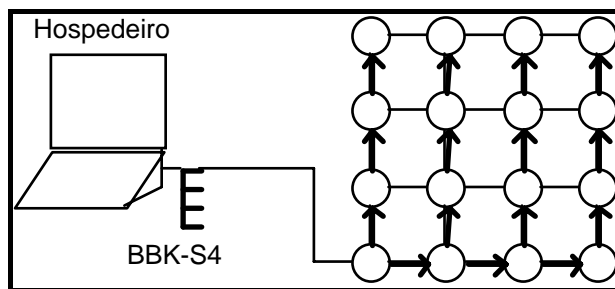


Figura 5.8 - Difusão dos dados do nó interface para os restantes

A fase de obtenção da relação de prevalência de síntese consiste em calcular o índice de credibilidade $\sigma(a,b)$ para todo o par ordenado de alternativas $(a,b) \in A \times A$, em que $A \times A$ representa o conjunto $\{(a,b): a \in A, b \in A\}$. A paralelização desta fase consistiu no cálculo simultâneo de vários pares ordenados, atribuindo-se um subconjunto de $A \times A$ a cada processador. Tenta-se assim distribuir o trabalho computacional requerido nesta fase equitativamente por todos os processadores disponíveis na rede, através de uma partição do conjunto $A \times A$ em p partes (assumindo que estão disponíveis p processadores) P_1, P_2, \dots, P_p .

Considerando a partição A_1, A_2, \dots, A_p do conjunto A de alternativas tal que $A_1 \approx A_2 \approx \dots \approx A_p = A$, com $A_i \leftrightarrow A_j = \{\}$ e $0 \leq \#A_i - \#A_j \leq 1$ ($1 \leq i < j \leq p$), então $P_x = \{(a,b): a \in A_x, b \in A\}$. Os conjuntos P_1, P_2, \dots, P_p constituirão, assim definidos, uma partição de $A \times A$, que se ilustra de seguida para um caso com 30 alternativas e uma topologia de rede matricial de dimensão 4×4 (note-se que a equidade não será absoluta, dado que 30 não é múltiplo do número de processadores):

| | | | |
|--|--|--|--|
| PROCESSADOR 12 a: 25 e 26 x b: 1 a 30 | PROCESSADOR 13 a: 27 e 28 x b: 1 a 30 | PROCESSADOR 14 a: 29 x b: 1 a 30 | PROCESSADOR 15 a: 30 x b: 1 a 30 |
| PROCESSADOR 8 a: 17 e 18 x b: 1 a 30 | PROCESSADOR 9 a: 19 e 20 x b: 1 a 30 | PROCESSADOR 10 a: 21 e 22 x b: 1 a 30 | PROCESSADOR 11 a: 23 e 24 x b: 1 a 30 |
| PROCESSADOR 4 a: 9 e 10 x b: 1 a 30 | PROCESSADOR 5 a: 11 e 12 x b: 1 a 30 | PROCESSADOR 6 a: 13 e 14 x b: 1 a 30 | PROCESSADOR 7 a: 15 e 16 x b: 1 a 30 |
| PROCESSADOR 0* a: 1 e 2 x b: 1 a 30 | PROCESSADOR 1 a: 3 e 4 x b: 1 a 30 | PROCESSADOR 2 a: 5 e 6 x b: 1 a 30 | PROCESSADOR 3 a: 7 e 8 x b: 1 a 30 |

* Processador de interface com o hospedeiro

Este modo de definir a partição é distinto do utilizado para o método PROMETHEE, tendo as escolhas sido influenciadas pela paralelização da fase de exploração da relação de prevalência, efectuada nos programas paralelos posteriores. De qualquer forma, não se crê que houvesse diferenças significativas no desempenho da

paralelização da primeira fase caso se tivesse efectuado a partição que foi definida para os programas PROMETHEE. O código (em linguagem C) utilizado para definir a partição é:

```
lsup = (nalt/gridsize) * (MyProcID+1);
lsup += (MyProcID+1 > nalt%gridsize) ? nalt%gridsize : MyProcID+1;
linf = (nalt/gridsize) * MyProcID+1;
linf += (MyProcID > nalt%gridsize) ? nalt%gridsize : MyProcID;
```

Em que:

- a rede matricial tem `gridsize` processadores;
- cada processador possui um número `MyProcID` entre 0 e `gridsize - 1`;
- `nalt` representa o número de alternativas;
- o processador de número `MyProcID` possuirá um subconjunto de alternativas correspondente a $\{linf, \dots, lsup\}$.

Cada processador i deve então calcular $\sigma(a,b)$ para todo o par ordenado de alternativas (a,b) pertencente ao conjunto P_i , através da confrontação de cada uma das "suas" alternativas $a \in A_i$ com todas as alternativas $b \in A - \{a\}$, considerando todos os critérios. Como o índice de credibilidade $\sigma(a,b)$ é calculado para todos os pares ordenados de alternativas, não resulta qualquer redundância (cálculos repetidos em processadores diferentes). À medida que os índices são calculados, são também enviados ao processador de interface de acordo com o padrão de comunicação ilustrado na Figura 5.9 (que é uma repetição da Figura 4.4), no qual o encaminhamento das mensagens está a cargo de fios de processamento cujo código varia consoante o processador em que se encontram, tal como descrito para o método PROMETHEE na secção 4.5.

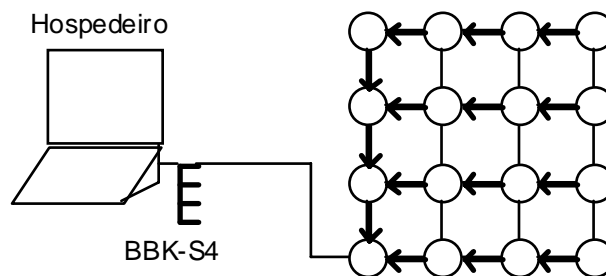


Figura 5.9 - "Recolha" dos resultados pelo nó interface

Esta fase termina logo que o processador de interface tenha recebido os índices de credibilidade relativos a todos os pares ordenados pertencentes a $A \times A$, com excepção dos pares $\{(a,a) : a \in A\}$, para os quais não se define este índice. O processador de interface procede então à execução dos passos 2 a 4 (exploração da relação de prevalência) do algoritmo apresentado na secção 5.3 de modo sequencial, enquanto os restantes processadores permanecem inactivos. No caso do utilizador desejar alterar alguns parâmetros podem distinguir-se duas situações: se a alteração se cingir aos parâmetros do limiar $s(\cdot)$ então repete-se a exploração utilizando apenas o processador de

interface (em modo sequencial, portanto); caso contrário, o processador de interface envia aos restantes processadores da rede as alterações ocorridas (conforme ilustrado na Figura 5.8) e o cálculo dos novos índices de credibilidade processa-se em modo paralelo. Nesta fase de análise de sensibilidade a carga computacional é eventualmente menor, repetindo-se apenas os cálculos estritamente necessários.

Este programa pp1 é pouco ambicioso, dado que apenas paraleliza a primeira fase do método ELECTRE III. A segunda fase do método é executada em modo sequencial pelo nó de interface, que foi escolhido para este fim dado que é aquele que mais rapidamente comunica com o hospedeiro.

Segundo programa paralelo (nome abreviado "pp2")

O segundo programa paralelo (pp2) foi construído a partir do programa pp1, acrescentando-lhe a característica de paralelizar uma parte da fase de exploração da relação de prevalência. Conforme se referiu na secção 5.4, a paralelização desta fase apresenta várias dificuldades, tendo-se identificado a possibilidade de paralelizar os subproblemas 1 a 4 do algoritmo apresentado na secção 5.3 (passo 2 e passo 3). O facto de estes subproblemas representarem a quase totalidade da computação necessária ao cálculo das destilações torna interessante a sua paralelização, mas é difícil antever se a quantidade de comunicação entre os diversos processadores implicada pela paralelização permitirá bons desempenhos. No caso deste programa pp2, a destilação descendente antecede a ascendente e a paralelização é limitada aos subproblemas 1 e 2, que se referem à identificação do máximo índice de credibilidade.

A partição P_1, P_2, \dots, P_p definida para o programa pp1 mantém-se inalterada na paralelização da segunda fase do método ELECTRE III. Cada processador i calcula inicialmente qual o maior índice de credibilidade, máximo $\sigma(a,b)$ para $(a,b) \in P_i$ (subproblema 1), e envia-o ao processador de interface. Este processador, em vez de ter de calcular o maior de entre $m^2 - m$ números, tem apenas de identificar qual o maior dos $p-1$ números que recebe, e compará-lo com o maior índice encontrado na sua partição. A paralelização do subproblema 2 é em tudo análoga. Toda a restante computação (nomeadamente a respeitante aos subproblemas 3 e 4) é executada em modo sequencial pelo processador de interface, que assume a função de coordenador (pivot) de todo o processo.

A grande dificuldade presente na paralelização do algoritmo de destilação (seja a descendente ou a ascendente) é a coordenação entre os processadores. De facto, cada processador que não seja nó de interface só pode contribuir para a resolução dos subproblemas 1 e 2 (ver algoritmo da secção 5.3) se conhecer, em cada momento, quais as alternativas que permanecem nos conjuntos B e D e qual o valor da variável λ . Tal facto requer que o processador de interface divulgue aos restantes processadores as alterações sofridas por D e λ , sempre que estas ocorram. Dado que os processadores

conseguirão deduzir as alterações no conjunto B a partir das alterações a D, não se torna necessário comunicar as primeiras.

A comunicação dos valores que a variável λ vai tomando é simples, implicando apenas a divulgação do novo valor da variável. No que respeita à comunicação da composição do conjunto D já surgem dificuldades, a maior das quais é decidir se se comunica quais as alternativas que permanecem no conjunto ou se se comunica quais as alternativas que são retiradas do conjunto. Optou-se pela primeira hipótese, após reconhecer que o algoritmo de destilação tenta em cada passo reduzir o mais possível o número de elementos do conjunto D, desejando-se que, se possível, D contenha apenas uma alternativa no fim de cada iteração do ciclo principal.

O programa pp2 pretende melhorar o pp1, o que só sucederá se a paralelização da identificação dos maiores índices de credibilidade, nas várias ocasiões em que deve ser efectuada, compensar a comunicação necessária à recolha dos resultados e à actualização das variáveis que contêm, nos diversos processadores, o conjunto D e o índice λ .

Terceiro programa paralelo (nome abreviado "pp3")

O programa pp3 é uma extensão do programa pp2, que além de paralelizar os subproblemas 1 e 2, paraleliza os subproblemas 3 e 4. O subproblema 3, recorde-se, consiste em calcular a qualificação de todas as alternativas, enquanto o subproblema 4 consiste em identificar qual a maior (ou menor, consoante se trate da destilação descendente ou ascendente, respectivamente) dessas qualificações.

Neste programa, a destilação descendente continua a preceder a destilação ascendente e o processador de interface coordena a execução das destilações. Para resolver o subproblema 3, cada processador i calcula a qualificação de todas as alternativas $a \in A_i$, necessitando para isso de conhecer os índices de credibilidade $\sigma(a,b)$ e $\sigma(b,a)$ para $(a,b) \in P_i$. Se para todos os pares ordenados $(a,b) \in P_i$ o processador i já calculou, durante a primeira fase do método, os índices $\sigma(a,b)$, o mesmo poderá não acontecer com os índices $\sigma(b,a)$. Em conformidade, para que cada processador possa calcular a qualificação das “suas” alternativas $a \in A_i$, houve que optar entre a existência de comunicação em larga escala, na qual o processador recebe de outros processadores os índices que necessita, e a existência de uma duplicação da computação, na qual o processador calcula esses índices.

Não é necessário efectuar experiências para concluir que a primeira possibilidade nunca poderia proporcionar desempenhos interessantes, dada a quantidade de comunicação que passaria a existir. De facto, a ser escolhida esta estratégia, cada processador teria de receber dados de todos os restantes. Optou-se por isso pela outra possibilidade, que introduz redundância entre os cálculos efectuados em processadores distintos. O esforço computacional durante a primeira fase do método é aproximadamente duplicado, dado que os índices $\sigma(b,a)$ para $(a,b) \in P_i$ terão já sido calculados por outro

processador. As consequências desse esforço adicional serão porém atenuadas pelo facto deste ocorrer enquanto se comunicam os resultados $\sigma(a,b)$ para $(a,b) \in P_i$.

Para resolver o subproblema 4, cada processador envia ao processador de interface a maior (ou menor, consoante a destilação: assume-se no que segue que se trata da destilação descendente) qualificação que encontrou no decurso do subproblema 3, acompanhada da referência à alternativa que obteve essa qualificação. No caso de haver várias alternativas com qualificação máxima, o processador envia-as todas, dado que o processador de interface necessitará de saber quais são. Como é natural, a qualificação máxima encontrada em cada processador não será a mesma, pelo que será necessário que o processador de interface identifique qual a qualificação máxima considerando as mensagens enviadas pelos restantes processadores e a compare com o máximo que o próprio processador de interface encontrou.

Assinale-se por último que cada nó, ao servir de intermediário na comunicação com o nó de interface (recorde-se a Figura 5.9), examina as mensagens de máxima qualificação e toma duas iniciativas: por um lado, actualiza o seu conceito de máxima qualificação; por outro lado, não retransmite a mensagem (filtra-a) se a qualificação máxima indicada na mesma for inferior ao máximo que já conhece. No entanto, esta filtragem obriga a que cada processador envie ao processador pivot uma mensagem especial que indica não ter mais qualificações a transmitir.

O programa pp3 é mais complexo que os programas anteriores, tentando paralelizar, na medida do possível, os algoritmos de destilação.

Quarto programa paralelo (nome abreviado "pp4")

O quarto programa paralelo distingue-se do terceiro pelo facto de calcular as destilações descendente e ascendente simultaneamente, em modo pseudo-paralelo. Enquanto no programa pp3 a destilação ascendente só era iniciada quando a descendente tivesse sido obtida, no programa pp4 as duas destilações são efectuadas em fios de processamento iniciados ao mesmo tempo. Deste modo, os fios que calculam as destilações entram em concorrência pelo tempo de processamento em cada processador. Nesta abordagem cada fio de processamento opera, em cada processador, sobre uma partição definida do modo descrito para o programa pp3, sendo o processo de cálculo em tudo idêntico ao descrito para esse programa, incluindo a necessidade de efectuar computação redundante.

A principal motivação para a experimentação desta estratégia é a de aumentar a ocupação útil de cada processador. De facto, no programa pp3, a contribuição de cada processador i para o cálculo de uma destilação era proporcional, em cada momento, ao número de alternativas pertencentes a A_i ainda por classificar. Por esse motivo, um processador ficaria rapidamente "sem trabalho" se as alternativas pertencentes a A_i fossem 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 (pp4), espera-se que haja um efeito de compensação entre o trabalho computacional envolvido nas duas destilações. Por exemplo, se um processador i possuir em A_i alternativas muito boas, então será pouco o seu trabalho no que respeita à destilação descendente (que classifica em primeiro lugar as alternativas mais qualificadas) e será muito o seu trabalho na destilação ascendente (que classifica em primeiro lugar as alternativas menos qualificadas). Pretende-se deste modo melhorar o programa pp3, obtendo uma distribuição mais equilibrada da carga computacional.

As vantagens de seguir a estratégia acima descrita são porém acompanhadas do inconveniente de requererem que a comunicação entre processadores, devida a cada destilação, ocorra em simultâneo. Dado que esse factor poderia causar maiores estrangulamentos nas comunicações, sobretudo no processador que é utilizado como pivot, comparou-se experimentalmente a utilização de um pivot comum (processador 0, ou seja o processador do "canto sudoeste") com uma versão em que coexistem dois pivots. A opção recaiu sobre a utilização de dois pivots e dois padrões de comunicação distintos (representados na Figura 5.10 para o caso da recolha de resultados pelo processador pivot), dado que foi a que obteve melhores desempenhos nas experiências efectuadas. Esta abordagem implica que no final da destilação ascendente (cujo pivot é o nó noroeste) os resultados da mesma sejam comunicados ao processador de interface (nó sudoeste).

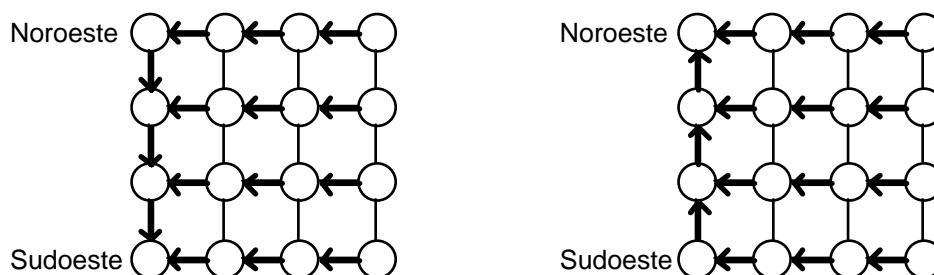


Figura 5.10 - Padrões de recolha quando o pivot é o processador no canto sudoeste (padrão da esquerda) e noroeste (padrão da direita)

Quinto programa paralelo (nome abreviado "pp5")

O quinto programa paralelo é uma extensão ao programa pp4, que executa em modo paralelo a combinação das pré-ordens completas resultantes das destilações numa pré-ordem parcial (P,I,R), isto é, o quarto passo do algoritmo apresentado na secção 5.3.

Se se atentar na forma como se paralelizaram os subproblemas 1 a 4, verifica-se que um efeito secundário benéfico resulta da necessidade de informar todos os processadores sobre as alternativas que foram classificadas em cada iteração dos algoritmos de destilação. O efeito consiste no facto de cada processador poder, com essa informação, conhecer os resultados das duas destilações sem necessitar de comunicação

adicional. Por esse motivo, assim que terminem ambas as destilações, cada processador está apto a contribuir para a obtenção da pré-ordem parcial final.

No programa pp5, cada processador é responsável pela determinação de qual das situações se verifica para cada par ordenado $(a,b) \square P_i$: $a P b$, $a I b$, $a R b$ ou nenhuma das anteriores ($b P a$). Os resultados vão sendo enviados para o processador de interface, aplicando-se uma estratégia de eliminação de comunicação redundante idêntica à utilizada na comunicação dos resultados do método PROMETHEE I. Assim, depois de calcular $a_j H a_k$ (H representa uma das relações binárias que pode resultar: P , I , ou R), um processador comporta-se de acordo com o seguinte:

- se $a_j P a_k$ então envia este resultado ao nó de interface;
- se $a_j I a_k$ então só envia este resultado ao nó de interface se $j < k$;
- se $a_j R a_k$ então não envia este resultado ao nó de interface.

O programa pp5 é o mais complexo dos construídos, tentando explorar as possibilidades mais viáveis identificadas na secção 5.4.

Funcionalidade dos programas

Os programas paralelos pp1 a pp5 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 requeridos pelo método ELECTRE III é pedida ao utilizador a informação de saída por este pretendida, sendo essa informação comunicada a todos os processadores juntamente com os dados do problema. Os diferentes processadores utilizam então essa informação de modo a não originar mais comunicação do que a necessária: os processadores só comunicarão ao processador de interface os índices de credibilidade se o decisor tiver expresso desejo de os conhecer.

O modo em que os diversos programas efectuem os cálculos é resumido no quadro seguinte:

| | série | pp1 | pp2 | pp3/4 | pp5 |
|---------------------------|------------|------------|------------|------------|----------|
| Índices de credibilidade | sequencial | paralelo | paralelo | paralelo | paralelo |
| Subproblemas 1 e 2 | sequencial | sequencial | paralelo | paralelo | paralelo |
| Subproblemas 3 e 4 | sequencial | sequencial | sequencial | paralelo | paralelo |
| Pré-ordem parcial (P,I,R) | sequencial | sequencial | sequencial | sequencial | paralelo |

5.6 Experiências efectuadas

Conduziram-se várias experiências de modo a estudar os desempenhos dos programas paralelos em diversas situações. Os aspectos passíveis de estudo são muito numerosos quando combinados, pelo que se optou por utilizar os resultados de algumas experiências para restringir as situações a estudar nas experiências seguintes. Os aspectos estudados foram:

Dimensões da matriz de decisão

O problema computacional de executar o método ELECTRE III pode tornar-se grande quer em número de alternativas quer em número de critérios. Uma primeira consequência é a de que a computação necessária à obtenção da relação de prevalência depende, aproximadamente, de modo linear do número de critérios e de modo quadrático do número de alternativas. Como segunda consequência, a computação necessária à exploração da relação de prevalência será geralmente maior quando aumenta o número de alternativas, não dependendo contudo do número de critérios. No entanto, a duração desta fase é menos previsível dada a sua dependência dos dados de entrada.

Face a restrições de memória, foram construídos, com dados fictícios, 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 de tão elevado número de critérios resulta da combinação de 4, 5, 8 ou 10 pontos de vista com 2, 5 ou 10 maneiras distintas de operacionalizar cada ponto de vista (que diferem apenas nos parâmetros que definem k_j , q_j , p_j e v_j). Permite-se assim um cenário no qual 2, 5 ou 10 actores operacionalizem 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. Adoptou-se esta estratégia meramente com vista à realização de testes com grande número de critérios. Tal estudo tornar-se-ia de outra forma pouco viável dado que, de acordo com experiências efectuadas, a geração de problemas com demasiados pontos de vista considerando um só actor conduziu a resultados em que todas as alternativas eram consideradas indiferentes (esta questão não se colocava no Cap. 4). A utilização desta estratégia permite ainda aumentar o número de critérios através do incremento do número de actores, sem com isso aumentar a dimensão da matriz de decisão (pressupõe-se que os actores estão de acordo em relação ao desempenho de cada alternativa, mas em desacordo sobre o peso e os limiares a atribuir a cada critério).

Para os bancos 1 e 2 gerou-se um problema para cada dimensão (pretende-se efectuar um estudo qualitativo, sem pretender obter dados significativos do ponto de vista estatístico), podendo sumariar-se o seu conteúdo como segue:

Banco 1

{4, 5, 8, 10} pontos de vista x {1} actor x {16, 32, 64, 128, 192, 256} alternativas

Banco 2

{4, 5, 8, 10} pontos de vista x {2, 5, 10} actores x {16, 20, 24, 28, 32, 50} alternativas

Etapa de utilização do programa

No início da utilização de um programa que executa o método ELECTRE III obtém-se um conjunto de resultados (uma solução) para um conjunto inicial de dados. No entanto, é muito natural que o decisor deseje verificar a robustez da solução obtida através de análise de sensibilidade, devido à incerteza com que indicou valores para alguns parâmetros. Os programas construídos permitem efectuar análises de sensibilidade aos parâmetros dos critérios e/ou aos parâmetros α e β do limiar $s(\cdot)$.

A análise de sensibilidade aos parâmetros dos critérios é importante, na medida em que estes expressam de algum modo valores do decisor, que este pode ter dificuldade em comunicar. Por outro lado, importa efectuar análises de sensibilidade ao limiar $s(\cdot)$, dado que este limiar não tem qualquer significado físico ou económico aparente, sendo por isso difícil defender a utilização de um conjunto particular de valores para os seus parâmetros α e β .

Podem assim identificar-se três etapas distintas da utilização de um dos programas construídos:

- obtenção de uma primeira solução;
- alteração de um parâmetro de um critério (análise de sensibilidade a $(k_j, q_j, p_j$ e $v_j)$);
- alteração da definição do limiar $s(\cdot)$ (análise de sensibilidade a α e β).

Mediram-se os desempenhos em cada uma destas etapas, dado que a importância de obter tempos de resposta rápidos é maior nas duas últimas do que na primeira, conforme se argumenta no capítulo 1. Embora tivesse sido possível, não se mediram desempenhos em situações de análise de sensibilidade em que se altera mais do que um critério ao mesmo tempo. Nesses casos o desempenho situar-se-á entre o da análise a um só critério e o da obtenção da primeira solução, estando tão próximo deste último quanto maior for o número de critérios afectado de cada vez.

Resultados pretendidos na utilização do programa

Ao utilizar uma ferramenta para executar o método ELECTRE III, dois resultados podem ser pretendidos: a relação de prevalência difusa (definida pelos índices de credibilidade) e os resultados da exploração dessa relação (definida pelas duas destilações e pela pré-ordem parcial).

O desejo, por parte do agente de decisão, de obter os índices de credibilidade afecta bastante o desempenho dos programas pp3, pp4 e pp5, dado que se evita muita

comunicação entre processadores no caso (muito provável caso haja muitas alternativas) de o decisor não desejar conhecer esses números. As experiências relativas a este aspecto cingiram-se à execução do programa pp4, dado que nas experiências anteriores este obteve melhores desempenhos que os programas pp3 e pp5.

Topologia utilizada

A topologia de rede matricial pode ser utilizada pelos programas qualquer que seja a sua dimensão (o programa não pressupõe nenhuma dimensão da rede). Experimentou-se utilizar as mesmas topologias escolhidas para o método PROMETHEE, ou seja, redes com quatro linhas por quatro colunas (4 x 4), quatro linhas por duas colunas (4 x 2), duas linhas por quatro colunas (2 x 4) e duas linhas por duas colunas (2 x 2). As experiências relativas a este aspecto cingiram-se à execução do programa pp4.

Os tempos de execução foram medidos utilizando o relógio de tempo real do PARIX, que permite medir tempos com um erro máximo de 64 μ s. A medida de desempenho (ver sec. 3.3) escolhida para avaliar o mérito relativo de cada paralelização foi o speedup real $S(n,p)$, ou seja, a razão entre os tempos de execução usando o programa sequencial e os tempos de execução usando um programa paralelo. Usar-se-á por comodidade a expressão "razão S" (ou "speedup"), estando o tamanho do programa e o número de processadores implícitos no contexto da expressão.

5.7 Resultados obtidos

Conjunto 1) de experiências: comparação dos vários programas paralelos na utilização dos 16 processadores

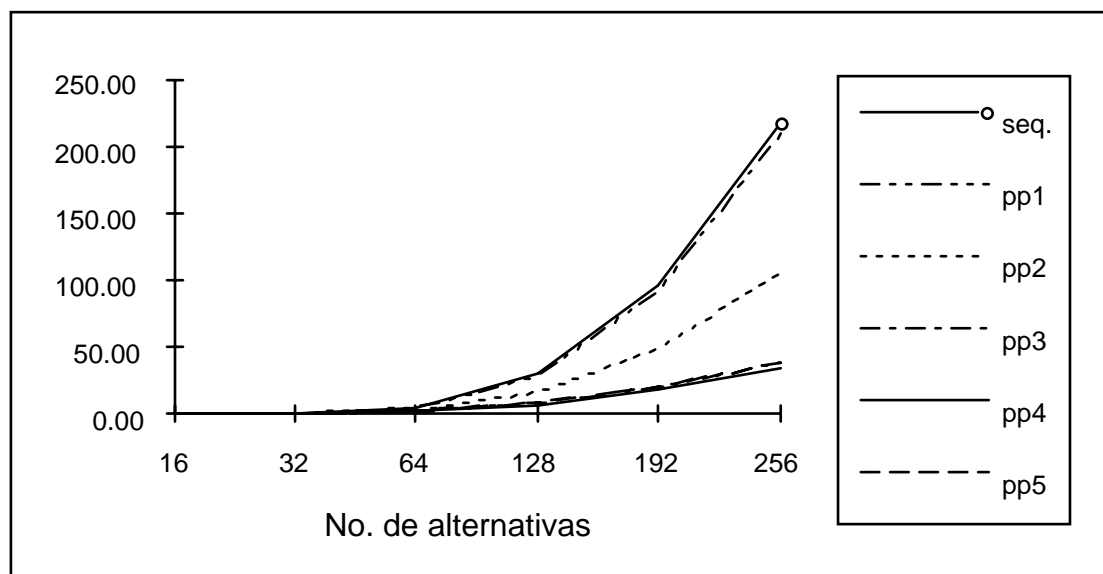


Figura 5.11a) - Tempo de execução para um problema com 4 critérios

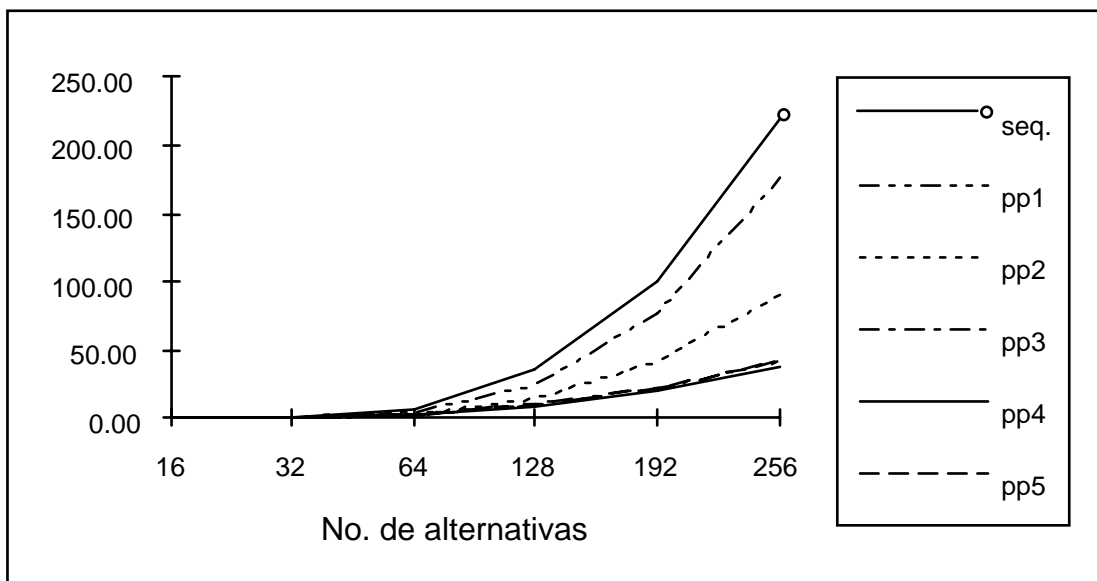


Figure 5.11b) - Tempo de execução para um problema com 10 critérios

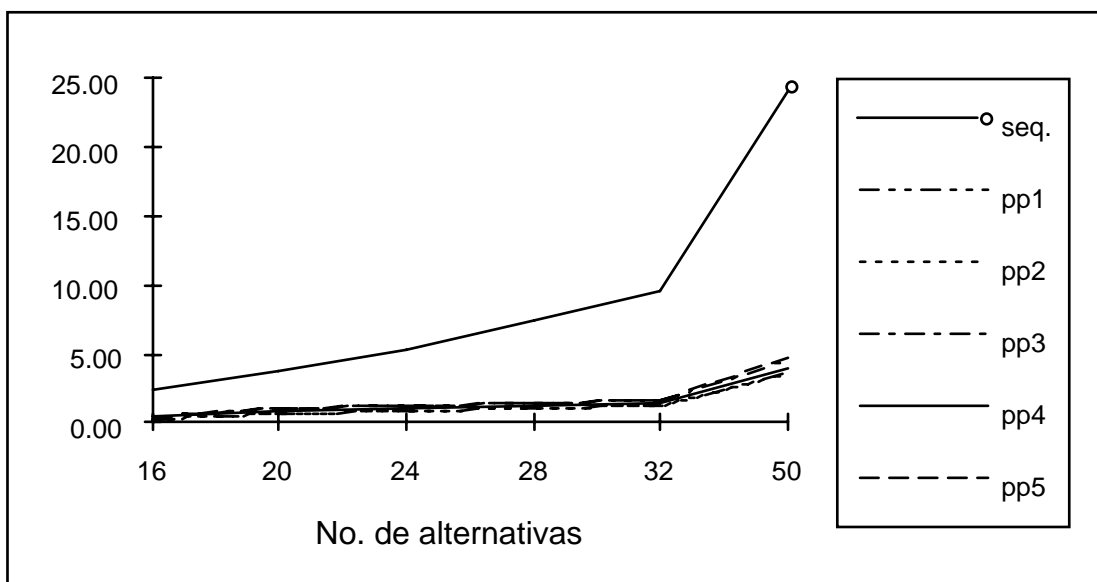


Figure 5.11c) - Tempo de execução para um problema com 10 x 10 critérios

As Figuras 5.11a) b) e c) apresentam o tempo de execução (em segundos) necessário para obter uma primeira solução para alguns problemas do banco 1 e 2, numa situação em que o decisor abdica de conhecer os índices de credibilidade. O tempo de execução apresentado é o que medeia entre o início da divulgação dos dados iniciais por todos os processadores (no caso dos programas paralelos) e o início da escrita no écran dos resultados.

Os resultados apresentados nestes gráficos podem também encontrar-se nos quadros 1 e 2 do Apêndice B, para o programa sequencial, e nos quadros 3 a 12 do mesmo apêndice, para os programas paralelos, já convertidos na razão S (pode calcular-se o

tempo de execução de um programa paralelo dividindo o tempo de execução do programa sequencial pela razão S).

Analisando os resultados relativos ao programa sequencial pode concluir-se que o número de alternativas contribuiu mais para a lentidão dos programas do que o número de critérios. Enquanto o tempo de cálculo máximo é de 24 segundos para os problemas no banco 2, encontra-se um máximo de cerca de 3 minutos e 39 segundos para os problemas do banco 1. Observa-se também, nas colunas A e B dos quadros 1 e 2 do Apêndice B, que nos problemas do banco 2 (que possuem muitos critérios e poucas alternativas) a proporção do tempo necessário ao cálculo da relação de prevalência é muito superior ao que se verifica para o banco 1 (que possuem poucos critérios e muitas alternativas).

Estudando o desempenho dos programas paralelos pode verificar-se que o programa pp4 foi o mais rápido nos problemas do banco 1 (veja-se também Figura 5.11 a) e b), em que o programa sequencial é o mais lento). No que concerne aos problemas do banco 2 torna-se difícil isolar um programa que exiba melhor desempenho (veja-se também Figura 5.11c), em que o número máximo de alternativas é de apenas 50). Uma análise mais exaustiva permite concluir que, em qualquer situação, os programas que exibem melhor desempenho são ora o pp1 ora o pp4. Por esse motivo, excluíram-se os programas pp2, pp3 e pp5 das experiências conduzidas posteriormente.

O facto de o programa pp3 ser mais lento que o programa pp4 não é surpreendente, dado que este último é claramente um melhoramento do primeiro. Já a transição do programa pp2 para o programa pp4 não constituía, à partida, garante de melhores desempenhos, devido à computação redundante necessária neste último (recorde-se a secção 5.5). Nos resultados que se apresentam no Apêndice B verifica-se que o programa pp4 foi mais rápido que o programa pp2 nos problemas com muitas alternativas do banco 1, mas mais lento nos programas com muitos critérios do banco 2 (em que o programa pp1 é o mais rápido). A explicação desse facto está precisamente na carga extra que representa o cálculo redundante: essa carga é inferior a 11% do tempo total de execução sequencial para os problemas do banco 1, e aumenta até perto de 70% para os problemas com 10 x 10 critérios do banco 2 (os resultados para os restantes casos não são apresentados, dado que estão relacionados com a proporção do tempo utilizado no cálculo da relação de prevalência, apresentado nos quadros 1 e 2 do Apêndice B). Por fim, na comparação do programa pp4 com o programa pp5 verifica-se que o primeiro foi sistematicamente mais rápido: a quantidade de computação necessária à obtenção da pré-ordem parcial (P,I,R) não parece ser suficiente para compensar a quantidade de comunicação entre processadores envolvida na paralelização deste passo do algoritmo do ELECTRE III.

Conjunto 2) de experiências: comparação dos programas pp1 e pp4 nas três etapas de utilização, numa rede com 16 processadores

Considerando agora apenas os programas paralelos pp1 e pp4, observe-se o seu desempenho medido pela razão S , relativo ao programa sequencial. As Figuras 5.12 a) b) e c) apresentam resultados para o caso com menor número de critérios (4 critérios - banco 1 de problemas) em três etapas de utilização do programa. As Figuras 5.13 a) b) e c) apresentam resultados análogos para o caso com maior número de critérios (10 x 10 critérios - banco 2 de problemas).

Nas Figuras 5.12 e 5.13 pode-se observar a diferença entre a necessidade de recolher os índices de credibilidade e a ausência dessa necessidade. A designação pp4* refere-se ao caso em que o decisor não está disposto a abdicar de conhecer os índices de credibilidade (veja-se também quadros 13 e 14 do Apêndice B), enquanto a designação pp4 continua a referir-se ao caso em que o decisor abdica de conhecer esses índices. Nas Figuras 5.12 e 5.13, alíneas a) e b), pode verificar-se que o programa pp4 é mais rápido quando não necessita de recolher os índices de credibilidade, enquanto nas Figuras 5.12c) e 5.13c) se confirma que essa distinção não influencia a análise de sensibilidade ao limiar $s(\cdot)$, que só intervém na fase de exploração. Para o programa pp1 essa distinção nunca influencia o seu desempenho, dado que o processador de interface necessita sempre de recolher os índices de credibilidade para proceder em modo sequencial à exploração da relação por estes definida.

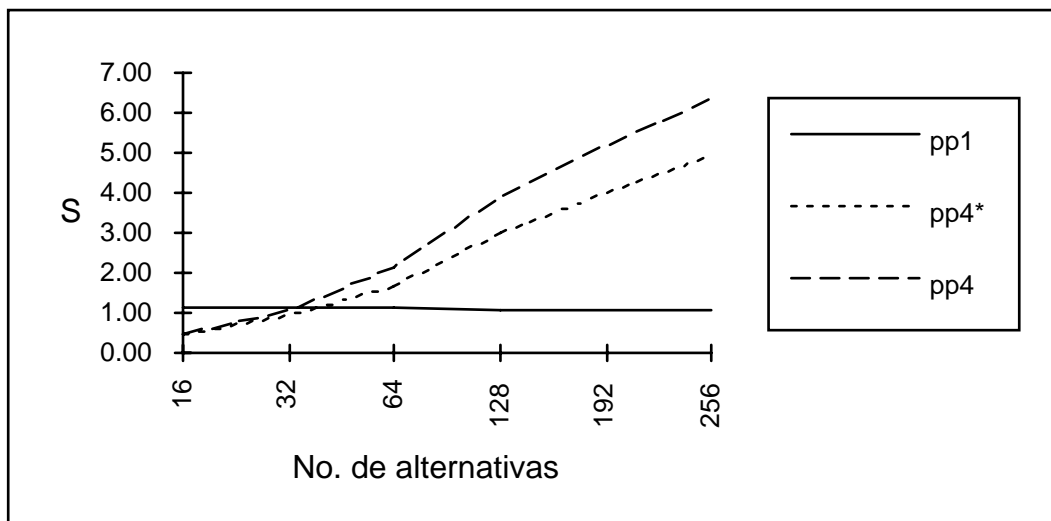


Figure 5.12a) - Razão S na obtenção da primeira solução

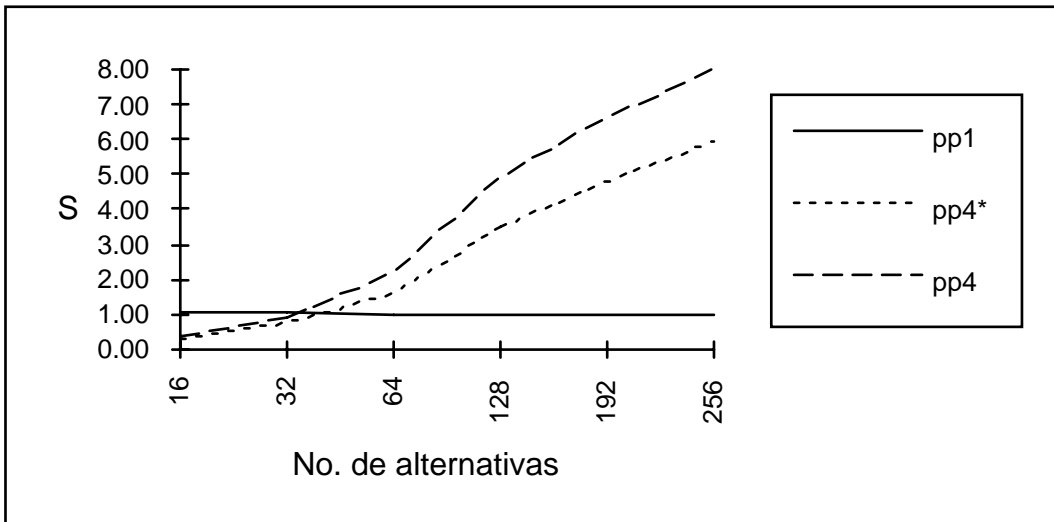


Figure 5.12b) - Razão S na análise da alteração de um critério

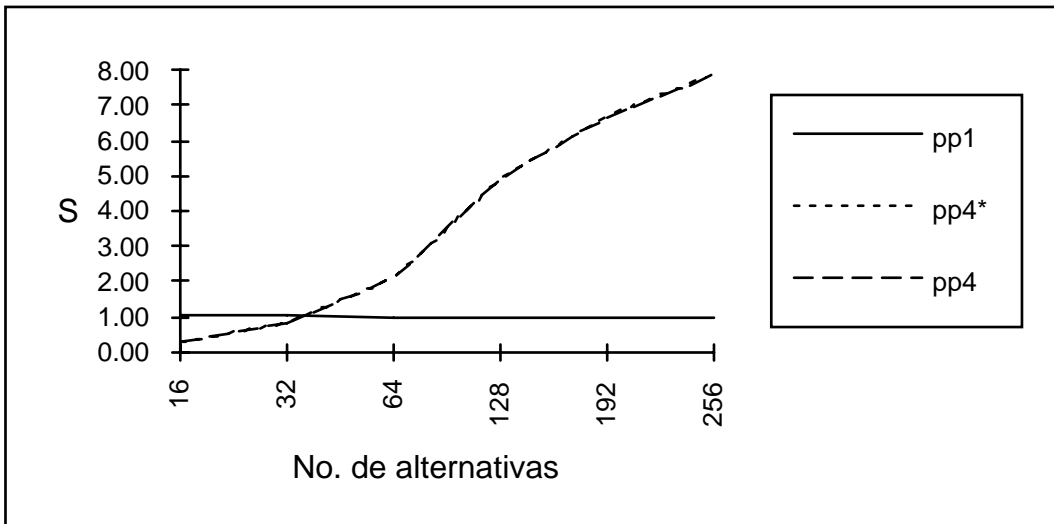


Figure 5.12c) - Razão S na análise da alteração do limiar s(.)

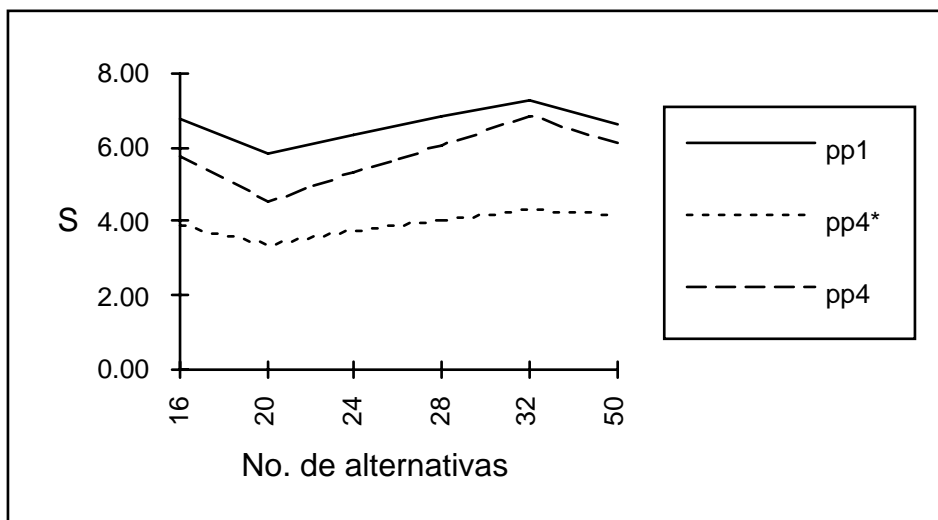


Figure 5.13a) - Razão S na obtenção da primeira solução

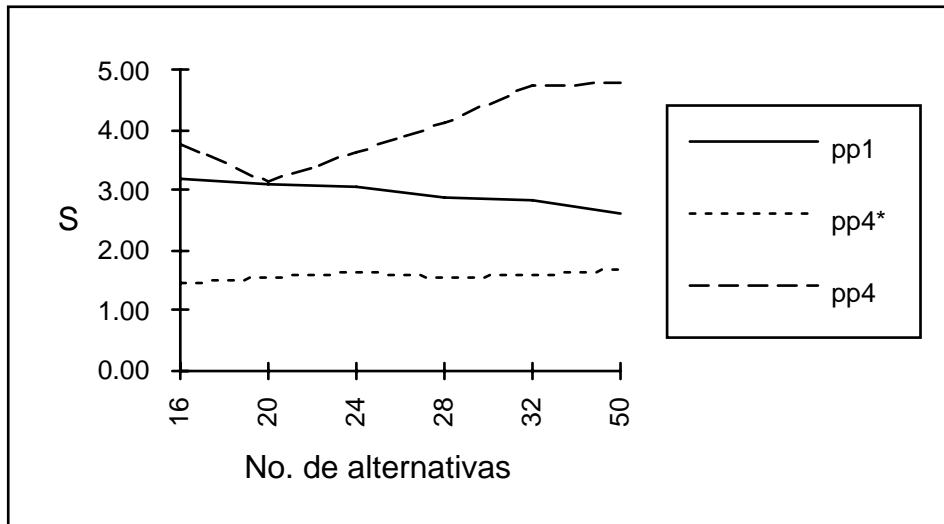


Figure 5.13b) - Razão S na análise da alteração de um critério

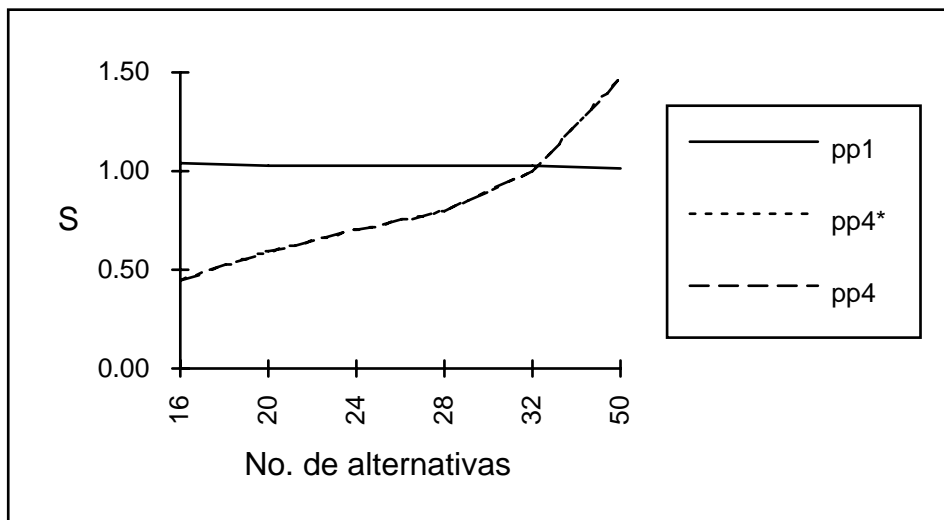


Figure 5.13c) - Razão S na análise da alteração do limiar $s(.)$

A Figura 5.13a) revela alguma irregularidade no desempenho dos programas, com picos nas 16 e 32 alternativas. A origem deste comportamento é a de a distribuição da carga computacional entre os diversos processadores ser mais equilibrada quando o número de alternativas é múltiplo do número de processadores. Consegue-se neste caso uma partição mais equitativa, conforme o exposto na secção 5.5. Note-se que na Figura 5.12 não surge esta irregularidade dado que os problemas do banco 1 possuem todos um número de alternativas múltiplo de 16.

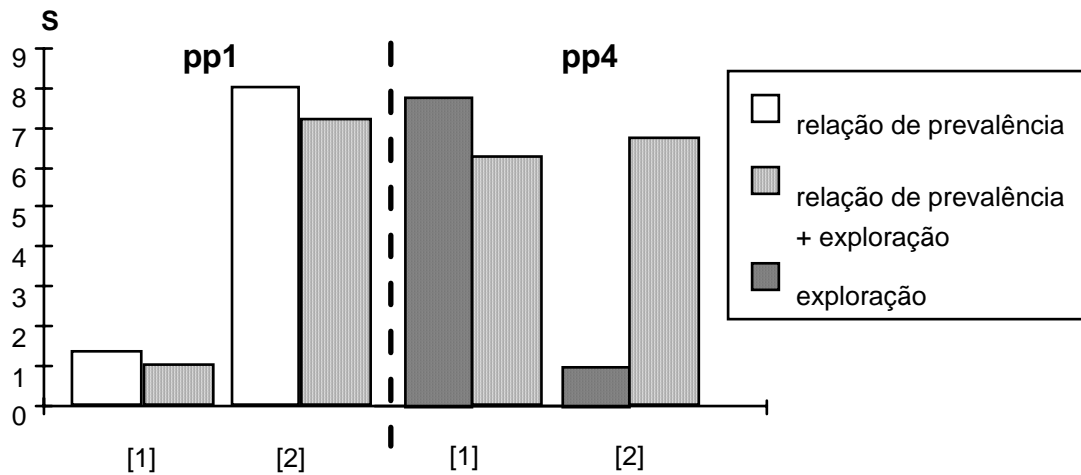


Figura 5.14 - Razão S: parciais e totais para o problema com 4 critérios e 256 alternativas [1] e para o problema com 10 x 10 critérios e 32 alternativas [2]

Comparando os desempenhos dos programas pp1 e pp4 (veja-se Figuras 5.12 e 5.13), pode observar-se que o programa pp4 ultrapassou em rapidez o programa pp1 quando o número de alternativas cresceu de 32 para 64 ou 50, conforme o caso. Como o programa pp1 só paraleliza a fase de obtenção dos índices de credibilidade, o seu desempenho piora à medida que a fase de exploração da relação de prevalência representa proporções maiores do tempo de execução total. Conjugando este facto com os resultados apresentados nos quadros 1 e 2 do Apêndice B conclui-se que o programa pp1 não consegue acompanhar o programa pp4 quando o número de alternativas aumenta. A Figura 5.14 mostra como a razão S, medida considerando só a obtenção da relação de prevalência ou só a exploração da relação de prevalência, contribui para a razão S na obtenção de uma solução inicial considerando ambas as fases, para os programas pp1 e pp4, respectivamente.

Não é possível de deixar de referir que, mesmo nos casos mais favoráveis, os desempenhos medidos pela razão S ficam aquém dos máximos obtidos pelos programas dedicados ao método PROMETHEE. Note-se que, no entanto, as dimensões de problema que foi possível estudar são muito menores do que as conseguidas no estudo da paralelização do método PROMETHEE. Aparentemente, face à evolução dos resultados relativos à execução do método ELECTRE III, a razão S poderia ser maior se fossem executados problemas de dimensão maior, sobretudo no número de alternativas, para o programa pp4.

Ao analisar-se somente a razão S torna-se difícil apontar um programa melhor entre o pp1 e o pp4: trata-se, de certa forma, de duas alternativas não dominadas. Porém, a análise dos tempos de execução desses programas permite extrair duas conclusões. Em primeiro lugar, o programa pp1 só foi mais rápido que o pp4 nas situações em que até o programa mais lento construído (o sequencial) é bastante rápido (cf. Apêndice B e Quadros 5.1 e 5.2).

| 4x256 | Obtenção da 1ª Solução | Alteração a um critério | Alteração ao limiar s(.) |
|--------------|------------------------|-------------------------|--------------------------|
| sequencial | 218 | 205 | 189 |
| pp1 | 210 | 204 | 188 |
| pp4* | 45 | 35 | 24 |
| pp4 | 35 | 26 | 24 |

Quadro 5.1 - tempo de execução (em segundos) para o caso mais favorável ao programa pp4 (problema com 4 critérios e 256 alternativas)

| 10x10x32 | Obtenção da 1ª Solução | Alteração a um critério | Alteração ao limiar s(.) |
|-----------------|------------------------|-------------------------|--------------------------|
| sequencial | 10 | 1 | 0 |
| pp1 | 1 | 0 | 0 |
| pp4* | 2 | 0 | 0 |
| pp4 | 1 | 0 | 0 |

Quadro 5.2 - tempo de execução (em segundos) para o caso mais favorável ao programa pp1 (problema com 10 x 10 critérios e 32 alternativas)

Em segundo lugar, verifica-se que no programa pp4 o melhor desempenho em termos de razão S ocorre nas situações em que o programa sequencial foi mais lento e nas situações de análise de sensibilidade, nas quais a importância de obter tempos de resposta curtos foi já sublinhada. Estes desempenhos são de facto muito interessantes para uma ferramenta de apoio à decisão. Por exemplo, quando se pretendeu conhecer o impacto da alteração da definição de um critério num problema com 256 alternativas e 4 critérios foi necessário esperar 3 minutos e 25 segundos quando se utilizou o programa sequencial e 26 segundos quando se utilizou o programa paralelo. Os tempos foram de 3m16s e 24s, respectivamente, para a situação de alteração da definição do limiar técnico s(.).

Considerando os dois argumentos acima mencionados elegeu-se o programa pp4 como sendo o melhor (mais útil) apesar de a escolha não poder ser justificada apenas com base em elementos objectivos. Mostra-se de seguida (Figura 5.15) como variou a razão S com as dimensões dos problemas contidos no banco 1. A Figura 5.15c) apresenta um gráfico da razão S em função apenas do número de alternativas, dado que o número de critérios não influencia os tempos de execução na análise de sensibilidade aos parâmetros do limiar s(.).

Estes gráficos mostram que a razão S cresce com o número de alternativas, e que aparentemente cresceria mais se houvesse problemas com mais alternativas disponíveis. Embora se saiba que existe sempre um ponto de saturação, a partir do qual a razão S não aumentará mais com o número de alternativas, é razoável antever que essa saturação não foi atingida, para a rede de 16 processadores, com os casos de teste utilizados.

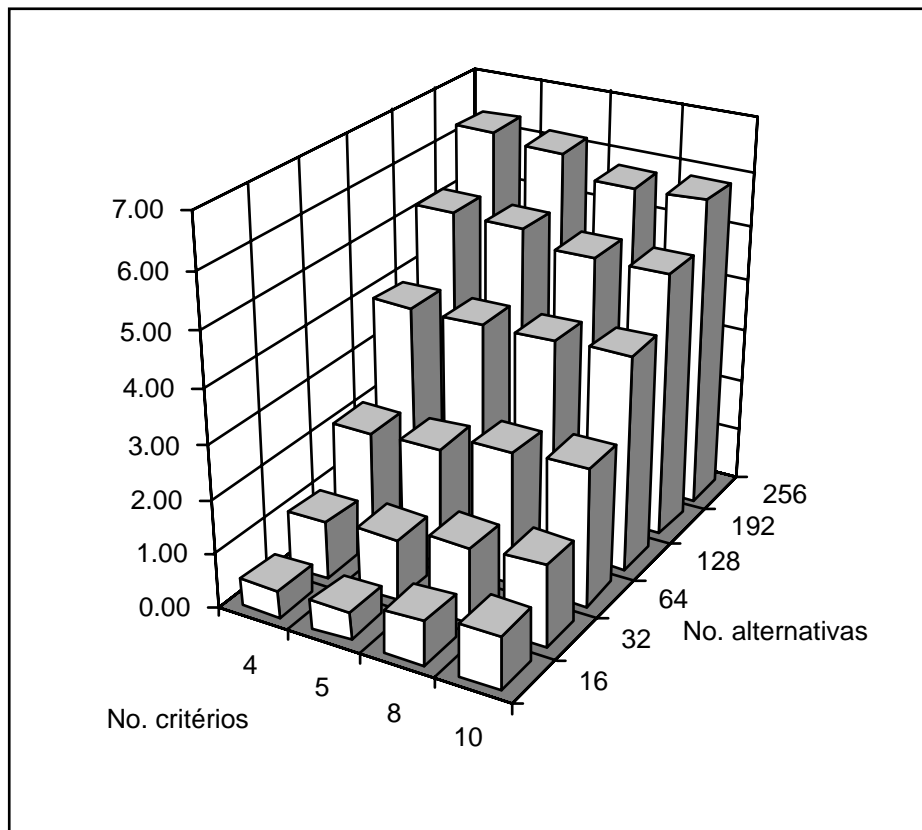


Figure 5.15a) - Evolução da razão S para o cálculo da primeira solução (pp4)

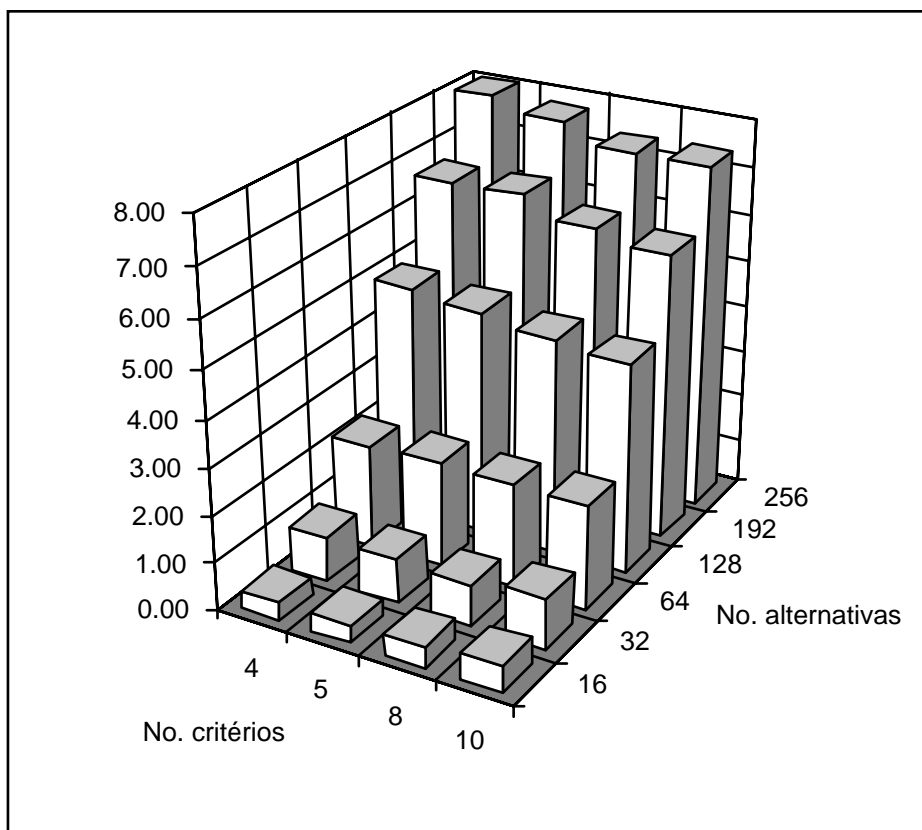


Figure 5.15b) - Evolução da razão S para analisar a alteração de um critério (pp4)

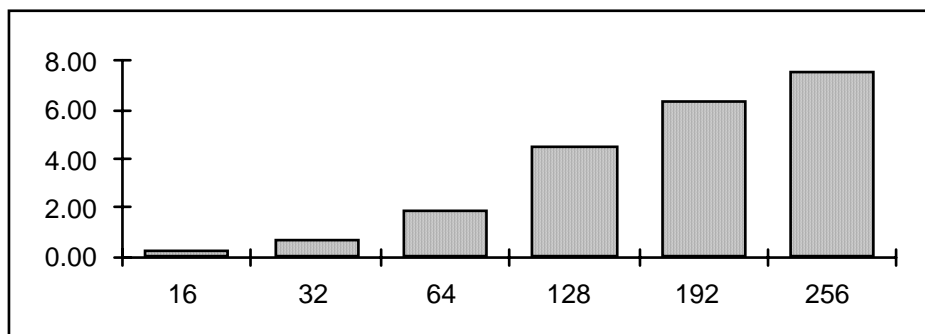


Figure 5.15c) - Evolução da razão S para analisar a alteração do limiar $s(.)$ (pp4)

Conjunto 3) de experiências: comparação de diferentes topologias de rede matricial, utilizando o programa pp4

O último estudo conduzido consistiu na comparação de várias dimensões da rede matricial. Experimentou-se correr os programas em redes quadrangulares com quatro e dezasseis processadores, e em redes rectangulares de oito processadores, estes dispostos em quatro linhas por duas colunas ou em duas linhas por quatro colunas. As razões S obtidas apresentam-se no Apêndice B (cf. quadros 5 e 6 para a rede 4 x 4 e quadros 15 a 20 para as restantes) e são ilustrados, para a obtenção de uma solução inicial dos problemas com quatro critérios, na Figura 5.16.

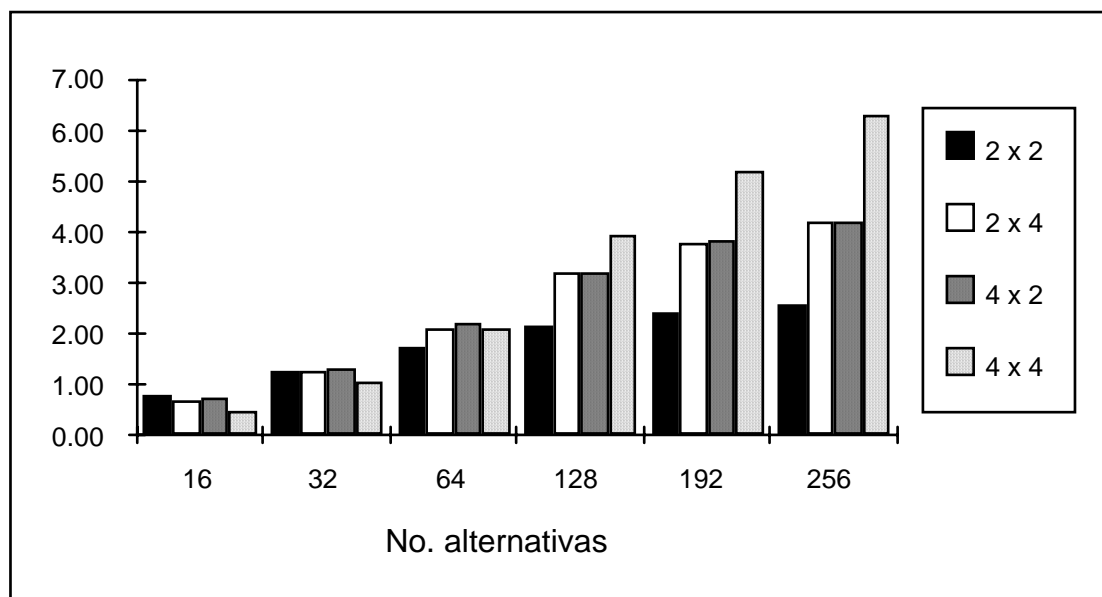


Figura 5.16 - Evolução da razão $S(p)$ na execução do programa pp4 para diferentes dimensões da rede matricial

A Figura 5.16 mostra que, em relação aos problemas do banco 1, só compensa ter mais de quatro processadores na rede quando o número de alternativas passa de 64 para 128. A partir desse ponto o desempenho da rede 2 x 2 atinge a saturação, dado que deixa

de aumentar significativamente com o número de alternativas. As redes com oito processadores não parecem estar saturadas, e a rede com os dezasseis processadores ainda o parece menos. Ao contrário do que sucedia com os programas dedicados ao PROMETHEE, nos programas dedicados ao ELECTRE III a diferença entre os desempenhos proporcionados pelas duas redes com oito processadores não é notória.

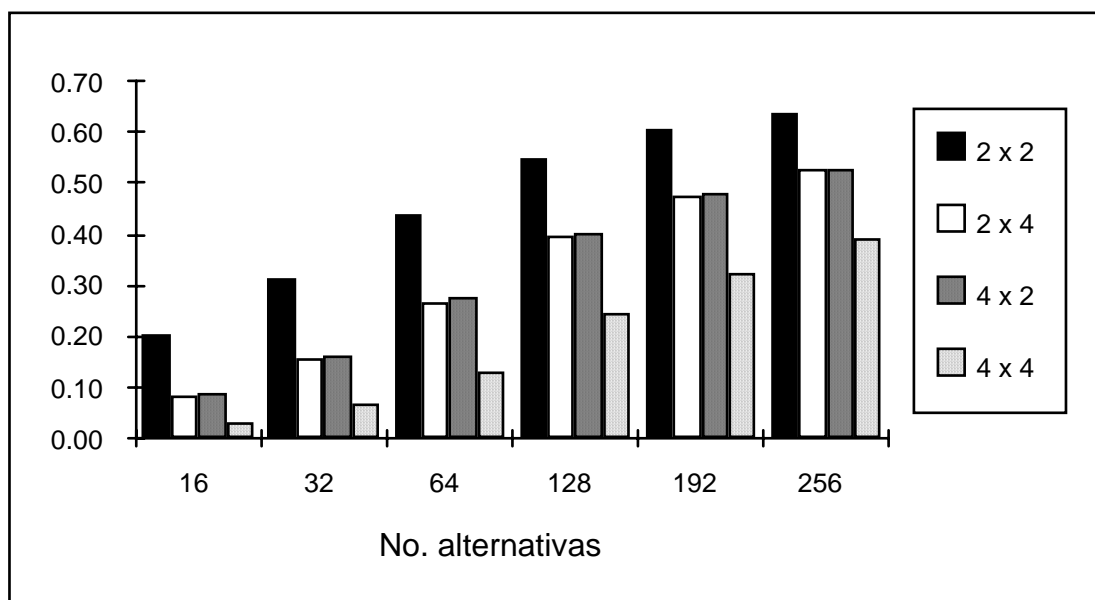


Figura 5.17 - Evolução da eficiência $E(p) = S(p) / p$

A Figura 5.17 apresenta a eficiência correspondente à razão S apresentada na Figura 5.16. Como é habitual nos programas paralelos, a eficiência decresce quando se aumenta o número de processadores (é mais fácil manter ocupados de modo útil poucos processadores do que muitos). Nesta figura pode novamente observar-se que parece haver lugar a melhoramento dos desempenhos, caso se possa lidar com maior número de alternativas.

5.8 Algumas conclusões

Os desempenhos mais interessantes, que não são necessariamente os maiores qualquer que seja a situação, foram obtidos pelo programa designado por pp4. Contudo, esses desempenhos 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. No ELECTRE III, tal como sucedeu no método PROMETHEE, os melhores desempenhos são desvalorizáveis por pressuporem que o decisor abdica de conhecer a relação de prevalência difusa. No entanto, é razoável supor que o decisor não terá grande interesse em conhecer os índices que definem essa relação quando o número de alternativas é elevado. Se por exemplo

houver cem alternativas então são necessários $10000 - 100 = 9900$ índices para definir a relação de prevalência difusa, que serão certamente de difícil apresentação no écran do computador e de difícil leitura. Nestes casos, as saídas mais legíveis são as pré-ordens completas das destilações descendente e ascendente.

Apesar de baixos relativamente à outra aplicação, os desempenhos conseguidos na execução do método ELECTRE III tornam-se atraentes por ocorrerem nas situações em que o programa sequencial é mais lento, i.e. problemas com grande número de alternativas. Não se pretende com isto negar a importância dos desempenhos obtidos nas situações com um grande número de critérios (simulação de decisão em grupo), que só não são mais elevados por a memória disponível não ter permitido criar problemas para o banco 2 com um número de alternativas semelhante aos do banco 1.

Um outro facto que contribui para valorizar os desempenhos obtidos é o facto de estes ocorrerem na fase de análise de sensibilidade, que se considera ser a ocasião em que a velocidade de execução é mais importante. Face aos resultados obtidos, tempos de resposta que poderiam ser considerados desapontadores (os obtidos por um programa sequencial) são agradavelmente reduzidos quando se utiliza o programa pp4.

6 - CONCLUSÕES E VIAS DE INVESTIGAÇÃO FUTURA

No âmbito do estudo da aplicação de computadores paralelos, em concreto o multicomputador da Parsytech apresentado na secção 3.6, a métodos de apoio multicritério à decisão, escolheram-se dois dos mais representativos métodos de prevalência: o PROMETHEE e o ELECTRE III. Para cada um destes métodos efectuou-se um estudo autónomo, com muitos pontos em comum. Foram para isso construídos vários programas paralelos para cada método, cada um explorando mais possibilidades de paralelismo que o anterior. A avaliação destes programas efectuou-se através de um conjunto de experiências que, não pretendendo ser exaustivo, fosse suficientemente multifacetado para permitir julgar os seus méritos relativos.

Os problemas de teste, em número não muito grande para não tornar excessiva a quantidade de experiências, possuem dimensões grandes em termos de número de alternativas e número de critérios, limitadas apenas pela memória disponível. A existência de grande número de critérios é questionável, não no aspecto informático mas no aspecto de apoio à decisão. Por exemplo, [Pomerol e Barba-Romero, 1993, p.339] indicam como número máximo razoável para uma situação de decisão cerca de vinte critérios, face a limitações cognitivas do decisor. Por esse motivo, sugeriu-se (especialmente no estudo relativo ao método ELECTRE III) que tais problemas representassem uma situação de decisão em grupo, em que cada actor operacionalizasse um número pequeno de pontos de vista. Esta sugestão não é suportada a nível teórico ou prático, pois tal não caberia no âmbito deste trabalho, mas parece ser merecedora de um estudo aprofundado, até porque os métodos de prevalência remetem facilmente para uma analogia de decisão em grupo, nas quais cada critério representa um decisor (cf. método de Condorcet, sec. 2.3).

Procedeu-se já a uma análise dos resultados, à medida que foram sendo apresentados, tendo-se sumariado algumas das mais importantes conclusões relativas a cada método nas secções 4.8 (PROMETHEE) e 5.8 (ELECTRE III). Neste capítulo apresentam-se algumas conclusões de carácter mais genérico, oriundas de ambos os estudos, incluindo nestas conclusões as perspectivas de investigação futura relativas a este programa de pesquisa.

Comparação dos resultados obtidos para os métodos PROMETHEE e ELECTRE III

Os resultados obtidos permitem concluir que as estratégias de paralelização seguidas obtêm os melhores desempenhos quando o agente de decisão sacrifica a possibilidade de conhecer os índices que definem a relação de prevalência. Nesse caso, a diminuição do tempo de resposta às solicitações do agente de decisão poderá ser bastante apreciável, encorajando a realização de análises de sensibilidade 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, pode argumentar-se que os números que o decisor é impedido 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.

Mostrou-se também que a razão entre os tempos de execução do programa sequencial e dos melhores programas paralelos aumenta em geral com a complexidade 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. Se se estabelecer a "razoabilidade" das dimensões do problema através do tempo necessário para o estudar, torna-se possível redefinir essa razoabilidade ao utilizar o programa paralelo.

Em ambos os estudos, verificou-se que nem todas as possibilidades de paralelização puderam ser eficientemente exploradas, nomeadamente quando a computação envolvida não compensa a quantidade de comunicação entre processadores necessária ao paralelismo. Verificou-se igualmente que a eficiência decrescia com o número de processadores utilizado. Por esse motivo, e face aos tempos de execução obtidos para as dimensões de problema estudadas, que foram limitadas pela memória disponível, parece ser despropositada a utilização de um número de processadores muito superior a dezasseis (isto, claro, considerando o tipo de computador paralelo utilizado).

O método PROMETHEE proporcionou melhores desempenhos e casos de teste de maior dimensão, factores obviamente relacionados, do que o ELECTRE III. De facto, ao contrário das aplicações que executam o PROMETHEE, as aplicações que executam o ELECTRE III não atingiram valores muito altos para o speedup (razão S). Todavia, em contrapartida, no ELECTRE III os tempos de execução parecem ser susceptíveis de melhoramento (caso se consiga testar casos com maior números de alternativas) e foram satisfatórios para as dimensões de problema utilizadas, sobretudo na importante fase de análise de sensibilidade.

Vias para investigação futura

O estudo efectuado, que se crê ser pioneiro na aplicação de processamento paralelo a métodos de prevalência, levanta mais questões do que aquelas que resolve. Encontra-se, de facto, um imenso conjunto de caminhos de investigação a explorar.

Em primeiro lugar, o trabalho efectuado pode ser desenvolvido, quer através da construção de novos programas, quer através da experimentação com maior número de situações ou, se possível, com situações de decisão reais.

Em segundo lugar, o trabalho desenvolvido pode dar origem a verdadeiros 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 empresas ou empreendimentos, 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). No caso de se pretender uma aplicação do tipo BANKADVISER, em que as alternativas são comparadas a um conjunto de alternativas de referência, pode desenvolver-se um sistema de apoio à decisão que também apoie a escolha de um conjunto de alternativas de referência.

Em terceiro lugar, os programas construídos podem ser adaptados por forma a proporcionar maior confiança ao decisor face aos resultados obtidos, utilizando o processamento paralelo para introduzir alguma automatização nas análises de sensibilidade, como é o caso da determinação das gamas de variação admissíveis para alguns parâmetros sem alterar o resultado obtido (regiões de indiferença).

Uma quarta via de investigação é a extensão deste estudo a métodos de prevalência dedicados a outras problemáticas, sobretudo os dedicados à problemática de afectação de categorias a classes definidas *a priori* (p. ex. ELECTRE TRI [Yu, 1992]), nos quais é possível antever algumas situações de decisão com grande número de alternativas.

Por fim, os programas paralelos de melhor desempenho podem ser utilizados para executar com maior celeridade um programa de experiências que vise comparar os resultados obtidos por variantes dos métodos. Pode pretender-se com esse estudo comparar um programa que execute, por exemplo, o ELECTRE III²¹, com um programa que execute uma versão diferente do método, analisando as diferenças entre os resultados do método e entre os seus desempenhos. As ferramentas construídas podem assim contribuir para a aferição das consequências de inovações a nível metodológico.

Como voto final, espera-se que a enumeração das vias para investigação futura contribua para que se explorem as possibilidades que uma abordagem deste tipo potencia, para desenvolvimentos a nível metodológico (como fazer) e prático (o que se torna possível fazer).

²¹ Note-se que na família ELECTRE é comum encontrar várias versões de cada método (veja-se [Maystre et al., 1994]).

APÊNDICE A: RESULTADOS RELATIVOS AO MÉTODO PROMETHEE

Neste apêndice apresentam-se alguns resultados das experiências conduzidas com os programas que executam o método PROMETHEE. Cada quadro é acompanhado de uma ficha como a seguinte, que identifica o seu conteúdo.

| | |
|------------------------------|----------------------------|
| <i>Resultado apresentado</i> | <i>Banco de problemas</i> |
| <i>Programas utilizados</i> | <i>Topologia utilizada</i> |
| <i>Nível de exigência</i> | |

Quadro 1

| | |
|---|----------------|
| Composição do tempo de execução | Banco G |
| Programa sequencial | Um processador |
| Máximo: índices $\pi(a,b)$, pré-ordens do PROMETHEE I e II | |

| | $\pi(a,b)$ | PROMET. I | PROMET. II | $\pi(a,b)$ | PROMET. I | PROMET. II |
|--------------|------------|-----------|------------|------------|-----------|------------|
| C. Normal | G5x5 | | | G50x5 | | |
| | 97.20% | 1.40% | 1.40% | 99.60% | 0.20% | 0.20% |
| C. Abreviado | 97.40% | 1.40% | 1.20% | 96.80% | 1.80% | 1.40% |
| | G5x10 | | | G50x10 | | |
| | 98.20% | 1.10% | 0.70% | 99.80% | 0.10% | 0.10% |
| | 98.30% | 1.00% | 0.70% | 98.20% | 1.10% | 0.70% |
| | G5x50 | | | G50x50 | | |
| | 98.80% | 0.90% | 0.30% | 99.90% | 0.10% | 0.00% |
| | 99.10% | 0.70% | 0.20% | 99.00% | 0.80% | 0.20% |
| | G5x100 | | | G50x100 | | |
| | 98.90% | 0.90% | 0.20% | 99.90% | 0.10% | 0.00% |
| | 99.20% | 0.70% | 0.10% | 99.10% | 0.80% | 0.10% |
| | G5x500 | | | G50x500 | | |
| | 99.10% | 0.90% | 0.00% | 99.90% | 0.10% | 0.00% |
| | 99.30% | 0.70% | 0.00% | 99.20% | 0.80% | 0.00% |
| | G10x5 | G10x5 | G10x5 | G100x5 | | |
| | 99.00% | 0.60% | 0.40% | 99.80% | 0.10% | 0.10% |
| | 97.20% | 1.40% | 1.40% | 96.80% | 1.40% | 1.80% |
| | G10x10 | | | G100x10 | | |
| | 99.10% | 0.60% | 0.30% | 99.90% | 0.10% | 0.00% |
| | 98.30% | 1.00% | 0.70% | 98.20% | 1.00% | 0.80% |
| | G10x50 | | | G100x50 | | |
| | 99.40% | 0.50% | 0.10% | 99.90% | 0.10% | 0.00% |
| | 99.00% | 0.80% | 0.20% | 99.00% | 0.80% | 0.20% |
| | G10x100 | | | G100x100 | | |
| | 99.40% | 0.50% | 0.10% | 99.90% | 0.10% | 0.00% |
| | 99.00% | 0.80% | 0.20% | 99.10% | 0.80% | 0.10% |
| | G10x500 | | | G100x500 | | |
| | 99.50% | 0.50% | 0.00% | 99.90% | 0.10% | 0.00% |
| | 99.20% | 0.80% | 0.00% | 99.20% | 0.80% | 0.00% |

Quadro 2

| | |
|---|--------------------|
| Tempos de execução (em segundos) | Banco G e Banco Gs |
| Programa sequencial | Um processador |
| Máximo: índices $\pi(a,b)$, pré-ordens do PROMETHEE I e II | |

Banco G

| | Nº Critér -> | 5 | 10 | 50 | 100 |
|--------|----------------|--------|---------|---------|----------|
| | Nº Alternat. \ | G5x5 | G10x5 | G50x5 | G100x5 |
| 1ª Vez | 5 | 0.02 | 0.05 | 0.19 | 0.38 |
| 2ª Vez | | 0.02 | 0.05 | 0.19 | 0.38 |
| Abrev | | 0.03 | 0.03 | 0.03 | 0.03 |
| | | G5x10 | G10x10 | G50x10 | G100x10 |
| | 10 | 0.10 | 0.20 | 0.85 | 1.70 |
| | | 0.10 | 0.20 | 0.85 | 1.70 |
| | | 0.12 | 0.12 | 0.12 | 0.12 |
| | | G5x50 | G10x50 | G50x50 | G100x50 |
| | 50 | 2.74 | 5.43 | 22.91 | 45.76 |
| | | 2.74 | 5.43 | 22.91 | 45.76 |
| | | 3.15 | 3.18 | 3.17 | 3.17 |
| | | G5x100 | G10x100 | G50x100 | G100x100 |
| | 100 | 11.07 | 21.90 | 92.46 | 184.85 |
| | | 11.07 | 21.90 | 92.46 | 184.85 |
| | | 12.70 | 12.73 | 12.79 | 12.78 |
| | | G5x500 | G10x500 | G50x500 | G100x500 |
| | 500 | 277.88 | 550.08 | 2326.85 | 4653.29 |
| | | 277.91 | 550.09 | 2326.85 | 4653.26 |
| | | 320.31 | 320.69 | 320.63 | 320.69 |

Banco Gs

| | Nº Critér -> | 5 | 10 | 50 | 100 |
|--------|----------------|---------|----------|----------|-----------|
| | Nº Alternat. \ | Gs5x5 | Gs10x5 | Gs50x5 | Gs100x5 |
| 1ª Vez | 5 | 0.01 | 0.03 | 0.11 | 0.22 |
| 2ª Vez | | 0.01 | 0.03 | 0.11 | 0.22 |
| Abrev | | 0.03 | 0.03 | 0.03 | 0.03 |
| | | Gs5x10 | Gs10x10 | Gs50x10 | Gs100x10 |
| | 10 | 0.06 | 0.11 | 0.50 | 0.94 |
| | | 0.06 | 0.11 | 0.50 | 0.94 |
| | | 0.12 | 0.12 | 0.12 | 0.12 |
| | | Gs5x50 | Gs10x50 | Gs50x50 | Gs100x50 |
| | 50 | 1.50 | 2.97 | 13.41 | 25.19 |
| | | 1.50 | 2.97 | 13.41 | 25.19 |
| | | 3.15 | 3.15 | 3.15 | 3.13 |
| | | Gs5x100 | Gs10x100 | Gs50x100 | Gs100x100 |
| | 100 | 6.06 | 11.98 | 54.13 | 101.66 |
| | | 6.05 | 11.98 | 54.13 | 101.66 |
| | | 12.68 | 12.68 | 12.68 | 12.65 |
| | | Gs5x500 | Gs10x500 | Gs50x500 | Gs100x500 |
| | 500 | 151.93 | 301.08 | 1361.55 | 2558.95 |
| | | 151.92 | 301.08 | 1361.53 | 2558.95 |
| | | 318.35 | 318.38 | 318.85 | 317.89 |

NOTA: “1ª Vez” designa a duração da primeira execução do problema, que inclui leitura do ficheiro; “2ª Vez” corresponde a um cálculo normal sem leitura de ficheiro; “Abrev” corresponde ao cálculo abreviado (após alteração de tipo ou parâmetro de um critério).

Quadro 3

| | |
|---|----------------------|
| Razão S (Speedup) | Banco G e Banco Gs |
| Programa pp1 | Rede matricial 4 x 4 |
| Máximo: índices $\pi(a,b)$, pré-ordens do PROMETHEE I e II | |

| Banco G | | | | | Banco Gs | | | | | | |
|---------|------------|------|------|------|----------|---------|------------|------|------|------|------|
| Crit -> | | 5 | 10 | 50 | 100 | Crit -> | | 5 | 10 | 50 | 100 |
| | Alt \ | | | | | | Alt \ | | | | |
| 1ª Vez | 5 | 0.75 | 0.90 | 0.80 | 0.94 | 1ª Vez | 5 | 0.45 | 0.54 | 0.49 | 0.54 |
| 2ª Vez | | 4.16 | 4.81 | 5.75 | 8.54 | 2ª Vez | | 2.45 | 3.19 | 5.08 | 6.62 |
| Abrev. | | 3.65 | 3.03 | 3.89 | 4.04 | Abrev. | | 2.94 | 3.29 | 3.68 | 3.27 |
| 1ª Vez | 10 | 1.62 | 1.76 | 1.87 | 2.21 | 1ª Vez | 10 | 0.96 | 1.22 | 1.26 | 1.38 |
| 2ª Vez | | 3.59 | 3.94 | 5.97 | 8.22 | 2ª Vez | | 2.23 | 3.73 | 5.57 | 6.63 |
| Abrev. | | 3.57 | 3.99 | 4.42 | 4.62 | Abrev. | | 3.45 | 4.31 | 3.52 | 3.55 |
| 1ª Vez | 50 | 2.63 | 3.30 | 4.54 | 5.55 | 1ª Vez | 50 | 1.73 | 2.31 | 3.40 | 3.90 |
| 2ª Vez | | 3.06 | 3.90 | 5.85 | 7.63 | 2ª Vez | | 2.10 | 2.88 | 4.76 | 5.92 |
| Abrev. | | 3.90 | 3.96 | 4.00 | 3.89 | Abrev. | | 3.56 | 3.70 | 4.10 | 3.71 |
| 1ª Vez | 100 | 2.66 | 3.43 | 5.24 | 6.41 | 1ª Vez | 100 | 2.13 | 2.58 | 4.00 | 4.95 |
| 2ª Vez | | 2.85 | 3.70 | 5.94 | 7.55 | 2ª Vez | | 2.35 | 2.88 | 4.77 | 6.22 |
| Abrev. | | 3.77 | 3.93 | 4.00 | 3.94 | Abrev. | | 3.89 | 3.76 | 3.97 | 3.97 |
| 1ª Vez | 500 | 2.92 | 3.45 | 5.63 | 7.26 | 1ª Vez | 500 | 2.29 | 2.99 | 4.65 | 5.90 |
| 2ª Vez | | 2.96 | 3.49 | 5.76 | 7.52 | 2ª Vez | | 2.33 | 3.06 | 4.83 | 6.18 |
| Abrev. | | 4.00 | 3.98 | 3.97 | 3.94 | Abrev. | | 3.89 | 4.00 | 3.88 | 3.84 |

Quadro 4

| | |
|---|----------------------|
| Razão S (Speedup) | Banco G e Banco Gs |
| Programa pp2 | Rede matricial 4 x 4 |
| Máximo: índices $\pi(a,b)$, pré-ordens do PROMETHEE I e II | |

| Banco G | | | | | Banco Gs | | | | | | |
|---------|-----------|------|------|------|----------|---------|-----------|------|------|------|------|
| Crit -> | | 5 | 10 | 50 | 100 | Crit -> | | 5 | 10 | 50 | 100 |
| | Alt \ | | | | | | Alt \ | | | | |
| 1ª Vez | 5 | 0.51 | 0.70 | 0.80 | 0.90 | 1ª Vez | 5 | 0.29 | 0.42 | 0.52 | 0.54 |
| 2ª Vez | | 1.29 | 2.15 | 3.96 | 5.70 | 2ª Vez | | 0.73 | 1.28 | 2.59 | 3.74 |
| Abrev. | | 1.45 | 1.42 | 1.54 | 1.49 | Abrev. | | 1.35 | 1.30 | 1.28 | 1.40 |
| 1ª Vez | 10 | 1.13 | 1.49 | 1.90 | 2.04 | 1ª Vez | 10 | 0.67 | 0.97 | 1.29 | 1.34 |
| 2ª Vez | | 1.91 | 2.79 | 5.37 | 6.51 | 2ª Vez | | 1.19 | 2.12 | 4.37 | 5.91 |
| Abrev. | | 2.04 | 2.21 | 2.40 | 2.35 | Abrev. | | 2.09 | 2.24 | 2.06 | 2.18 |
| 1ª Vez | 50 | 2.16 | 2.92 | 4.29 | 5.39 | 1ª Vez | 50 | 1.36 | 1.93 | 3.23 | 3.81 |
| 2ª Vez | | 2.38 | 3.35 | 5.39 | 7.22 | 2ª Vez | | 1.54 | 2.29 | 4.40 | 5.65 |
| Abrev. | | 2.89 | 2.93 | 2.98 | 2.92 | Abrev. | | 2.76 | 2.80 | 3.00 | 2.82 |

| | | | | | | | | | | |
|--------|------------|------|------|------|------|------------|------|------|------|------|
| 1ª Vez | 100 | 2.19 | 3.01 | 4.98 | 6.28 | 100 | 1.55 | 2.15 | 3.82 | 4.75 |
| 2ª Vez | | 2.30 | 3.21 | 5.64 | 7.33 | | 1.66 | 2.34 | 4.51 | 5.92 |
| Abrev. | | 2.89 | 3.01 | 3.05 | 3.03 | | 2.94 | 2.89 | 2.98 | 2.97 |
| 1ª Vez | 500 | 2.36 | 3.05 | 5.47 | 7.01 | 500 | 1.67 | 2.44 | 4.25 | 5.59 |
| 2ª Vez | | 2.38 | 3.09 | 5.59 | 7.25 | | 1.69 | 2.47 | 4.40 | 5.88 |
| Abrev. | | 3.06 | 3.05 | 3.04 | 3.05 | | 2.99 | 3.03 | 2.99 | 2.98 |

Quadro 5

| | |
|---|----------------------|
| Razão S (Speedup) | Banco G e Banco Gs |
| Programa pp3 | Rede matricial 4 x 4 |
| Máximo: índices $\pi(a,b)$, pré-ordens do PROMETHEE I e II | |

| | | Banco G | | | | Banco Gs | | | | |
|--------|------------|---------|------|------|------|------------|------|------|------|------|
| | | Crit -> | | | | Crit -> | | | | |
| | | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 | |
| Alt \ | | | | | | | | | | |
| 1ª Vez | 5 | 0.32 | 0.50 | 0.72 | 0.86 | 5 | 0.21 | 0.31 | 0.46 | 0.48 |
| 2ª Vez | | 1.49 | 2.21 | 3.81 | 5.89 | | 0.79 | 1.34 | 3.32 | 3.81 |
| Abrev. | | 1.45 | 1.44 | 1.61 | 1.51 | | 1.46 | 1.52 | 1.45 | 1.51 |
| 1ª Vez | 10 | 0.94 | 1.26 | 1.81 | 1.96 | 10 | 0.50 | 0.84 | 1.22 | 1.28 |
| 2ª Vez | | 2.02 | 2.88 | 5.32 | 6.44 | | 1.19 | 2.22 | 4.68 | 5.81 |
| Abrev. | | 2.14 | 2.29 | 2.41 | 2.36 | | 2.17 | 2.31 | 2.19 | 2.26 |
| 1ª Vez | 50 | 2.13 | 2.89 | 4.26 | 5.35 | 50 | 1.34 | 1.91 | 3.21 | 3.75 |
| 2ª Vez | | 2.40 | 3.35 | 5.38 | 7.26 | | 1.54 | 2.29 | 4.41 | 5.60 |
| Abrev. | | 2.90 | 2.93 | 2.98 | 2.91 | | 2.77 | 2.83 | 3.02 | 2.83 |
| 1ª Vez | 100 | 2.18 | 3.00 | 4.97 | 6.25 | 100 | 1.54 | 2.13 | 3.77 | 4.72 |
| 2ª Vez | | 2.30 | 3.21 | 5.61 | 7.34 | | 1.66 | 2.34 | 4.46 | 5.90 |
| Abrev. | | 2.91 | 3.02 | 3.06 | 3.02 | | 2.95 | 2.90 | 2.99 | 2.97 |
| 1ª Vez | 500 | 2.36 | 3.05 | 5.46 | 7.02 | 500 | 1.67 | 2.44 | 4.33 | 5.59 |
| 2ª Vez | | 2.38 | 3.09 | 5.60 | 7.24 | | 1.69 | 2.48 | 4.47 | 5.87 |
| Abrev. | | 3.07 | 3.04 | 3.04 | 3.06 | | 3.00 | 3.03 | 3.00 | 2.98 |

Quadro 6

| | |
|---|----------------------|
| Razão S (Speedup) | Banco G |
| Programas pp1, pp2 e pp3 | Rede matricial 2 x 2 |
| Máximo: índices $\pi(a,b)$, pré-ordens do PROMETHEE I e II | |

| | <u>pp1</u> | | | | | <u>pp2</u> | | | | | <u>pp3</u> | | | | | | |
|--------|------------|------|------|------|------|------------|------|------|------|------|------------|------|------|---|----|----|-----|
| | Crit | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 |
| | Alt | | | | | | | | | | | | | | | | |
| 1ª Vez | 5 | 0.46 | 0.53 | 0.73 | 0.76 | 0.55 | 0.67 | 0.76 | 0.79 | 0.38 | 0.53 | 0.71 | 0.73 | | | | |
| 2ª Vez | | 1.88 | 2.23 | 3.09 | 3.20 | 1.43 | 1.88 | 2.89 | 3.08 | 1.42 | 1.86 | 2.88 | 3.08 | | | | |
| Abrev | | 1.70 | 2.14 | 1.73 | 2.09 | 1.34 | 1.71 | 1.43 | 1.61 | 1.32 | 1.68 | 1.42 | 1.69 | | | | |
| 1ª Vez | 10 | 0.94 | 1.18 | 1.53 | 1.58 | 1.05 | 1.32 | 1.54 | 1.59 | 0.82 | 1.17 | 1.50 | 1.56 | | | | |
| 2ª Vez | | 2.13 | 2.59 | 3.43 | 3.46 | 1.68 | 2.25 | 3.26 | 3.38 | 1.71 | 2.25 | 3.26 | 3.38 | | | | |
| Abrev | | 2.12 | 2.30 | 2.27 | 2.11 | 1.71 | 1.80 | 1.82 | 1.75 | 1.72 | 1.85 | 1.81 | 1.73 | | | | |
| 1ª Vez | 50 | 1.65 | 2.23 | 3.08 | 3.15 | 1.50 | 2.12 | 2.99 | 3.11 | 1.49 | 2.10 | 2.99 | 3.10 | | | | |
| 2ª Vez | | 1.83 | 2.54 | 3.63 | 3.72 | 1.62 | 2.34 | 3.49 | 3.66 | 1.62 | 2.34 | 3.51 | 3.66 | | | | |
| Abrev | | 2.07 | 2.19 | 2.18 | 2.17 | 1.81 | 1.91 | 1.89 | 1.92 | 1.82 | 1.91 | 1.90 | 1.92 | | | | |
| 1ª Vez | 100 | 1.75 | 2.27 | 3.46 | 3.53 | 1.56 | 2.11 | 3.36 | 3.47 | 1.56 | 2.11 | 3.35 | 3.47 | | | | |
| 2ª Vez | | 1.82 | 2.38 | 3.77 | 3.85 | 1.62 | 2.21 | 3.64 | 3.77 | 1.62 | 2.21 | 3.64 | 3.77 | | | | |
| Abrev | | 2.09 | 2.13 | 2.17 | 2.13 | 1.85 | 1.89 | 1.92 | 1.88 | 1.85 | 1.89 | 1.92 | 1.89 | | | | |
| 1ª Vez | 500 | 1.83 | 2.41 | 3.77 | 3.80 | 1.63 | 2.23 | 3.66 | 3.73 | 1.63 | 2.23 | 3.66 | 3.73 | | | | |
| 2ª Vez | | 1.85 | 2.43 | 3.84 | 3.86 | 1.64 | 2.25 | 3.72 | 3.80 | 1.64 | 2.26 | 3.72 | 3.80 | | | | |
| Abrev | | 2.14 | 2.16 | 2.16 | 2.14 | 1.89 | 1.90 | 1.90 | 1.89 | 1.89 | 1.90 | 1.91 | 1.89 | | | | |

Quadro 7

| | |
|---|----------------------|
| Razão S (Speedup) | Banco G |
| Programas pp1, pp2 e pp3 | Rede matricial 2 x 4 |
| Máximo: índices $\pi(a,b)$, pré-ordens do PROMETHEE I e II | |

| | <u>pp1</u> | | | | | <u>pp2</u> | | | | | <u>pp3</u> | | | | | | |
|--------|------------|------|------|------|------|------------|------|------|------|------|------------|------|------|---|----|----|-----|
| | Crit | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 |
| | Alt | | | | | | | | | | | | | | | | |
| 1ª Vez | 5 | 0.64 | 0.80 | 0.86 | 0.86 | 0.48 | 0.66 | 0.81 | 0.83 | 0.48 | 0.66 | 0.81 | 0.84 | | | | |
| 2ª Vez | | 2.17 | 3.19 | 4.73 | 4.48 | 1.12 | 1.67 | 3.52 | 4.01 | 1.14 | 1.87 | 3.72 | 4.03 | | | | |
| Abrev | | 2.23 | 2.57 | 2.49 | 2.60 | 1.25 | 1.35 | 1.38 | 1.33 | 1.27 | 1.40 | 1.45 | 1.41 | | | | |
| 1ª Vez | 10 | 1.39 | 1.52 | 1.82 | 2.03 | 1.05 | 1.31 | 1.74 | 1.97 | 1.05 | 1.32 | 1.70 | 1.97 | | | | |
| 2ª Vez | | 2.64 | 2.87 | 4.67 | 6.18 | 1.77 | 2.24 | 4.14 | 5.75 | 1.79 | 2.27 | 4.25 | 5.73 | | | | |
| Abrev | | 3.00 | 2.95 | 2.59 | 2.96 | 1.96 | 1.99 | 1.81 | 1.95 | 1.97 | 2.04 | 1.83 | 1.96 | | | | |
| 1ª Vez | 50 | 1.81 | 2.34 | 3.59 | 5.23 | 1.51 | 2.08 | 3.40 | 5.09 | 1.52 | 2.09 | 3.40 | 5.03 | | | | |
| 2ª Vez | | 1.98 | 2.60 | 4.36 | 6.95 | 1.63 | 2.30 | 4.13 | 6.63 | 1.63 | 2.30 | 4.08 | 6.72 | | | | |
| Abrev | | 2.52 | 2.66 | 2.67 | 2.71 | 2.02 | 2.15 | 2.15 | 2.17 | 2.03 | 2.15 | 2.14 | 2.17 | | | | |

| | | | | | | | | | | | | | |
|--------|------------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1ª Vez | 100 | 1.90 | 2.34 | 3.91 | 5.94 | 1.62 | 2.12 | 3.82 | 5.70 | 1.63 | 2.12 | 3.85 | 5.71 |
| 2ª Vez | | 1.97 | 2.45 | 4.29 | 6.85 | 1.68 | 2.22 | 4.18 | 6.55 | 1.68 | 2.22 | 4.22 | 6.52 |
| Abrev | | 2.63 | 2.63 | 2.62 | 2.64 | 2.15 | 2.16 | 2.17 | 2.16 | 2.14 | 2.16 | 2.17 | 2.16 |
| 1ª Vez | 500 | 2.00 | 2.44 | 4.33 | 6.68 | 1.70 | 2.19 | 4.11 | 6.41 | 1.70 | 2.19 | 4.12 | 6.41 |
| 2ª Vez | | 2.01 | 2.47 | 4.41 | 6.88 | 1.71 | 2.21 | 4.18 | 6.60 | 1.71 | 2.21 | 4.20 | 6.60 |
| Abrev | | 2.62 | 2.67 | 2.64 | 2.63 | 2.17 | 2.20 | 2.19 | 2.18 | 2.17 | 2.20 | 2.19 | 2.18 |

Quadro 8

| | |
|---|----------------------|
| Razão S (Speedup) | Banco G |
| Programas pp1, pp2 e pp3 | Rede matricial 4 x 2 |
| Máximo: índices $\pi(a,b)$, pré-ordens do PROMETHEE I e II | |

| | | <u>pp1</u> | | | | <u>pp2</u> | | | | <u>pp3</u> | | | | |
|--------|------------|-------------|------|------|------|------------|------|------|------|------------|------|------|------|-----|
| | | <u>Crit</u> | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 |
| | | <u>Alt</u> | | | | | | | | | | | | |
| 1ª Vez | 5 | 0.72 | 0.80 | 0.84 | 0.87 | 0.53 | 0.67 | 0.80 | 0.85 | 0.54 | 0.67 | 0.80 | 0.84 | |
| 2ª Vez | | 3.27 | 3.30 | 4.28 | 4.93 | 1.44 | 1.87 | 3.54 | 4.38 | 1.49 | 1.95 | 3.57 | 4.41 | |
| Abrev | | 2.35 | 2.47 | 2.56 | 2.86 | 1.39 | 1.38 | 1.47 | 1.53 | 1.37 | 1.37 | 1.48 | 1.55 | |
| 1ª Vez | 10 | 1.41 | 1.56 | 1.80 | 1.95 | 1.08 | 1.33 | 1.71 | 1.91 | 1.10 | 1.35 | 1.72 | 1.92 | |
| 2ª Vez | | 2.81 | 3.01 | 4.54 | 5.65 | 1.82 | 2.32 | 4.00 | 5.28 | 1.89 | 2.37 | 4.15 | 5.30 | |
| Abrev | | 2.94 | 3.13 | 3.19 | 3.22 | 1.99 | 2.06 | 2.07 | 2.11 | 2.03 | 2.15 | 2.19 | 2.17 | |
| 1ª Vez | 50 | 2.08 | 2.74 | 4.56 | 5.40 | 1.75 | 2.45 | 4.33 | 5.24 | 1.75 | 2.45 | 4.33 | 5.24 | |
| 2ª Vez | | 2.30 | 3.11 | 5.84 | 7.30 | 1.90 | 2.75 | 5.49 | 7.01 | 1.91 | 2.76 | 5.45 | 7.01 | |
| Abrev | | 2.81 | 2.96 | 2.96 | 2.85 | 2.32 | 2.40 | 2.40 | 2.33 | 2.32 | 2.40 | 2.40 | 2.34 | |
| 1ª Vez | 100 | 2.11 | 2.69 | 5.18 | 6.23 | 1.81 | 2.44 | 5.01 | 6.03 | 1.82 | 2.44 | 4.98 | 6.05 | |
| 2ª Vez | | 2.21 | 2.84 | 5.84 | 7.27 | 1.89 | 2.57 | 5.68 | 6.99 | 1.89 | 2.57 | 5.62 | 7.02 | |
| Abrev | | 2.72 | 2.99 | 3.04 | 2.89 | 2.29 | 2.48 | 2.52 | 2.41 | 2.29 | 2.48 | 2.52 | 2.42 | |
| 1ª Vez | 500 | 2.29 | 2.83 | 5.74 | 7.23 | 1.95 | 2.53 | 5.42 | 7.01 | 1.95 | 2.55 | 5.41 | 6.96 | |
| 2ª Vez | | 2.31 | 2.87 | 5.85 | 7.49 | 1.97 | 2.56 | 5.58 | 7.20 | 1.97 | 2.57 | 5.51 | 7.24 | |
| Abrev | | 2.97 | 2.93 | 2.96 | 2.94 | 2.48 | 2.45 | 2.47 | 2.47 | 2.48 | 2.46 | 2.47 | 2.47 | |

Nota: Os resultados para a topologia matricial 4 x 4 foram já apresentados neste Apêndice, nos Quadros 3 a 5.

Quadro 9

| | |
|--|----------------------|
| Razão S (Speedup) | Banco G |
| Programas pp1, pp2\R e pp3\R | Rede matricial 4 x 4 |
| Parcial: somente pré-ordens do PROMETHEE I e II (abdica-se dos índices π) | |

| | Crit Alt | <u>pp1</u> | | | | <u>pp2\R</u> | | | | <u>pp3\R</u> | | | |
|--------|-------------|------------|-----|-----|-----|--------------|-----|------|------|--------------|-----|------|------|
| | | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 |
| 1ª Vez | 5 | 0.8 | 0.9 | 0.8 | 0.9 | 0.2 | 0.6 | 0.9 | 0.8 | 0.6 | 0.7 | 0.9 | 0.9 |
| 2ª Vez | | 4.2 | 4.8 | 5.8 | 8.5 | 1.5 | 2.6 | 5.9 | 7.1 | 1.7 | 2.8 | 6.0 | 7.3 |
| Abrev | | 3.7 | 3.0 | 3.9 | 4.0 | 1.9 | 1.7 | 1.8 | 1.7 | 1.9 | 1.9 | 1.9 | 1.8 |
| 1ª Vez | 10 | 1.6 | 1.8 | 1.9 | 2.2 | 1.0 | 1.5 | 2.0 | 2.2 | 1.4 | 1.9 | 2.1 | 2.2 |
| 2ª Vez | | 3.6 | 3.9 | 6.0 | 8.2 | 2.6 | 4.3 | 7.5 | 8.5 | 2.7 | 4.5 | 7.7 | 8.6 |
| Abrev | | 3.6 | 4.0 | 4.4 | 4.6 | 2.6 | 2.9 | 2.5 | 2.7 | 2.7 | 3.1 | 2.6 | 2.8 |
| 1ª Vez | 50 | 2.6 | 3.3 | 4.5 | 5.6 | 3.6 | 5.3 | 7.3 | 7.8 | 3.8 | 5.4 | 7.4 | 7.8 |
| 2ª Vez | | 3.1 | 3.9 | 5.9 | 7.6 | 4.5 | 7.0 | 11.5 | 12.5 | 4.6 | 7.0 | 11.5 | 12.5 |
| Abrev | | 3.9 | 4.0 | 4.0 | 3.9 | 5.0 | 5.2 | 5.3 | 5.1 | 5.1 | 5.2 | 5.4 | 5.1 |
| 1ª Vez | 100 | 2.7 | 3.4 | 5.2 | 6.4 | 4.6 | 6.3 | 9.6 | 10.4 | 4.7 | 6.4 | 9.6 | 10.4 |
| 2ª Vez | | 2.9 | 3.7 | 5.9 | 7.6 | 5.2 | 7.3 | 12.4 | 13.7 | 5.2 | 7.3 | 12.4 | 13.7 |
| Abrev | | 3.8 | 3.9 | 4.0 | 3.9 | 5.3 | 5.5 | 5.6 | 5.7 | 5.3 | 5.5 | 5.6 | 5.8 |
| 1ª Vez | 500 | 2.9 | 3.5 | 5.6 | 7.3 | 5.4 | 7.5 | 11.9 | 13.2 | 5.4 | 7.6 | 11.9 | 13.2 |
| 2ª Vez | | 3.0 | 3.5 | 5.8 | 7.5 | 5.5 | 7.8 | 12.6 | 14.1 | 5.6 | 7.8 | 12.6 | 14.1 |
| Abrev | | 4.0 | 4.0 | 4.0 | 3.9 | 6.0 | 6.0 | 6.1 | 6.1 | 6.0 | 6.0 | 6.1 | 6.1 |

Quadro 10

| | |
|---|----------------------|
| Razão S (Speedup) | Banco G |
| Programas pp1 e pp2\R | Rede matricial 4 x 4 |
| Parcial: somente pré-ordem parcial do PROMETHEE I (abdica-se dos índices π e da pré-ordem completa do PROMETHEE II) | |

| | Crit Alt | <u>pp1</u> | | | | <u>pp2\R</u> | | | |
|--------|-------------|------------|-----|-----|-----|--------------|-----|------|------|
| | | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 |
| 1ª Vez | 5 | 0.4 | 0.6 | 0.8 | 0.9 | 0.6 | 0.7 | 0.9 | 0.9 |
| 2ª Vez | | 4.5 | 4.9 | 6.0 | 8.6 | 1.5 | 2.7 | 5.9 | 7.3 |
| Abrev | | 3.3 | 3.4 | 4.1 | 4.2 | 1.8 | 1.7 | 1.8 | 1.7 |
| 1ª Vez | 10 | 1.0 | 1.4 | 1.9 | 2.1 | 1.4 | 1.9 | 2.1 | 2.2 |
| 2ª Vez | | 3.9 | 3.9 | 6.0 | 8.2 | 2.7 | 4.4 | 7.6 | 8.6 |
| Abrev | | 3.6 | 4.2 | 4.5 | 4.5 | 2.7 | 3.1 | 2.5 | 2.7 |
| 1ª Vez | 50 | 2.6 | 3.3 | 4.5 | 5.5 | 3.8 | 5.4 | 7.4 | 7.8 |
| 2ª Vez | | 3.1 | 3.9 | 5.8 | 7.5 | 4.5 | 7.0 | 11.5 | 12.5 |
| Abrev | | 3.9 | 4.0 | 4.0 | 3.9 | 5.1 | 5.2 | 5.3 | 5.1 |

| | | | | | | | | | |
|--------|------------|-----|-----|-----|-----|-----|-----|------|------|
| 1ª Vez | 100 | 2.7 | 3.4 | 5.2 | 6.5 | 4.7 | 6.3 | 9.6 | 10.4 |
| 2ª Vez | | 2.9 | 3.7 | 6.0 | 7.6 | 5.2 | 7.3 | 12.4 | 13.7 |
| Abrev | | 3.8 | 4.0 | 4.0 | 4.0 | 5.4 | 5.5 | 5.7 | 5.8 |
| 1ª Vez | 500 | 2.9 | 3.4 | 5.8 | 7.3 | 5.4 | 7.6 | 11.9 | 13.3 |
| 2ª Vez | | 3.0 | 3.5 | 5.9 | 7.5 | 5.5 | 7.8 | 12.6 | 14.1 |
| Abrev | | 4.0 | 4.0 | 4.0 | 3.9 | 6.0 | 6.0 | 6.1 | 6.1 |

NOTA: Não se apresentam resultados para o programa pp3, dado que este está vocacionado para a situação patente no quadro seguinte.

Quadro 11

| | |
|---|----------------------|
| Razão S (Speedup) | Banco G |
| Programas pp1, pp2\RI e pp3\RI | Rede matricial 4 x 4 |
| Parcial: somente pré-ordem completa do PROMETHEE II (abdica-se dos índices π e da pré-ordem parcial do PROMETHEE I) | |

| | Crit Alt | <u>pp1</u> | | | | <u>pp2\RI</u> | | | | <u>pp3\RI</u> | | | |
|--------|-------------|------------|-----|-----|-----|---------------|------|------|------|---------------|------|------|------|
| | | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 | 5 | 10 | 50 | 100 |
| 1ª Vez | 5 | 0.5 | 0.7 | 0.8 | 0.9 | 0.6 | 0.8 | 0.9 | 0.9 | 0.6 | 0.8 | 0.9 | 0.9 |
| 2ª Vez | | 4.4 | 4.3 | 6.8 | 8.6 | 2.2 | 3.3 | 7.0 | 7.9 | 2.5 | 3.7 | 6.8 | 7.8 |
| Abrev | | 3.5 | 3.1 | 4.2 | 4.1 | 2.1 | 2.3 | 2.4 | 2.4 | 2.4 | 2.5 | 2.6 | 2.6 |
| 1ª Vez | 10 | 1.2 | 1.5 | 1.9 | 2.1 | 1.8 | 2.1 | 2.2 | 2.3 | 1.7 | 2.1 | 2.2 | 2.3 |
| 2ª Vez | | 3.9 | 4.1 | 6.1 | 8.3 | 4.6 | 6.2 | 8.7 | 9.3 | 3.9 | 5.9 | 8.5 | 9.1 |
| Abrev | | 3.7 | 4.1 | 4.6 | 4.6 | 4.2 | 4.8 | 3.9 | 4.4 | 4.0 | 4.3 | 3.7 | 4.2 |
| 1ª Vez | 50 | 2.6 | 3.3 | 4.5 | 5.5 | 6.6 | 7.9 | 8.3 | 8.3 | 6.4 | 7.7 | 8.2 | 8.3 |
| 2ª Vez | | 3.1 | 4.0 | 5.9 | 7.4 | 9.5 | 11.9 | 13.7 | 13.7 | 9.1 | 11.5 | 13.7 | 13.7 |
| Abrev | | 4.0 | 4.1 | 4.1 | 4.0 | 10.5 | 11.3 | 11.5 | 10.7 | 10.0 | 10.6 | 11.0 | 10.3 |
| 1ª Vez | 100 | 2.6 | 3.5 | 5.3 | 6.5 | 10.1 | 10.0 | 10.9 | 11.2 | 9.9 | 9.9 | 10.9 | 11.2 |
| 2ª Vez | | 2.9 | 3.8 | 6.0 | 7.6 | 12.9 | 12.6 | 14.6 | 15.1 | 12.6 | 12.4 | 14.6 | 15.0 |
| Abrev | | 3.9 | 4.0 | 4.1 | 4.0 | 11.6 | 12.1 | 13.1 | 13.8 | 11.3 | 11.9 | 12.5 | 13.6 |
| 1ª Vez | 500 | 3.0 | 3.5 | 5.8 | 7.3 | 14.0 | 13.4 | 14.3 | 14.5 | 13.9 | 13.3 | 14.2 | 14.5 |
| 2ª Vez | | 3.0 | 3.5 | 5.9 | 7.5 | 14.8 | 14.1 | 15.2 | 15.5 | 14.7 | 14.1 | 15.2 | 15.5 |
| Abrev | | 4.1 | 4.1 | 4.1 | 4.0 | 14.8 | 14.6 | 15.1 | 15.1 | 14.6 | 14.5 | 14.9 | 14.8 |

APÊNDICE B: RESULTADOS RELATIVOS AO MÉTODO ELECTRE III

Neste apêndice apresentam-se alguns resultados das experiências conduzidas com os programas que executam o método ELECTRE III. Cada quadro é acompanhado de uma ficha como a seguinte, que identifica o seu conteúdo.

| | |
|------------------------------|----------------------------|
| <i>Resultado apresentado</i> | <i>Banco de problemas</i> |
| <i>Programas utilizados</i> | <i>Topologia utilizada</i> |
| <i>Nível de exigência</i> | |

Quadro 1

| | |
|--|----------------|
| Tempos de execução (segundos) | Banco 1 |
| Programa sequencial | Um processador |
| Máximo: relação de prevalência + resultado da exploração | |

| Problema | A | B | C | D |
|----------|------|-------|-------|-------|
| 4x16 | 0.1 | 0.2 | 0.1 | 0.1 |
| 4x32 | 0.4 | 0.9 | 0.6 | 0.5 |
| 4x64 | 1.6 | 4.8 | 3.6 | 3.2 |
| 4x128 | 6.4 | 30.3 | 26.9 | 23.8 |
| 4x192 | 14.4 | 96.1 | 87.5 | 83.9 |
| 4x256 | 25.6 | 218.4 | 205.3 | 188.6 |
| 5x16 | 0.1 | 0.2 | 0.1 | 0.1 |
| 5x32 | 0.5 | 0.9 | 0.6 | 0.5 |
| 5x64 | 2.0 | 5.1 | 3.7 | 3.2 |
| 5x128 | 8.0 | 32.6 | 26.9 | 23.5 |
| 5x192 | 18.0 | 102.5 | 87.8 | 80.2 |
| 5x256 | 32.0 | 229.7 | 193.8 | 190.7 |
| 8x16 | 0.2 | 0.2 | 0.1 | 0.1 |
| 8x32 | 0.8 | 1.2 | 0.6 | 0.4 |
| 8x64 | 3.1 | 6.1 | 3.6 | 3.0 |
| 8x128 | 12.6 | 34.1 | 24.4 | 21.1 |
| 8x192 | 28.5 | 99.7 | 77.8 | 69.6 |
| 8x256 | 50.7 | 213.4 | 175.2 | 173.9 |
| 10x16 | 0.2 | 0.3 | 0.1 | 0.0 |
| 10x32 | 1.0 | 1.4 | 0.6 | 0.4 |
| 10x64 | 3.9 | 6.7 | 3.7 | 2.7 |
| 10x128 | 15.8 | 35.5 | 21.8 | 19.7 |
| 10x192 | 35.5 | 100.3 | 70.8 | 65.0 |
| 10x256 | 63.2 | 219.2 | 158.7 | 145.3 |

Problema: número de critérios x número de alternativas.

A: Obtenção da relação da relação de prevalência (cronometragem parcial).

B: Obtenção de uma solução inicial.

C: Análise de sensibilidade a parâmetros de um só critério.

D: Análise de sensibilidade aos parâmetros da função $s(\lambda)$.

Quadro 2

| | |
|--|----------------|
| Tempos de execução (segundos) | Banco 2 |
| Programa sequencial | Um processador |
| Máximo: relação de prevalência + resultado da exploração | |

| Problema | A | B | C | D | Problema | A | B | C | D |
|----------|------|------|-----|-----|----------|------|------|-----|-----|
| 4x2x16 | 0.2 | 0.2 | 0.1 | 0.1 | 8x2x16 | 0.4 | 0.4 | 0.1 | 0.0 |
| 4x2x20 | 0.3 | 0.4 | 0.2 | 0.1 | 8x2x20 | 0.6 | 0.7 | 0.2 | 0.1 |
| 4x2x24 | 0.4 | 0.6 | 0.3 | 0.2 | 8x2x24 | 0.9 | 1.0 | 0.3 | 0.2 |
| 4x2x28 | 0.6 | 0.9 | 0.4 | 0.3 | 8x2x28 | 1.2 | 1.4 | 0.5 | 0.3 |
| 4x2x32 | 0.8 | 1.2 | 0.5 | 0.3 | 8x2x32 | 1.5 | 1.9 | 0.6 | 0.3 |
| 4x2x50 | 1.9 | 3.3 | 1.8 | 1.3 | 8x2x50 | 3.8 | 5.0 | 1.8 | 1.1 |
| 4x5x16 | 0.5 | 0.5 | 0.1 | 0.1 | 8x5x16 | 0.9 | 1.0 | 0.2 | 0.1 |
| 4x5x20 | 0.7 | 0.8 | 0.2 | 0.1 | 8x5x20 | 1.5 | 1.5 | 0.3 | 0.1 |
| 4x5x24 | 1.1 | 1.2 | 0.3 | 0.2 | 8x5x24 | 2.1 | 2.3 | 0.5 | 0.1 |
| 4x5x28 | 1.5 | 1.7 | 0.5 | 0.2 | 8x5x28 | 2.9 | 3.1 | 0.6 | 0.2 |
| 4x5x32 | 1.9 | 2.3 | 0.7 | 0.3 | 8x5x32 | 3.8 | 4.1 | 0.9 | 0.3 |
| 4x5x50 | 4.7 | 6.0 | 2.2 | 1.3 | 8x5x50 | 9.4 | 10.3 | 2.3 | 0.9 |
| 4x10x16 | 0.9 | 1.0 | 0.2 | 0.1 | 8x10x16 | 1.8 | 1.9 | 0.3 | 0.0 |
| 4x10x20 | 1.5 | 1.6 | 0.3 | 0.1 | 8x10x20 | 2.9 | 3.0 | 0.5 | 0.1 |
| 4x10x24 | 2.1 | 2.3 | 0.5 | 0.2 | 8x10x24 | 4.2 | 4.4 | 0.7 | 0.1 |
| 4x10x28 | 2.9 | 3.2 | 0.7 | 0.3 | 8x10x28 | 5.8 | 6.0 | 1.0 | 0.2 |
| 4x10x32 | 3.8 | 4.2 | 1.0 | 0.4 | 8x10x32 | 7.6 | 7.9 | 1.3 | 0.3 |
| 4x10x50 | 9.4 | 10.8 | 2.9 | 1.5 | 8x10x50 | 18.8 | 19.7 | 3.6 | 0.9 |
| 5x2x16 | 0.2 | 0.3 | 0.1 | 0.1 | 10x2x16 | 0.5 | 0.5 | 0.1 | 0.1 |
| 5x2x20 | 0.4 | 0.5 | 0.2 | 0.1 | 10x2x20 | 0.7 | 0.8 | 0.2 | 0.1 |
| 5x2x24 | 0.5 | 0.7 | 0.3 | 0.2 | 10x2x24 | 1.1 | 1.2 | 0.3 | 0.1 |
| 5x2x28 | 0.7 | 1.0 | 0.4 | 0.3 | 10x2x28 | 1.5 | 1.7 | 0.5 | 0.2 |
| 5x2x32 | 1.0 | 1.3 | 0.6 | 0.3 | 10x2x32 | 1.9 | 2.2 | 0.6 | 0.3 |
| 5x2x50 | 2.4 | 3.8 | 1.9 | 1.5 | 10x2x50 | 4.7 | 5.9 | 1.9 | 1.0 |
| 5x5x16 | 0.6 | 0.6 | 0.1 | 0.1 | 10x5x16 | 1.2 | 1.2 | 0.2 | 0.0 |
| 5x5x20 | 0.9 | 1.0 | 0.3 | 0.1 | 10x5x20 | 1.8 | 1.9 | 0.3 | 0.1 |
| 5x5x24 | 1.3 | 1.5 | 0.4 | 0.2 | 10x5x24 | 2.7 | 2.7 | 0.5 | 0.1 |
| 5x5x28 | 1.8 | 2.1 | 0.6 | 0.3 | 10x5x28 | 3.6 | 3.8 | 0.7 | 0.2 |
| 5x5x32 | 2.4 | 2.7 | 0.8 | 0.4 | 10x5x32 | 4.8 | 5.0 | 0.9 | 0.2 |
| 5x5x50 | 5.9 | 7.1 | 2.2 | 1.3 | 10x5x50 | 11.7 | 12.5 | 2.4 | 0.7 |
| 5x10x16 | 1.2 | 1.2 | 0.2 | 0.1 | 10x10x16 | 2.3 | 2.3 | 0.3 | 0.0 |
| 5x10x20 | 1.8 | 1.9 | 0.4 | 0.1 | 10x10x20 | 3.6 | 3.7 | 0.6 | 0.0 |
| 5x10x24 | 2.7 | 2.8 | 0.6 | 0.2 | 10x10x24 | 5.3 | 5.4 | 0.8 | 0.1 |
| 5x10x28 | 3.6 | 3.9 | 0.8 | 0.2 | 10x10x28 | 7.3 | 7.4 | 1.1 | 0.1 |
| 5x10x32 | 4.8 | 5.1 | 1.1 | 0.4 | 10x10x32 | 9.5 | 9.7 | 1.5 | 0.2 |
| 5x10x50 | 11.8 | 12.9 | 2.8 | 1.2 | 10x10x50 | 23.5 | 24.0 | 3.8 | 0.4 |

Problema: número de pontos de vista x número de instâncias x número de alternativas.

(Nota: número de critérios = número de pontos de vista x número de instâncias)

A: Obtenção da relação da relação de prevalência (cronometragem parcial).

B: Obtenção de uma solução inicial.

C: Análise de sensibilidade a parâmetros de um só critério.

D: Análise de sensibilidade aos parâmetros da função $s(\lambda)$.

Quadro 3

| | Problema | B | C | D | Problema | B | C | D |
|-------------------------------------|----------|------|------|------|----------|------|------|------|
| | 4x16 | 1.12 | 1.06 | 1.05 | 8x16 | 1.53 | 1.19 | 1.05 |
| Razão S | 4x32 | 1.14 | 1.04 | 1.02 | 8x32 | 1.58 | 1.13 | 1.03 |
| Banco 1 | 4x64 | 1.11 | 1.02 | 1.01 | 8x64 | 1.44 | 1.08 | 1.01 |
| Programa pp1 | 4x128 | 1.07 | 1.01 | 1.01 | 8x128 | 1.29 | 1.05 | 1.01 |
| | 4x192 | 1.05 | 1.01 | 1.00 | 8x192 | 1.21 | 1.03 | 1.00 |
| Rede 4 x 4 | 4x256 | 1.04 | 1.01 | 1.00 | 8x256 | 1.17 | 1.03 | 1.00 |
| | 5x16 | 1.22 | 1.09 | 1.05 | 10x16 | 1.70 | 1.28 | 1.05 |
| Parcial: só resultado da exploração | 5x32 | 1.25 | 1.06 | 1.03 | 10x32 | 1.75 | 1.18 | 1.03 |
| | 5x64 | 1.20 | 1.04 | 1.01 | 10x64 | 1.62 | 1.11 | 1.01 |
| | 5x128 | 1.12 | 1.02 | 1.01 | 10x128 | 1.41 | 1.07 | 1.01 |
| | 5x192 | 1.08 | 1.01 | 1.00 | 10x192 | 1.31 | 1.05 | 1.00 |
| | 5x256 | 1.06 | 1.01 | 1.00 | 10x256 | 1.24 | 1.04 | 1.00 |

Quadro 4

| | | | | | | | | |
|-------------------------------------|---------|------|------|----------|---------|------|------|------|
| | 4x2x16 | 1.73 | 1.20 | 1.05 | 8x2x16 | 2.50 | 1.49 | 1.05 |
| | 4x2x20 | 1.68 | 1.21 | 1.04 | 8x2x20 | 2.49 | 1.46 | 1.04 |
| Razão S | 4x2x24 | 1.67 | 1.18 | 1.03 | 8x2x24 | 2.54 | 1.42 | 1.04 |
| Banco 2 | 4x2x28 | 1.65 | 1.15 | 1.03 | 8x2x28 | 2.47 | 1.34 | 1.03 |
| Programa pp1 | 4x2x32 | 1.64 | 1.14 | 1.03 | 8x2x32 | 2.50 | 1.34 | 1.03 |
| | 4x2x50 | 1.53 | 1.11 | 1.02 | 8x2x50 | 2.28 | 1.27 | 1.02 |
| Rede 4 x 4 | 4x5x16 | 3.17 | 1.59 | 1.05 | 8x5x16 | 4.34 | 1.97 | 1.05 |
| | 4x5x20 | 2.95 | 1.58 | 1.04 | 8x5x20 | 4.13 | 2.01 | 1.04 |
| Parcial: só resultado da exploração | 4x5x24 | 3.07 | 1.54 | 1.03 | 8x5x24 | 4.20 | 1.96 | 1.03 |
| | 4x5x28 | 3.02 | 1.46 | 1.03 | 8x5x28 | 4.47 | 1.89 | 1.03 |
| | 4x5x32 | 2.97 | 1.44 | 1.03 | 8x5x32 | 4.58 | 1.83 | 1.02 |
| | 4x5x50 | 2.59 | 1.29 | 1.02 | 8x5x50 | 4.11 | 1.74 | 1.02 |
| | 4x10x16 | 4.48 | 1.95 | 1.05 | 8x10x16 | 6.15 | 2.77 | 1.05 |
| | 4x10x20 | 4.08 | 1.92 | 1.04 | 8x10x20 | 5.32 | 2.63 | 1.04 |
| | 4x10x24 | 4.14 | 1.83 | 1.03 | 8x10x24 | 5.80 | 2.62 | 1.03 |
| | 4x10x28 | 4.27 | 1.72 | 1.03 | 8x10x28 | 6.04 | 2.41 | 1.03 |
| | 4x10x32 | 4.24 | 1.64 | 1.03 | 8x10x32 | 6.36 | 2.37 | 1.03 |
| | 4x10x50 | 3.70 | 1.50 | 1.02 | 8x10x50 | 5.64 | 2.09 | 1.02 |
| | 5x2x16 | 1.90 | 1.26 | 1.05 | 10x2x16 | 2.74 | 1.56 | 1.04 |
| | 5x2x20 | 1.83 | 1.27 | 1.04 | 10x2x20 | 2.75 | 1.61 | 1.04 |
| | 5x2x24 | 1.88 | 1.24 | 1.03 | 10x2x24 | 2.81 | 1.50 | 1.03 |
| | 5x2x28 | 1.85 | 1.20 | 1.03 | 10x2x28 | 2.87 | 1.46 | 1.03 |
| | 5x2x32 | 1.88 | 1.18 | 1.03 | 10x2x32 | 2.89 | 1.45 | 1.03 |
| | 5x2x50 | 1.68 | 1.13 | 1.02 | 10x2x50 | 2.59 | 1.34 | 1.02 |
| 5x5x16 | 3.43 | 1.70 | 1.05 | 10x5x16 | 5.05 | 2.48 | 1.05 | |
| 5x5x20 | 3.32 | 1.66 | 1.04 | 10x5x20 | 4.53 | 2.37 | 1.04 | |
| 5x5x24 | 3.42 | 1.57 | 1.03 | 10x5x24 | 4.88 | 2.30 | 1.03 | |
| 5x5x28 | 3.28 | 1.51 | 1.03 | 10x5x28 | 5.03 | 2.10 | 1.03 | |
| 5x5x32 | 3.43 | 1.46 | 1.02 | 10x5x32 | 5.15 | 2.05 | 1.02 | |
| 5x5x50 | 2.95 | 1.38 | 1.02 | 10x5x50 | 4.81 | 1.95 | 1.02 | |
| 5x10x16 | 4.94 | 2.04 | 1.04 | 10x10x16 | 6.73 | 3.18 | 1.03 | |
| 5x10x20 | 4.44 | 2.01 | 1.04 | 10x10x20 | 5.80 | 3.11 | 1.03 | |
| 5x10x24 | 4.74 | 2.02 | 1.03 | 10x10x24 | 6.36 | 3.05 | 1.03 | |
| 5x10x28 | 4.70 | 1.85 | 1.03 | 10x10x28 | 6.81 | 2.89 | 1.03 | |
| 5x10x32 | 4.89 | 1.82 | 1.02 | 10x10x32 | 7.27 | 2.84 | 1.02 | |
| 5x10x50 | 4.38 | 1.72 | 1.02 | 10x10x50 | 6.60 | 2.62 | 1.02 | |

NOTA: caso se pretendesse conhecer a relação de prevalência o tempo de execução não se alteraria

Quadro 5

| | Problema | B | C | D | Problema | B | C | D |
|-------------------------------------|----------|------|------|------|----------|------|------|------|
| | 4x16 | 0.58 | 0.44 | 0.36 | 8x16 | 0.90 | 0.50 | 0.34 |
| Razão S | 4x32 | 1.00 | 0.86 | 0.86 | 8x32 | 1.35 | 0.89 | 0.78 |
| Banco 1 | 4x64 | 1.45 | 1.37 | 1.50 | 8x64 | 1.85 | 1.46 | 1.49 |
| Programa pp2 | 4x128 | 1.82 | 1.85 | 1.99 | 8x128 | 2.16 | 1.92 | 2.03 |
| | 4x192 | 1.98 | 2.02 | 2.14 | 8x192 | 2.26 | 2.09 | 2.19 |
| Rede 4 x 4 | 4x256 | 2.07 | 2.11 | 2.21 | 8x256 | 2.33 | 2.19 | 2.28 |
| | 5x16 | 0.63 | 0.47 | 0.36 | 10x16 | 1.10 | 0.60 | 0.38 |
| Parcial: só resultado da exploração | 5x32 | 1.09 | 0.90 | 0.83 | 10x32 | 1.54 | 0.99 | 0.82 |
| | 5x64 | 1.55 | 1.41 | 1.48 | 10x64 | 2.03 | 1.49 | 1.50 |
| | 5x128 | 1.92 | 1.89 | 2.02 | 10x128 | 2.32 | 1.94 | 2.05 |
| | 5x192 | 2.07 | 2.07 | 2.19 | 10x192 | 2.41 | 2.12 | 2.23 |
| | 5x256 | 2.14 | 2.14 | 2.25 | 10x256 | 2.44 | 2.22 | 2.31 |

Quadro 6

| | | | | | | | | |
|-------------------------------------|---------|------|------|----------|---------|------|------|------|
| | 4x2x16 | 1.01 | 0.54 | 0.37 | 8x2x16 | 1.65 | 0.71 | 0.37 |
| | 4x2x20 | 1.16 | 0.68 | 0.49 | 8x2x20 | 1.75 | 0.84 | 0.49 |
| Razão S | 4x2x24 | 1.24 | 0.78 | 0.60 | 8x2x24 | 2.00 | 0.96 | 0.61 |
| Banco 2 | 4x2x28 | 1.41 | 0.90 | 0.76 | 8x2x28 | 2.13 | 1.05 | 0.78 |
| Programa pp2 | 4x2x32 | 1.48 | 0.96 | 0.85 | 8x2x32 | 2.21 | 1.16 | 0.86 |
| | 4x2x50 | 1.73 | 1.28 | 1.23 | 8x2x50 | 2.56 | 1.44 | 1.23 |
| Rede 4 x 4 | 4x5x16 | 2.06 | 0.79 | 0.38 | 8x5x16 | 3.35 | 1.18 | 0.44 |
| | 4x5x20 | 2.16 | 0.91 | 0.50 | 8x5x20 | 3.60 | 1.51 | 0.61 |
| Parcial: só resultado da exploração | 4x5x24 | 2.43 | 1.02 | 0.61 | 8x5x24 | 3.88 | 1.65 | 0.72 |
| | 4x5x28 | 2.53 | 1.10 | 0.73 | 8x5x28 | 4.22 | 1.69 | 0.87 |
| | 4x5x32 | 2.63 | 1.19 | 0.83 | 8x5x32 | 4.53 | 1.80 | 0.99 |
| | 4x5x50 | 2.92 | 1.56 | 1.31 | 8x5x50 | 4.58 | 2.10 | 1.42 |
| | 4x10x16 | 3.11 | 1.01 | 0.36 | 8x10x16 | 5.29 | 1.90 | 0.44 |
| | 4x10x20 | 3.24 | 1.27 | 0.49 | 8x10x20 | 4.88 | 2.10 | 0.61 |
| | 4x10x24 | 3.47 | 1.36 | 0.66 | 8x10x24 | 5.48 | 2.26 | 0.70 |
| | 4x10x28 | 3.81 | 1.44 | 0.74 | 8x10x28 | 5.90 | 2.28 | 0.87 |
| | 4x10x32 | 4.01 | 1.52 | 0.86 | 8x10x32 | 6.29 | 2.30 | 0.98 |
| | 4x10x50 | 4.18 | 1.84 | 1.34 | 8x10x50 | 6.04 | 2.45 | 1.37 |
| | 5x2x16 | 1.13 | 0.58 | 0.37 | 10x2x16 | 2.13 | 0.88 | 0.45 |
| | 5x2x20 | 1.28 | 0.76 | 0.51 | 10x2x20 | 2.29 | 1.01 | 0.50 |
| | 5x2x24 | 1.40 | 0.83 | 0.62 | 10x2x24 | 2.34 | 1.07 | 0.63 |
| | 5x2x28 | 1.55 | 0.95 | 0.73 | 10x2x28 | 2.59 | 1.20 | 0.78 |
| | 5x2x32 | 1.67 | 1.04 | 0.83 | 10x2x32 | 2.72 | 1.30 | 0.89 |
| | 5x2x50 | 1.90 | 1.36 | 1.32 | 10x2x50 | 2.93 | 1.61 | 1.33 |
| 5x5x16 | 2.29 | 0.88 | 0.39 | 10x5x16 | 4.53 | 1.85 | 0.53 | |
| 5x5x20 | 2.59 | 1.06 | 0.55 | 10x5x20 | 4.22 | 1.95 | 0.71 | |
| 5x5x24 | 2.78 | 1.17 | 0.68 | 10x5x24 | 4.62 | 2.00 | 0.77 | |
| 5x5x28 | 2.90 | 1.26 | 0.80 | 10x5x28 | 4.99 | 2.05 | 0.98 | |
| 5x5x32 | 3.25 | 1.39 | 0.96 | 10x5x32 | 5.22 | 2.09 | 1.06 | |
| 5x5x50 | 3.35 | 1.64 | 1.32 | 10x5x50 | 5.19 | 2.28 | 1.42 | |
| 5x10x16 | 3.53 | 1.04 | 0.40 | 10x10x16 | 6.46 | 2.78 | 0.60 | |
| 5x10x20 | 3.63 | 1.31 | 0.53 | 10x10x20 | 5.75 | 3.01 | 0.77 | |
| 5x10x24 | 4.01 | 1.46 | 0.65 | 10x10x24 | 6.21 | 2.81 | 0.75 | |
| 5x10x28 | 4.26 | 1.54 | 0.78 | 10x10x28 | 6.70 | 2.76 | 0.86 | |
| 5x10x32 | 4.53 | 1.60 | 0.89 | 10x10x32 | 7.31 | 2.88 | 1.08 | |
| 5x10x50 | 4.76 | 2.00 | 1.30 | 10x10x50 | 6.91 | 2.92 | 1.28 | |

NOTA: caso se pretendesse conhecer a relação de prevalência o tempo de execução não se alteraria

Quadro 7

| | Problema | B | C | D | Problema | B | C | D |
|-------------------------------------|-----------------|----------|----------|----------|-----------------|----------|----------|----------|
| | 4x16 | 0.44 | 0.32 | 0.23 | 8x16 | 0.70 | 0.37 | 0.22 |
| Razão S | 4x32 | 0.93 | 0.79 | 0.67 | 8x32 | 1.19 | 0.77 | 0.57 |
| Banco 1 | 4x64 | 1.87 | 1.90 | 1.78 | 8x64 | 2.17 | 1.94 | 1.63 |
| Programa pp3 | 4x128 | 3.47 | 4.30 | 4.17 | 8x128 | 3.53 | 4.03 | 3.86 |
| | 4x192 | 4.61 | 5.79 | 5.77 | 8x192 | 4.46 | 5.49 | 5.39 |
| Rede 4 x 4 | 4x256 | 5.57 | 6.94 | 6.89 | 8x256 | 5.28 | 6.62 | 6.62 |
| | 5x16 | 0.46 | 0.33 | 0.22 | 10x16 | 0.88 | 0.46 | 0.24 |
| Parcial: só resultado da exploração | 5x32 | 1.01 | 0.82 | 0.63 | 10x32 | 1.34 | 0.89 | 0.59 |
| | 5x64 | 1.93 | 1.91 | 1.66 | 10x64 | 2.28 | 1.89 | 1.61 |
| | 5x128 | 3.54 | 4.21 | 3.99 | 10x128 | 3.65 | 4.03 | 3.79 |
| | 5x192 | 4.72 | 5.96 | 5.82 | 10x192 | 4.60 | 5.57 | 5.47 |
| | 5x256 | 5.51 | 6.83 | 6.82 | 10x256 | 5.35 | 6.70 | 6.66 |

Quadro 8

| | | | | | | | | |
|-------------------------------------|---------|------|------|----------|---------|------|------|------|
| | 4x2x16 | 0.78 | 0.41 | 0.24 | 8x2x16 | 1.30 | 0.55 | 0.23 |
| | 4x2x20 | 0.92 | 0.54 | 0.33 | 8x2x20 | 1.34 | 0.65 | 0.33 |
| Razão S | 4x2x24 | 1.01 | 0.64 | 0.41 | 8x2x24 | 1.61 | 0.80 | 0.42 |
| Banco 2 | 4x2x28 | 1.27 | 0.81 | 0.57 | 8x2x28 | 1.80 | 0.94 | 0.57 |
| Programa pp3 | 4x2x32 | 1.38 | 0.90 | 0.65 | 8x2x32 | 1.92 | 1.11 | 0.65 |
| | 4x2x50 | 1.80 | 1.47 | 1.15 | 8x2x50 | 2.52 | 1.65 | 1.13 |
| Rede 4 x 4 | 4x5x16 | 1.57 | 0.62 | 0.25 | 8x5x16 | 2.63 | 0.99 | 0.29 |
| | 4x5x20 | 1.62 | 0.71 | 0.34 | 8x5x20 | 2.64 | 1.32 | 0.43 |
| Parcial: só resultado da exploração | 4x5x24 | 1.92 | 0.85 | 0.42 | 8x5x24 | 3.06 | 1.59 | 0.53 |
| | 4x5x28 | 2.08 | 0.97 | 0.52 | 8x5x28 | 3.54 | 1.82 | 0.67 |
| | 4x5x32 | 2.26 | 1.13 | 0.62 | 8x5x32 | 3.98 | 2.09 | 0.79 |
| | 4x5x50 | 2.83 | 1.93 | 1.34 | 8x5x50 | 4.04 | 2.91 | 1.45 |
| | 4x10x16 | 2.34 | 0.80 | 0.23 | 8x10x16 | 4.01 | 1.78 | 0.28 |
| | 4x10x20 | 2.32 | 1.03 | 0.32 | 8x10x20 | 3.39 | 1.91 | 0.43 |
| | 4x10x24 | 2.67 | 1.20 | 0.47 | 8x10x24 | 4.04 | 2.30 | 0.51 |
| | 4x10x28 | 3.09 | 1.38 | 0.55 | 8x10x28 | 4.67 | 2.71 | 0.68 |
| | 4x10x32 | 3.38 | 1.54 | 0.67 | 8x10x32 | 5.17 | 2.84 | 0.79 |
| | 4x10x50 | 3.69 | 2.33 | 1.38 | 8x10x50 | 4.77 | 3.35 | 1.36 |
| | 5x2x16 | 0.87 | 0.44 | 0.23 | 10x2x16 | 1.74 | 0.72 | 0.30 |
| | 5x2x20 | 1.01 | 0.61 | 0.35 | 10x2x20 | 1.80 | 0.82 | 0.33 |
| | 5x2x24 | 1.13 | 0.69 | 0.44 | 10x2x24 | 1.89 | 0.91 | 0.43 |
| | 5x2x28 | 1.33 | 0.85 | 0.53 | 10x2x28 | 2.20 | 1.10 | 0.56 |
| | 5x2x32 | 1.52 | 1.00 | 0.63 | 10x2x32 | 2.43 | 1.29 | 0.67 |
| | 5x2x50 | 1.97 | 1.63 | 1.34 | 10x2x50 | 2.82 | 1.93 | 1.25 |
| | 5x5x16 | 1.74 | 0.69 | 0.25 | 10x5x16 | 3.66 | 1.90 | 0.36 |
| | 5x5x20 | 1.94 | 0.85 | 0.38 | 10x5x20 | 3.05 | 1.77 | 0.49 |
| | 5x5x24 | 2.18 | 1.02 | 0.49 | 10x5x24 | 3.58 | 2.02 | 0.56 |
| | 5x5x28 | 2.42 | 1.20 | 0.60 | 10x5x28 | 4.14 | 2.44 | 0.77 |
| 5x5x32 | 2.90 | 1.46 | 0.80 | 10x5x32 | 4.47 | 2.63 | 0.87 | |
| 5x5x50 | 3.20 | 2.00 | 1.32 | 10x5x50 | 4.39 | 3.26 | 1.47 | |
| 5x10x16 | 2.60 | 0.80 | 0.26 | 10x10x16 | 5.08 | 3.53 | 0.42 | |
| 5x10x20 | 2.54 | 1.02 | 0.35 | 10x10x20 | 3.94 | 3.15 | 0.58 | |
| 5x10x24 | 2.98 | 1.24 | 0.46 | 10x10x24 | 4.54 | 3.25 | 0.57 | |
| 5x10x28 | 3.38 | 1.47 | 0.58 | 10x10x28 | 5.20 | 3.69 | 0.66 | |
| 5x10x32 | 3.67 | 1.59 | 0.70 | 10x10x32 | 5.91 | 4.40 | 0.88 | |
| 5x10x50 | 4.01 | 2.48 | 1.26 | 10x10x50 | 5.21 | 4.48 | 1.30 | |

Quadro 9

| | Problema | B | C | D | Problema | B | C | D |
|-------------------------------------|----------|------|------|------|----------|------|------|------|
| | 4x16 | 0.50 | 0.37 | 0.28 | 8x16 | 0.79 | 0.45 | 0.25 |
| Razão S | 4x32 | 1.08 | 0.95 | 0.81 | 8x32 | 1.41 | 0.88 | 0.66 |
| Banco 1 | 4x64 | 2.14 | 2.20 | 2.09 | 8x64 | 2.50 | 2.23 | 1.95 |
| Programa pp4 | 4x128 | 3.94 | 4.97 | 4.88 | 8x128 | 3.94 | 4.60 | 4.46 |
| | 4x192 | 5.22 | 6.64 | 6.68 | 8x192 | 4.94 | 6.29 | 6.20 |
| Rede 4 x 4 | 4x256 | 6.30 | 7.99 | 7.86 | 8x256 | 5.73 | 7.34 | 7.43 |
| | 5x16 | 0.53 | 0.39 | 0.26 | 10x16 | 0.98 | 0.55 | 0.28 |
| Parcial: só resultado da exploração | 5x32 | 1.17 | 0.96 | 0.73 | 10x32 | 1.51 | 1.07 | 0.70 |
| | 5x64 | 2.21 | 2.26 | 2.00 | 10x64 | 2.59 | 2.21 | 1.85 |
| | 5x128 | 3.94 | 4.81 | 4.54 | 10x128 | 3.99 | 4.46 | 4.27 |
| | 5x192 | 5.20 | 6.70 | 6.56 | 10x192 | 4.94 | 6.07 | 6.00 |
| | 5x256 | 6.15 | 7.72 | 7.75 | 10x256 | 5.78 | 7.37 | 7.32 |

Quadro 10

| | | | | | | | | |
|-------------------------------------|---------|------|------|----------|----------|------|------|------|
| | 4x2x16 | 0.89 | 0.49 | 0.28 | 8x2x16 | 1.48 | 0.65 | 0.27 |
| | 4x2x20 | 1.05 | 0.63 | 0.39 | 8x2x20 | 1.53 | 0.76 | 0.38 |
| Razão S | 4x2x24 | 1.17 | 0.76 | 0.49 | 8x2x24 | 1.80 | 0.91 | 0.49 |
| Banco 2 | 4x2x28 | 1.45 | 0.97 | 0.68 | 8x2x28 | 2.02 | 1.12 | 0.69 |
| Programa pp4 | 4x2x32 | 1.59 | 1.09 | 0.76 | 8x2x32 | 2.18 | 1.31 | 0.78 |
| | 4x2x50 | 2.10 | 1.73 | 1.36 | 8x2x50 | 2.93 | 1.96 | 1.37 |
| Rede 4 x 4 | 4x5x16 | 1.81 | 0.74 | 0.29 | 8x5x16 | 2.91 | 1.13 | 0.33 |
| | 4x5x20 | 1.84 | 0.81 | 0.39 | 8x5x20 | 3.03 | 1.50 | 0.49 |
| Parcial: só resultado da exploração | 4x5x24 | 2.20 | 0.99 | 0.49 | 8x5x24 | 3.48 | 1.74 | 0.58 |
| | 4x5x28 | 2.39 | 1.11 | 0.60 | 8x5x28 | 4.00 | 2.03 | 0.75 |
| | 4x5x32 | 2.61 | 1.26 | 0.72 | 8x5x32 | 4.53 | 2.36 | 0.90 |
| | 4x5x50 | 3.19 | 2.20 | 1.52 | 8x5x50 | 4.66 | 3.30 | 1.73 |
| | 4x10x16 | 2.65 | 0.91 | 0.27 | 8x10x16 | 4.51 | 1.91 | 0.29 |
| | 4x10x20 | 2.70 | 1.15 | 0.38 | 8x10x20 | 3.87 | 1.99 | 0.45 |
| | 4x10x24 | 3.08 | 1.41 | 0.56 | 8x10x24 | 4.70 | 2.54 | 0.59 |
| | 4x10x28 | 3.53 | 1.57 | 0.63 | 8x10x28 | 5.43 | 3.03 | 0.79 |
| | 4x10x32 | 3.98 | 1.81 | 0.78 | 8x10x32 | 5.99 | 3.17 | 0.95 |
| | 4x10x50 | 4.31 | 2.66 | 1.65 | 8x10x50 | 5.51 | 3.62 | 1.58 |
| | 5x2x16 | 1.01 | 0.51 | 0.27 | 10x2x16 | 1.88 | 0.85 | 0.35 |
| | 5x2x20 | 1.17 | 0.65 | 0.39 | 10x2x20 | 1.96 | 0.89 | 0.37 |
| | 5x2x24 | 1.31 | 0.82 | 0.52 | 10x2x24 | 2.15 | 1.03 | 0.49 |
| | 5x2x28 | 1.53 | 1.02 | 0.62 | 10x2x28 | 2.46 | 1.20 | 0.61 |
| | 5x2x32 | 1.75 | 1.22 | 0.76 | 10x2x32 | 2.70 | 1.45 | 0.74 |
| | 5x2x50 | 2.28 | 1.97 | 1.64 | 10x2x50 | 3.20 | 2.24 | 1.51 |
| | 5x5x16 | 2.02 | 0.81 | 0.28 | 10x5x16 | 4.05 | 2.04 | 0.38 |
| | 5x5x20 | 2.25 | 0.98 | 0.45 | 10x5x20 | 3.44 | 1.84 | 0.50 |
| | 5x5x24 | 2.52 | 1.19 | 0.57 | 10x5x24 | 4.10 | 2.24 | 0.64 |
| | 5x5x28 | 2.78 | 1.39 | 0.71 | 10x5x28 | 4.75 | 2.69 | 0.89 |
| | 5x5x32 | 3.32 | 1.70 | 0.93 | 10x5x32 | 5.13 | 2.93 | 1.00 |
| | 5x5x50 | 3.71 | 2.31 | 1.66 | 10x5x50 | 5.06 | 3.61 | 1.68 |
| | 5x10x16 | 2.97 | 0.92 | 0.30 | 10x10x16 | 5.74 | 3.77 | 0.45 |
| | 5x10x20 | 2.92 | 1.18 | 0.40 | 10x10x20 | 4.51 | 3.13 | 0.60 |
| | 5x10x24 | 3.40 | 1.38 | 0.52 | 10x10x24 | 5.31 | 3.64 | 0.71 |
| | 5x10x28 | 3.91 | 1.66 | 0.67 | 10x10x28 | 6.07 | 4.12 | 0.80 |
| | 5x10x32 | 4.27 | 1.82 | 0.81 | 10x10x32 | 6.82 | 4.73 | 1.00 |
| 5x10x50 | 4.60 | 2.70 | 1.47 | 10x10x50 | 6.10 | 4.77 | 1.48 | |

Quadro 11

| | Problema | B | C | D | Problema | B | C | D |
|-------------------------------------|----------|------|------|------|----------|------|------|------|
| | 4x16 | 0.47 | 0.35 | 0.26 | 8x16 | 0.75 | 0.42 | 0.24 |
| Razão S | 4x32 | 1.02 | 0.88 | 0.75 | 8x32 | 1.30 | 0.82 | 0.60 |
| Banco 1 | 4x64 | 1.92 | 1.95 | 1.85 | 8x64 | 2.21 | 1.97 | 1.70 |
| Programa pp5 | 4x128 | 3.50 | 4.32 | 4.14 | 8x128 | 3.48 | 4.05 | 3.79 |
| | 4x192 | 4.64 | 5.84 | 5.89 | 8x192 | 4.43 | 5.44 | 5.40 |
| Rede 4 x 4 | 4x256 | 5.58 | 7.08 | 7.00 | 8x256 | 5.19 | 6.51 | 6.62 |
| | 5x16 | 0.50 | 0.37 | 0.25 | 10x16 | 0.92 | 0.51 | 0.27 |
| Parcial: só resultado da exploração | 5x32 | 1.08 | 0.90 | 0.67 | 10x32 | 1.40 | 0.97 | 0.64 |
| | 5x64 | 1.98 | 1.99 | 1.77 | 10x64 | 2.36 | 1.95 | 1.61 |
| | 5x128 | 3.57 | 4.18 | 3.97 | 10x128 | 3.62 | 3.87 | 3.67 |
| | 5x192 | 4.72 | 5.95 | 5.73 | 10x192 | 4.43 | 5.30 | 5.18 |
| | 5x256 | 5.61 | 6.80 | 6.88 | 10x256 | 5.22 | 6.38 | 6.34 |

Quadro 12

| | | | | | | | | |
|-------------------------------------|---------|------|------|----------|---------|------|------|------|
| | 4x2x16 | 0.84 | 0.45 | 0.27 | 8x2x16 | 1.38 | 0.60 | 0.25 |
| | 4x2x20 | 0.99 | 0.60 | 0.37 | 8x2x20 | 1.42 | 0.71 | 0.36 |
| Razão S | 4x2x24 | 1.10 | 0.71 | 0.46 | 8x2x24 | 1.65 | 0.84 | 0.46 |
| Banco 2 | 4x2x28 | 1.35 | 0.90 | 0.63 | 8x2x28 | 1.85 | 1.03 | 0.62 |
| Programa pp5 | 4x2x32 | 1.45 | 0.99 | 0.68 | 8x2x32 | 1.99 | 1.16 | 0.68 |
| | 4x2x50 | 1.90 | 1.54 | 1.23 | 8x2x50 | 2.60 | 1.72 | 1.18 |
| Rede 4 x 4 | 4x5x16 | 1.67 | 0.68 | 0.27 | 8x5x16 | 2.67 | 1.05 | 0.31 |
| | 4x5x20 | 1.69 | 0.75 | 0.36 | 8x5x20 | 2.70 | 1.40 | 0.45 |
| Parcial: só resultado da exploração | 4x5x24 | 2.00 | 0.92 | 0.45 | 8x5x24 | 3.09 | 1.59 | 0.52 |
| | 4x5x28 | 2.18 | 1.01 | 0.55 | 8x5x28 | 3.54 | 1.83 | 0.67 |
| | 4x5x32 | 2.37 | 1.17 | 0.65 | 8x5x32 | 3.97 | 2.09 | 0.79 |
| | 4x5x50 | 2.84 | 1.97 | 1.36 | 8x5x50 | 3.99 | 2.82 | 1.39 |
| | 4x10x16 | 2.45 | 0.85 | 0.26 | 8x10x16 | 4.02 | 1.75 | 0.27 |
| | 4x10x20 | 2.43 | 1.09 | 0.36 | 8x10x20 | 3.37 | 1.86 | 0.41 |
| | 4x10x24 | 2.76 | 1.31 | 0.52 | 8x10x24 | 4.08 | 2.34 | 0.53 |
| | 4x10x28 | 3.16 | 1.44 | 0.58 | 8x10x28 | 4.68 | 2.72 | 0.67 |
| | 4x10x32 | 3.54 | 1.66 | 0.71 | 8x10x32 | 5.18 | 2.85 | 0.79 |
| | 4x10x50 | 3.77 | 2.39 | 1.47 | 8x10x50 | 4.70 | 3.29 | 1.31 |
| | 5x2x16 | 0.95 | 0.48 | 0.26 | 10x2x16 | 1.74 | 0.77 | 0.33 |
| | 5x2x20 | 1.09 | 0.61 | 0.36 | 10x2x20 | 1.79 | 0.83 | 0.34 |
| | 5x2x24 | 1.22 | 0.76 | 0.48 | 10x2x24 | 1.97 | 0.96 | 0.45 |
| | 5x2x28 | 1.41 | 0.93 | 0.57 | 10x2x28 | 2.23 | 1.10 | 0.55 |
| | 5x2x32 | 1.60 | 1.10 | 0.68 | 10x2x32 | 2.43 | 1.30 | 0.66 |
| | 5x2x50 | 2.05 | 1.74 | 1.46 | 10x2x50 | 2.82 | 1.98 | 1.29 |
| | 5x5x16 | 1.87 | 0.75 | 0.26 | 10x5x16 | 3.59 | 1.78 | 0.33 |
| | 5x5x20 | 2.03 | 0.91 | 0.42 | 10x5x20 | 3.02 | 1.69 | 0.45 |
| | 5x5x24 | 2.27 | 1.10 | 0.51 | 10x5x24 | 3.57 | 2.00 | 0.55 |
| | 5x5x28 | 2.50 | 1.28 | 0.64 | 10x5x28 | 4.11 | 2.36 | 0.74 |
| 5x5x32 | 2.96 | 1.52 | 0.82 | 10x5x32 | 4.45 | 2.54 | 0.82 | |
| 5x5x50 | 3.27 | 2.09 | 1.42 | 10x5x50 | 4.33 | 3.09 | 1.29 | |
| 5x10x16 | 2.72 | 0.87 | 0.28 | 10x10x16 | 4.99 | 3.26 | 0.35 | |
| 5x10x20 | 2.62 | 1.12 | 0.38 | 10x10x20 | 3.87 | 2.90 | 0.51 | |
| 5x10x24 | 3.04 | 1.30 | 0.49 | 10x10x24 | 4.55 | 3.32 | 0.58 | |
| 5x10x28 | 3.48 | 1.55 | 0.61 | 10x10x28 | 5.18 | 3.67 | 0.65 | |
| 5x10x32 | 3.81 | 1.69 | 0.74 | 10x10x32 | 5.82 | 4.14 | 0.78 | |
| 5x10x50 | 3.99 | 2.42 | 1.28 | 10x10x50 | 5.14 | 4.16 | 1.09 | |

Quadro 13

| | Problema | B | C | D | Problema | B | C | D |
|------------------------------|-----------------|----------|----------|----------|-----------------|----------|----------|----------|
| | 4x16 | 0.45 | 0.33 | 0.27 | 8x16 | 0.70 | 0.38 | 0.25 |
| Razão S | 4x32 | 0.90 | 0.77 | 0.81 | 8x32 | 1.17 | 0.72 | 0.65 |
| Banco 1 | 4x64 | 1.67 | 1.62 | 2.08 | 8x64 | 1.94 | 1.61 | 1.94 |
| Programa pp4 | 4x128 | 3.00 | 3.51 | 4.88 | 8x128 | 3.01 | 3.14 | 4.43 |
| | 4x192 | 4.01 | 4.82 | 6.70 | 8x192 | 3.83 | 4.33 | 6.15 |
| Rede 4 x 4 | 4x256 | 4.91 | 5.92 | 7.87 | 8x256 | 4.49 | 5.21 | 7.49 |
| | 5x16 | 0.47 | 0.35 | 0.26 | 10x16 | 0.85 | 0.45 | 0.28 |
| Máximo: relação + exploração | 5x32 | 0.98 | 0.78 | 0.72 | 10x32 | 1.26 | 0.82 | 0.70 |
| | 5x64 | 1.75 | 1.66 | 2.00 | 10x64 | 2.03 | 1.57 | 1.79 |
| | 5x128 | 3.04 | 3.38 | 4.55 | 10x128 | 3.02 | 2.90 | 4.31 |
| | 5x192 | 4.06 | 4.80 | 6.60 | 10x192 | 3.75 | 4.04 | 5.98 |
| | 5x256 | 4.88 | 5.65 | 7.83 | 10x256 | 4.39 | 4.91 | 7.31 |

Quadro 14

| | | | | | | | | |
|------------------------------|---------|------|------|----------|----------|------|------|------|
| | 4x2x16 | 0.77 | 0.42 | 0.28 | 8x2x16 | 1.24 | 0.52 | 0.27 |
| | 4x2x20 | 0.91 | 0.53 | 0.39 | 8x2x20 | 1.30 | 0.62 | 0.38 |
| Razão S | 4x2x24 | 1.00 | 0.63 | 0.49 | 8x2x24 | 1.50 | 0.71 | 0.49 |
| Banco 2 | 4x2x28 | 1.20 | 0.77 | 0.68 | 8x2x28 | 1.64 | 0.85 | 0.68 |
| Programa pp4 | 4x2x32 | 1.28 | 0.83 | 0.77 | 8x2x32 | 1.74 | 0.93 | 0.76 |
| | 4x2x50 | 1.66 | 1.27 | 1.37 | 8x2x50 | 2.20 | 1.31 | 1.37 |
| Rede 4 x 4 | 4x5x16 | 1.49 | 0.57 | 0.29 | 8x5x16 | 2.28 | 0.80 | 0.33 |
| | 4x5x20 | 1.54 | 0.64 | 0.39 | 8x5x20 | 2.41 | 1.02 | 0.49 |
| Máximo: relação + exploração | 4x5x24 | 1.79 | 0.76 | 0.48 | 8x5x24 | 2.67 | 1.14 | 0.58 |
| | 4x5x28 | 1.90 | 0.82 | 0.59 | 8x5x28 | 2.88 | 1.17 | 0.75 |
| | 4x5x32 | 2.03 | 0.91 | 0.72 | 8x5x32 | 3.16 | 1.27 | 0.90 |
| | 4x5x50 | 2.42 | 1.47 | 1.53 | 8x5x50 | 3.26 | 1.56 | 1.67 |
| | 4x10x16 | 2.12 | 0.67 | 0.27 | 8x10x16 | 3.21 | 1.06 | 0.29 |
| | 4x10x20 | 2.22 | 0.86 | 0.37 | 8x10x20 | 2.99 | 1.22 | 0.44 |
| | 4x10x24 | 2.42 | 1.01 | 0.56 | 8x10x24 | 3.38 | 1.40 | 0.59 |
| | 4x10x28 | 2.68 | 1.04 | 0.63 | 8x10x28 | 3.70 | 1.43 | 0.78 |
| | 4x10x32 | 2.91 | 1.17 | 0.78 | 8x10x32 | 3.93 | 1.42 | 0.92 |
| | 4x10x50 | 3.11 | 1.58 | 1.64 | 8x10x50 | 3.89 | 1.64 | 1.57 |
| | 5x2x16 | 0.88 | 0.43 | 0.27 | 10x2x16 | 1.55 | 0.64 | 0.35 |
| | 5x2x20 | 1.01 | 0.54 | 0.38 | 10x2x20 | 1.63 | 0.69 | 0.37 |
| | 5x2x24 | 1.12 | 0.67 | 0.51 | 10x2x24 | 1.76 | 0.79 | 0.49 |
| | 5x2x28 | 1.28 | 0.80 | 0.61 | 10x2x28 | 1.93 | 0.87 | 0.61 |
| | 5x2x32 | 1.42 | 0.92 | 0.76 | 10x2x32 | 2.08 | 0.98 | 0.73 |
| | 5x2x50 | 1.79 | 1.42 | 1.60 | 10x2x50 | 2.40 | 1.42 | 1.49 |
| | 5x5x16 | 1.67 | 0.63 | 0.28 | 10x5x16 | 2.92 | 1.08 | 0.38 |
| | 5x5x20 | 1.84 | 0.76 | 0.45 | 10x5x20 | 2.69 | 1.14 | 0.50 |
| | 5x5x24 | 2.02 | 0.89 | 0.56 | 10x5x24 | 3.01 | 1.28 | 0.63 |
| | 5x5x28 | 2.16 | 0.97 | 0.71 | 10x5x28 | 3.28 | 1.34 | 0.88 |
| | 5x5x32 | 2.47 | 1.13 | 0.92 | 10x5x32 | 3.47 | 1.38 | 0.97 |
| | 5x5x50 | 2.72 | 1.49 | 1.64 | 10x5x50 | 3.46 | 1.58 | 1.66 |
| | 5x10x16 | 2.33 | 0.69 | 0.30 | 10x10x16 | 3.88 | 1.46 | 0.45 |
| | 5x10x20 | 2.37 | 0.88 | 0.40 | 10x10x20 | 3.39 | 1.55 | 0.59 |
| | 5x10x24 | 2.63 | 0.98 | 0.52 | 10x10x24 | 3.72 | 1.65 | 0.70 |
| | 5x10x28 | 2.88 | 1.08 | 0.66 | 10x10x28 | 4.03 | 1.56 | 0.80 |
| | 5x10x32 | 3.08 | 1.13 | 0.81 | 10x10x32 | 4.31 | 1.59 | 1.00 |
| 5x10x50 | 3.26 | 1.46 | 1.46 | 10x10x50 | 4.15 | 1.69 | 1.47 | |

Quadro 15

| | Problema | B | C | D | Problema | B | C | D |
|-------------------------------------|-----------------|----------|----------|----------|-----------------|----------|----------|----------|
| | 4x16 | 0.50 | 0.37 | 0.28 | 8x16 | 0.79 | 0.45 | 0.25 |
| Razão S | 4x32 | 1.08 | 0.95 | 0.81 | 8x32 | 1.41 | 0.88 | 0.66 |
| Banco 1 | 4x64 | 2.14 | 2.20 | 2.09 | 8x64 | 2.50 | 2.23 | 1.95 |
| Programa pp4 | 4x128 | 3.94 | 4.97 | 4.88 | 8x128 | 3.94 | 4.60 | 4.46 |
| | 4x192 | 5.22 | 6.64 | 6.68 | 8x192 | 4.94 | 6.29 | 6.20 |
| Rede 2 x 2 | 4x256 | 6.30 | 7.99 | 7.86 | 8x256 | 5.73 | 7.34 | 7.43 |
| | 5x16 | 0.53 | 0.39 | 0.26 | 10x16 | 0.98 | 0.55 | 0.28 |
| Parcial: só resultado da exploração | 5x32 | 1.17 | 0.96 | 0.73 | 10x32 | 1.51 | 1.07 | 0.70 |
| | 5x64 | 2.21 | 2.26 | 2.00 | 10x64 | 2.59 | 2.21 | 1.85 |
| | 5x128 | 3.94 | 4.81 | 4.54 | 10x128 | 3.99 | 4.46 | 4.27 |
| | 5x192 | 5.20 | 6.70 | 6.56 | 10x192 | 4.94 | 6.07 | 6.00 |
| | 5x256 | 6.15 | 7.72 | 7.75 | 10x256 | 5.78 | 7.37 | 7.32 |

Quadro 16

| | | | | | | | | |
|-------------------------------------|---------|------|------|----------|----------|------|------|------|
| | 4x2x16 | 0.89 | 0.49 | 0.28 | 8x2x16 | 1.48 | 0.65 | 0.27 |
| | 4x2x20 | 1.05 | 0.63 | 0.39 | 8x2x20 | 1.53 | 0.76 | 0.38 |
| Razão S | 4x2x24 | 1.17 | 0.76 | 0.49 | 8x2x24 | 1.80 | 0.91 | 0.49 |
| Banco 2 | 4x2x28 | 1.45 | 0.97 | 0.68 | 8x2x28 | 2.02 | 1.12 | 0.69 |
| Programa pp4 | 4x2x32 | 1.59 | 1.09 | 0.76 | 8x2x32 | 2.18 | 1.31 | 0.78 |
| | 4x2x50 | 2.10 | 1.73 | 1.36 | 8x2x50 | 2.93 | 1.96 | 1.37 |
| Rede 2 x 2 | 4x5x16 | 1.81 | 0.74 | 0.29 | 8x5x16 | 2.91 | 1.13 | 0.33 |
| | 4x5x20 | 1.84 | 0.81 | 0.39 | 8x5x20 | 3.03 | 1.50 | 0.49 |
| Parcial: só resultado da exploração | 4x5x24 | 2.20 | 0.99 | 0.49 | 8x5x24 | 3.48 | 1.74 | 0.58 |
| | 4x5x28 | 2.39 | 1.11 | 0.60 | 8x5x28 | 4.00 | 2.03 | 0.75 |
| | 4x5x32 | 2.61 | 1.26 | 0.72 | 8x5x32 | 4.53 | 2.36 | 0.90 |
| | 4x5x50 | 3.19 | 2.20 | 1.52 | 8x5x50 | 4.66 | 3.30 | 1.73 |
| | 4x10x16 | 2.65 | 0.91 | 0.27 | 8x10x16 | 4.51 | 1.91 | 0.29 |
| | 4x10x20 | 2.70 | 1.15 | 0.38 | 8x10x20 | 3.87 | 1.99 | 0.45 |
| | 4x10x24 | 3.08 | 1.41 | 0.56 | 8x10x24 | 4.70 | 2.54 | 0.59 |
| | 4x10x28 | 3.53 | 1.57 | 0.63 | 8x10x28 | 5.43 | 3.03 | 0.79 |
| | 4x10x32 | 3.98 | 1.81 | 0.78 | 8x10x32 | 5.99 | 3.17 | 0.95 |
| | 4x10x50 | 4.31 | 2.66 | 1.65 | 8x10x50 | 5.51 | 3.62 | 1.58 |
| | 5x2x16 | 1.01 | 0.51 | 0.27 | 10x2x16 | 1.88 | 0.85 | 0.35 |
| | 5x2x20 | 1.17 | 0.65 | 0.39 | 10x2x20 | 1.96 | 0.89 | 0.37 |
| | 5x2x24 | 1.31 | 0.82 | 0.52 | 10x2x24 | 2.15 | 1.03 | 0.49 |
| | 5x2x28 | 1.53 | 1.02 | 0.62 | 10x2x28 | 2.46 | 1.20 | 0.61 |
| | 5x2x32 | 1.75 | 1.22 | 0.76 | 10x2x32 | 2.70 | 1.45 | 0.74 |
| | 5x2x50 | 2.28 | 1.97 | 1.64 | 10x2x50 | 3.20 | 2.24 | 1.51 |
| | 5x5x16 | 2.02 | 0.81 | 0.28 | 10x5x16 | 4.05 | 2.04 | 0.38 |
| | 5x5x20 | 2.25 | 0.98 | 0.45 | 10x5x20 | 3.44 | 1.84 | 0.50 |
| | 5x5x24 | 2.52 | 1.19 | 0.57 | 10x5x24 | 4.10 | 2.24 | 0.64 |
| | 5x5x28 | 2.78 | 1.39 | 0.71 | 10x5x28 | 4.75 | 2.69 | 0.89 |
| | 5x5x32 | 3.32 | 1.70 | 0.93 | 10x5x32 | 5.13 | 2.93 | 1.00 |
| | 5x5x50 | 3.71 | 2.31 | 1.66 | 10x5x50 | 5.06 | 3.61 | 1.68 |
| | 5x10x16 | 2.97 | 0.92 | 0.30 | 10x10x16 | 5.74 | 3.77 | 0.45 |
| | 5x10x20 | 2.92 | 1.18 | 0.40 | 10x10x20 | 4.51 | 3.13 | 0.60 |
| | 5x10x24 | 3.40 | 1.38 | 0.52 | 10x10x24 | 5.31 | 3.64 | 0.71 |
| | 5x10x28 | 3.91 | 1.66 | 0.67 | 10x10x28 | 6.07 | 4.12 | 0.80 |
| | 5x10x32 | 4.27 | 1.82 | 0.81 | 10x10x32 | 6.82 | 4.73 | 1.00 |
| 5x10x50 | 4.60 | 2.70 | 1.47 | 10x10x50 | 6.10 | 4.77 | 1.48 | |

Quadro 17

| | Problema | B | C | D | Problema | B | C | D |
|-------------------------------------|-----------------|----------|----------|----------|-----------------|----------|----------|----------|
| | 4x16 | 0.67 | 0.57 | 0.44 | 8x16 | 1.01 | 0.64 | 0.37 |
| Razão S | 4x32 | 1.26 | 1.19 | 1.05 | 8x32 | 1.54 | 1.12 | 0.87 |
| Banco 1 | 4x64 | 2.12 | 2.35 | 2.28 | 8x64 | 2.33 | 2.29 | 2.12 |
| Programa pp4 | 4x128 | 3.19 | 3.85 | 3.84 | 8x128 | 3.17 | 3.73 | 3.72 |
| | 4x192 | 3.79 | 4.56 | 4.61 | 8x192 | 3.64 | 4.38 | 4.41 |
| Rede 2 x 4 | 4x256 | 4.21 | 5.01 | 5.00 | 8x256 | 4.03 | 4.89 | 4.89 |
| | 5x16 | 0.73 | 0.57 | 0.39 | 10x16 | 1.18 | 0.77 | 0.43 |
| Parcial: só resultado da exploração | 5x32 | 1.36 | 1.23 | 0.98 | 10x32 | 1.67 | 1.30 | 0.93 |
| | 5x64 | 2.20 | 2.39 | 2.19 | 10x64 | 2.44 | 2.26 | 2.12 |
| | 5x128 | 3.18 | 3.79 | 3.71 | 10x128 | 3.20 | 3.63 | 3.58 |
| | 5x192 | 3.85 | 4.62 | 4.60 | 10x192 | 3.65 | 4.27 | 4.28 |
| | 5x256 | 4.19 | 4.95 | 5.00 | 10x256 | 4.02 | 4.78 | 4.87 |

Quadro 18

| | | | | | | | | |
|-------------------------------------|---------|------|------|----------|----------|------|------|------|
| | 4x2x16 | 1.12 | 0.66 | 0.39 | 8x2x16 | 1.66 | 0.91 | 0.41 |
| | 4x2x20 | 1.24 | 0.85 | 0.55 | 8x2x20 | 1.70 | 1.00 | 0.53 |
| Razão S | 4x2x24 | 1.43 | 1.04 | 0.71 | 8x2x24 | 1.96 | 1.10 | 0.71 |
| Banco 2 | 4x2x28 | 1.61 | 1.25 | 0.95 | 8x2x28 | 2.01 | 1.39 | 0.92 |
| Programa pp4 | 4x2x32 | 1.75 | 1.34 | 0.99 | 8x2x32 | 2.17 | 1.59 | 0.98 |
| | 4x2x50 | 2.12 | 1.92 | 1.62 | 8x2x50 | 2.63 | 2.12 | 1.61 |
| Rede 2 x 4 | 4x5x16 | 1.92 | 0.98 | 0.42 | 8x5x16 | 2.67 | 1.37 | 0.47 |
| | 4x5x20 | 1.97 | 1.01 | 0.54 | 8x5x20 | 2.76 | 1.67 | 0.66 |
| Parcial: só resultado da exploração | 4x5x24 | 2.30 | 1.24 | 0.66 | 8x5x24 | 3.18 | 1.94 | 0.78 |
| | 4x5x28 | 2.30 | 1.33 | 0.81 | 8x5x28 | 3.05 | 2.02 | 1.01 |
| | 4x5x32 | 2.48 | 1.52 | 0.93 | 8x5x32 | 3.44 | 2.30 | 1.12 |
| | 4x5x50 | 2.79 | 2.28 | 1.76 | 8x5x50 | 3.49 | 2.84 | 1.87 |
| | 4x10x16 | 2.52 | 1.17 | 0.39 | 8x10x16 | 3.46 | 1.95 | 0.40 |
| | 4x10x20 | 2.59 | 1.39 | 0.52 | 8x10x20 | 3.23 | 2.03 | 0.59 |
| | 4x10x24 | 2.96 | 1.60 | 0.78 | 8x10x24 | 3.86 | 2.54 | 0.79 |
| | 4x10x28 | 2.90 | 1.69 | 0.84 | 8x10x28 | 3.61 | 2.59 | 1.01 |
| | 4x10x32 | 3.22 | 1.94 | 1.01 | 8x10x32 | 4.03 | 2.81 | 1.24 |
| | 4x10x50 | 3.34 | 2.56 | 1.80 | 8x10x50 | 3.83 | 2.96 | 1.74 |
| | 5x2x16 | 1.24 | 0.72 | 0.40 | 10x2x16 | 1.98 | 1.09 | 0.50 |
| | 5x2x20 | 1.37 | 0.90 | 0.58 | 10x2x20 | 2.04 | 1.11 | 0.51 |
| | 5x2x24 | 1.53 | 1.09 | 0.72 | 10x2x24 | 2.23 | 1.28 | 0.64 |
| | 5x2x28 | 1.64 | 1.28 | 0.85 | 10x2x28 | 2.28 | 1.39 | 0.81 |
| | 5x2x32 | 1.88 | 1.49 | 1.01 | 10x2x32 | 2.53 | 1.69 | 0.93 |
| | 5x2x50 | 2.24 | 2.13 | 1.88 | 10x2x50 | 2.77 | 2.30 | 1.73 |
| | 5x5x16 | 2.10 | 1.05 | 0.40 | 10x5x16 | 3.23 | 2.02 | 0.50 |
| | 5x5x20 | 2.25 | 1.22 | 0.62 | 10x5x20 | 2.97 | 1.91 | 0.65 |
| | 5x5x24 | 2.54 | 1.44 | 0.78 | 10x5x24 | 3.57 | 2.37 | 0.88 |
| | 5x5x28 | 2.55 | 1.65 | 0.97 | 10x5x28 | 3.36 | 2.38 | 1.08 |
| | 5x5x32 | 2.91 | 1.91 | 1.20 | 10x5x32 | 3.71 | 2.61 | 1.19 |
| | 5x5x50 | 3.05 | 2.37 | 1.83 | 10x5x50 | 3.64 | 2.97 | 1.74 |
| | 5x10x16 | 2.74 | 1.16 | 0.43 | 10x10x16 | 3.91 | 2.94 | 0.57 |
| | 5x10x20 | 2.74 | 1.43 | 0.56 | 10x10x20 | 3.52 | 2.68 | 0.74 |
| | 5x10x24 | 3.18 | 1.66 | 0.72 | 10x10x24 | 4.14 | 3.16 | 0.87 |
| | 5x10x28 | 3.08 | 1.82 | 0.86 | 10x10x28 | 3.80 | 3.00 | 1.02 |
| | 5x10x32 | 3.40 | 1.96 | 1.03 | 10x10x32 | 4.27 | 3.39 | 1.24 |
| 5x10x50 | 3.44 | 2.45 | 1.69 | 10x10x50 | 4.00 | 3.28 | 1.60 | |

Quadro 19

| | Problema | B | C | D | Problema | B | C | D |
|-------------------------------------|-----------------|----------|----------|----------|-----------------|----------|----------|----------|
| | 4x16 | 0.73 | 0.62 | 0.47 | 8x16 | 1.04 | 0.72 | 0.42 |
| Razão S | 4x32 | 1.32 | 1.33 | 1.16 | 8x32 | 1.61 | 1.29 | 1.00 |
| Banco 1 | 4x64 | 2.21 | 2.52 | 2.48 | 8x64 | 2.37 | 2.44 | 2.24 |
| Programa pp4 | 4x128 | 3.22 | 3.95 | 3.95 | 8x128 | 3.14 | 3.80 | 3.78 |
| | 4x192 | 3.84 | 4.65 | 4.67 | 8x192 | 3.67 | 4.47 | 4.42 |
| Rede 4 x 2 | 4x256 | 4.22 | 4.99 | 5.04 | 8x256 | 4.04 | 4.95 | 4.95 |
| | 5x16 | 0.76 | 0.63 | 0.44 | 10x16 | 1.20 | 0.86 | 0.48 |
| Parcial: só resultado da exploração | 5x32 | 1.42 | 1.37 | 1.08 | 10x32 | 1.75 | 1.42 | 1.03 |
| | 5x64 | 2.23 | 2.53 | 2.32 | 10x64 | 2.43 | 2.45 | 2.23 |
| | 5x128 | 3.21 | 3.91 | 3.83 | 10x128 | 3.23 | 3.70 | 3.71 |
| | 5x192 | 3.85 | 4.61 | 4.65 | 10x192 | 3.69 | 4.40 | 4.38 |
| | 5x256 | 4.21 | 4.96 | 5.06 | 10x256 | 4.01 | 4.84 | 4.84 |

Quadro 20

| | | | | | | | | |
|-------------------------------------|---------|------|------|----------|----------|------|------|------|
| | 4x2x16 | 1.15 | 0.78 | 0.47 | 8x2x16 | 1.69 | 1.01 | 0.47 |
| | 4x2x20 | 1.30 | 0.97 | 0.64 | 8x2x20 | 1.71 | 1.10 | 0.60 |
| Razão S | 4x2x24 | 1.48 | 1.16 | 0.79 | 8x2x24 | 2.01 | 1.32 | 0.75 |
| Banco 2 | 4x2x28 | 1.64 | 1.32 | 0.98 | 8x2x28 | 2.05 | 1.54 | 1.01 |
| Programa pp4 | 4x2x32 | 1.78 | 1.43 | 1.07 | 8x2x32 | 2.21 | 1.69 | 1.05 |
| | 4x2x50 | 2.13 | 1.98 | 1.69 | 8x2x50 | 2.58 | 2.09 | 1.62 |
| Rede 4 x 2 | 4x5x16 | 1.99 | 1.07 | 0.47 | 8x5x16 | 2.73 | 1.50 | 0.53 |
| | 4x5x20 | 2.03 | 1.15 | 0.62 | 8x5x20 | 2.74 | 1.79 | 0.72 |
| Parcial: só resultado da exploração | 4x5x24 | 2.42 | 1.41 | 0.78 | 8x5x24 | 3.18 | 2.08 | 0.87 |
| | 4x5x28 | 2.35 | 1.46 | 0.95 | 8x5x28 | 3.05 | 2.08 | 1.06 |
| | 4x5x32 | 2.56 | 1.69 | 1.07 | 8x5x32 | 3.43 | 2.44 | 1.29 |
| | 4x5x50 | 2.84 | 2.34 | 1.88 | 8x5x50 | 3.45 | 2.86 | 1.96 |
| | 4x10x16 | 2.60 | 1.25 | 0.45 | 8x10x16 | 3.42 | 2.11 | 0.46 |
| | 4x10x20 | 2.63 | 1.50 | 0.59 | 8x10x20 | 3.20 | 2.14 | 0.66 |
| | 4x10x24 | 3.01 | 1.73 | 0.83 | 8x10x24 | 3.83 | 2.67 | 0.87 |
| | 4x10x28 | 2.96 | 1.89 | 0.97 | 8x10x28 | 3.58 | 2.66 | 1.09 |
| | 4x10x32 | 3.34 | 2.14 | 1.23 | 8x10x32 | 4.00 | 2.91 | 1.30 |
| | 4x10x50 | 3.35 | 2.62 | 1.96 | 8x10x50 | 3.82 | 3.02 | 1.84 |
| | 5x2x16 | 1.29 | 0.80 | 0.45 | 10x2x16 | 1.95 | 1.20 | 0.56 |
| | 5x2x20 | 1.43 | 1.00 | 0.64 | 10x2x20 | 2.01 | 1.21 | 0.57 |
| | 5x2x24 | 1.60 | 1.19 | 0.80 | 10x2x24 | 2.27 | 1.43 | 0.76 |
| | 5x2x28 | 1.69 | 1.39 | 0.94 | 10x2x28 | 2.29 | 1.57 | 0.91 |
| | 5x2x32 | 1.92 | 1.60 | 1.10 | 10x2x32 | 2.52 | 1.80 | 1.07 |
| | 5x2x50 | 2.25 | 2.22 | 1.92 | 10x2x50 | 2.79 | 2.37 | 1.79 |
| | 5x5x16 | 2.19 | 1.18 | 0.47 | 10x5x16 | 3.14 | 2.18 | 0.57 |
| | 5x5x20 | 2.30 | 1.33 | 0.69 | 10x5x20 | 2.95 | 2.05 | 0.73 |
| | 5x5x24 | 2.62 | 1.63 | 0.86 | 10x5x24 | 3.48 | 2.46 | 0.87 |
| | 5x5x28 | 2.56 | 1.76 | 1.08 | 10x5x28 | 3.29 | 2.43 | 1.10 |
| | 5x5x32 | 2.94 | 2.08 | 1.32 | 10x5x32 | 3.67 | 2.75 | 1.30 |
| | 5x5x50 | 3.06 | 2.41 | 1.89 | 10x5x50 | 3.60 | 2.97 | 1.93 |
| | 5x10x16 | 2.79 | 1.29 | 0.49 | 10x10x16 | 3.79 | 3.05 | 0.64 |
| | 5x10x20 | 2.76 | 1.54 | 0.62 | 10x10x20 | 3.46 | 2.76 | 0.83 |
| | 5x10x24 | 3.22 | 1.80 | 0.81 | 10x10x24 | 4.08 | 3.26 | 0.95 |
| | 5x10x28 | 3.13 | 1.97 | 0.99 | 10x10x28 | 3.75 | 3.06 | 1.10 |
| | 5x10x32 | 3.48 | 2.15 | 1.16 | 10x10x32 | 4.23 | 3.46 | 1.32 |
| 5x10x50 | 3.50 | 2.60 | 1.84 | 10x10x50 | 3.98 | 3.33 | 1.56 | |

REFERÊNCIAS E BIBLIOGRAFIA

- [ACE, 1992a] *ANSI-C front-end documentation, release 92.1*, Novembro 1992.
- [ACE, 1992b] *T800/T9000 back-end documentation, release 92.1*, ACE, Dezembro 1992.
- [Adams, 1990] Adams, D. A., Parallel processing implications for management scientists, *Interfaces*, Vol. 20, No. 3, pp. 88-98.
- [Amdahl, 1967] Amdahl, G. M., Validity of the single processor approach to achieving large scale computer capabilities, *Proceedings of AFIPS Spring Joint Computer Conference*, Atlantic City (NJ), EUA, Abril 1967.
- [Antunes et al., 1994] Antunes, C. H., L. A. Almeida, V. Lopes e J. N. Clímaco, A decision support system dedicated to discrete multiple criteria problems, *Decision Support Systems*, Vol. 12, pp. 327-335.
- [Athas e Seitz, 1988] Athas, W. C. e C. L. Seitz, Multicomputers: message-passing concurrent computers, *IEEE Computer*, Vol. 21, No. 8, Agosto 1988, pp. 26-37.
- [Bana e Costa, 1992] Bana e Costa, C. A., Convictions et aide à la décision, *Bulletin du Groupe de Travail Européen "Aide Multicritère à la Décision"*, Séries 2, No. 3, Automne 93.
- [Bana e Costa, 1992] Bana e Costa, C. A., *Structuration, construction et exploitation d'un modèle multicritère d'aide à la décision*, Tese de Doutoramento em Engenharia de Sistemas, UTL, IST.
- [Bana e Costa, 1994] Bana e Costa, C. A., Problemática e metodologia de estruturação no apoio multicritério à decisão, apresentado em 6º Congresso da APDIO (IO 94), Braga, Portugal, 28-30 Março.
- [Bana e Costa e Ferreira, 1994] Bana e Costa, C. A. e J. A. A. Ferreira, Avaliação multicritério de propostas para a execução de uma grande empreitada de obras públicas: estruturação e avaliação de um caso real de aplicação, apresentado em 6º Congresso da APDIO (IO 94), Braga, Portugal, 28-30 Março.
- [Bana e Costa e Vansnick, 1993] Bana e Costa, C. A. e J.-C. Vansnick, Sur la quantification de jugements de valeur: l'approche MACBETH, *Cahier du LAMSADE*, No. 117, Université Paris-Dauphine, Paris.

- [Barnes et al., 1968] Barnes G. H., R. M. Brown, M. Kato, D. J. Kuck, D. L. Slotnick e R. A. Stokes, The ILLIAC IV computer, *IEEE Transactions on Computers*, Vol. 17, No. 8, pp. 746-757.
- [Ben-Ari, 1982] Ben-Ari, M., *Principles of concurrent programming*, Prentice Hall, Englewood Cliffs, NJ.
- [Bhuyan et al., 1989] Bhuyan, L. N., Q. Yang e D. P. Agrawal, Performance of multiprocessor interconnection networks, *IEEE Computer*, Vol. 22, No. 2, Fevereiro 1989, pp. 25-37.
- [Bouyssou, 1993] Bouyssou, D., Décision multicritère ou aide multicritère?, *Bulletin du Groupe de Travail Européen "Aide Multicritère à la Décision"*, Séries 2, No. 2, Printemps 93.
- [Brans, 1982] 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, pp.183-213.
- [Brans e Vincke, 1985] Brans, J. P. e Ph. Vincke, A preference ranking organisation method (the PROMETHEE method for multiple-criteria decision making), *Management Science*, Vol. 31, No. 6, pp. 647-656.
- [Brans et al., 1986] Brans, J. P., B. Mareshal e Ph. Vincke, How to select and how to rank projects. The PROMETHEE method, *European Journal of Operations Research*, N° 24, pp. 228-238.
- [Chaudhuri, 1982] Chaudhuri, P., *Parallel algorithms*, Prentice-Hall of Australia.
- [Cole, 1992] Cole, M., Parallel software paradigms, in L. Kronsjö e D. Shumsheruddin (Eds.), *Advances in parallel algorithms*, Blackwell, Oxford, pp. 1-25.
- [Costa e Clímaco, 1993] Costa, J. P. e J. N. Clímaco, O processamento paralelo em problemas de otimização, *Investigação Operacional*, Vol. 13, No. 2, pp. 149-168.
- [Costa e Clímaco, 1994] Costa, J. P. e J. N. Clímaco, 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, pp. 255-263.
- [Dennis, 1980] Dennis, J. B., Data flow supercomputers, *IEEE Computer*, Vol. 13, No. 11, November 1980, pp. 48-56.

- [Dias et al., 1993] Dias, L., J. P. Costa e J. N. Clímaco, A parallel approach to the Analytic Hierarchy Process decision support tool, apresentado em 1st International Meeting on Vector and Parallel Processing, Porto, Portugal, 29 Set. - 1 Out. 1993.
- [Dijkstra, 1968] Dijkstra, E. W., Co-operating sequential processes, in F. Genuys (Ed.), *Programming languages*, Academic Press, London, pp. 43-112.
- [Dongarra, 1993] Dongarra, J. J., PVM and MPI: tools for concurrent computing, apresentado em 1st International Meeting on Vector and Parallel Processing, Porto, Portugal, 29 Set. - 1 Out. 1993.
- [Duncan, 1990] Duncan, R., A survey of parallel computer architectures, *IEEE Computer*, Vol. 23, No. 2, Fevereiro 1990, pp. 5-16.
- [Flynn, 1966] Flynn, M. J., Very high-speed computing systems, *Proceedings of the IEEE*, Vol. 54, pp. 1901-1909.
- [Flynn, 1972] Flynn, M. J., Some computer organizations and their effectiveness, *IEEE Transactions on Computers*, Vol. 21, No. 9, pp. 948-960.
- [Gelernter, 1988] Gelernter, D., Getting the job done, *BYTE*, Novembro 1988, pp. 301-308.
- [Gentleman, 1981] Gentleman, W., Message passing between sequential processes: the reply primitive and the administrator concept, *Software Practice and Experience*, Vol. 11, No. 5, pp. 453-466.
- [Gibbons e Rytter, 1988] Gibbons, A. e W. Rytter, *Efficient parallel algorithms*, Cambridge University Press, Cambridge.
- [Gustafson, 1988] Gustafson, J. L., Reevaluating Amdahl's law, *Communications of the ACM*, Vol. 31, No. 5, pp. 532-533.
- [Hagerstern et al., 1992] Hagerstern, E., A. Laudin e S. Haridi, DDM - a cache-only memory architecture, *IEEE Computer*, Vol. 25, No. 9, Setembro 1992, pp. 44-54.
- [Hillis, 1985] Hillis, W. D., *The Connection Machines*, MIT Press, Cambridge, MA.
- [Hwang e Yoon, 1981] Hwang, C.-L. e K. Yoon, *Multiple attribute decision making: methods and applications*, Lecture Notes in Economics and Mathematical Systems, No. 186, Springer-Verlag, Berlim.
- [Inmos, 1989] *The transputer data book*, INMOS, 1989.

- [Jacquet-Lagrèze e Siskos, 1982] Jacquet-Lagrèze, E. e J. Siskos, Assessing a set of additive utility functions for multicriteria decision making - the UTA method, *European Journal of Operations Research*, Vol. 10, No. 2, pp. 151-164.
- [Karp e Flatt, 1990] Karp, A. H. e H. P. Flatt, Measuring parallel processor performance, *Communications of the ACM*, Vol. 33, No. 5, pp. 539-543.
- [Keeney, 1992] Keeney, R. L., *Value-focused thinking*, Harvard University Press, Cambridge, MA.
- [Keeney e Raiffa, 1976] Keeney, R. L. e H. Raiffa, *Decisions with multiple objectives: preferences and value tradeoffs*, John Wiley and Sons, New York.
- [Kohonen, 1987] Kohonen, T., *Content-addressable memories*, Springer-Verlag, New York.
- [Kung, 1982] Kung, H. T., Why systolic architectures?, *IEEE Computer*, Vol. 15, No. 1, Janeiro 1982, pp. 37-46.
- [Lakshmivarahan e Dhall, 1990] Lakshmivarahan, S. e S. K. Dhall, *Analysis and design of parallel algorithms: arithmetic and matrix problems*, McGraw-Hill, New York.
- [Mareshal e Brans, 1991] Mareshal, B. e J. P. Brans, BANKADVISER: An industrial evaluation system, *European Journal of Operations Research*, N° 54, pp. 318-324.
- [Maystre et al., 1994] Maystre, L.-Y., J. Pictet e J. Simos, *Méthodes multicritères ELECTRE*, Presses Polytechniques et Universitaires Romandes, Lausanne.
- [Mladineo et al., 1987] Mladineo, N, J. Margeta, J. P. Brans, B. Mareshal, Multicriteria ranking of alternative locations for small scale hydro plants, *European Journal of Operations Research*, N° 31, pp. 215-222.
- [Nudelman, 1991] Nudelman, M., Symmetry, thy name is UNIX, *BYTE*, Junho 1991, pp. 245-253.
- [Nussbaum e Agarwal, 1991] Nussbaum D. e A. Agarwal, Scalability of parallel machines, *Communications of the ACM*, Vol. 34, No. 3, pp. 57-61.
- [Parsytech, 1993] *PARIX, Release 1.2, Software documentation*, Parsytech, Aachen, Março 1993.
- [Pomerol e Barba-Romero, 1993] Pomerol, J.-C. e S. Barba-Romero, *Choix multicritère dans l'entreprise: principe et pratique*, Editions Hermes, Paris.

- [Pountain e Bryan, 1992] Pountain, D. e J. Bryan, All systems go, *BYTE*, Agosto 1992, pp. 112-136.
- [Pountain, 1990] Pountain, D., Virtual channels: the next generation of transputers, *BYTE*, Abril 1990, pp. 3-12.
- [Riganati e Schneck, 1984] Riganati, J. P. e P. B. Schneck, Supercomputing, *IEEE Computer*, Vol. 17, No. 10, Outubro 1984, pp. 97-111.
- [Robinson, 1991] Robinson, M., Popular and parallel, *BYTE*, Junho 1991, pp. 219-228.
- [Roy, 1968] Roy, B., Classement et choix en présence de points de vue multiples (la méthode ELECTRE), *Revue Informatique et Recherche Opérationnelle*, 2e. Année, No. 8, pp. 57-75.
- [Roy, 1978] Roy, B., ELECTRE III: un algorithme de classement fondé sur une représentation flue des préférences en présence de critères multiples, *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, Vol. 20, No. 1, pp. 3-24.
- [Roy, 1985] Roy, B., *Méthodologie multicritère d'aide à la décision*, Economica, Paris.
- [Roy, 1990a] Roy, B., Decision-aid and decision-making, in C. A. Bana e Costa (Ed.), *Readings in Multiple Criteria Decision Aid*, Springer-Verlag, pp. 17-35.
- [Roy, 1990b] 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.
- [Roy, 1993] Roy, B., Decision science or decision-aid science?, *European Journal of Operations Research*, Vol. 66, pp. 184-203.
- [Roy e Bertier, 1971] Roy B. e P. Bertier, La méthode ELECTRE II: une méthode de classement en présence de critères multiples, *Note de travail No. 142, SEMA*, Direction Scientifique, Paris.
- [Roy e Bouyssou 1993] Roy, B. e D. Bouyssou, *Aide multicritère à la décision: méthodes et cas*, Economica, Paris.
- [Roy e Skalka, 1985] Roy, B. e J. M. Skalka, ELECTRE IS: aspects méthodologiques et guide d'utilisation, *Document du LAMSADE*, No. 30, Université Paris-Dauphine.
- [Russel, 1978] Russel, R. M., The CRAY-1 computer system, *Communications of the ACM*, Vol. 21, No. 1, pp. 63-72.

- [Ryan, 1991] Ryan, B., Multiprocessor surf's up, *BYTE*, Junho 1991, pp. 199-206.
- [Saaty, 1980] Saaty, T. L., *The analytic hierarchy process*, McGraw-Hill, New York.
- [Schärlig, 1985] Schärlig, A., *Décider sur plusieurs critères*, Collection Diriger l'Entreprise, Presses Polytechniques Romandes.
- [Seitz, 1985] Seitz, C. L., The cosmic cube, *Communications of the ACM*, Vol. 28, No. 1, pp. 22-33.
- [Shumsheruddin, 1992] Shumsheruddin, D., The neural network paradigm, in L. Kronsjö e D. Shumsheruddin (Eds.), *Advances in parallel algorithms*, Blackwell, Oxford, pp. 66-85.
- [Skalka et al., 1986] Skalka, J., D. Bouyssou et Y. Bernabeu, ELECTRE III et IV - Aspects methodologiques et guide d'utilisation. *Document du LAMSADE*, No. 25, Université Paris-Dauphine.
- [Stankovic, 1988] Stankovic, J. A., Misconceptions about real-time computing: a serious problem for next-generation systems, *IEEE Computer*, Vol. 21, No. 10, Outubro 1988, pp. 10-19.
- [Stankovic, 1992] Stankovic, J. A., Real-time computing, *BYTE*, Agosto 1992, pp. 155-160.
- [Stein, 1991] Stein, R. M., Scaling up: get the message?, *BYTE*, Junho 1991, pp. 231-240.
- [Stenström, 1988] Stenström, P., Reducing contention in shared-memory multiprocessors, *IEEE Computer*, Vol. 21, No. 11, Novembro 1988, pp. 26-37.
- [Steuer, 1986] Steuer, R., *Multiple criteria optimization: theory, computation and application*, Wiley, New York.
- [Thakkar et al., 1990] Thakkar, S., M. Dubois, A. T. Laundrie e G. S. Sohi, Scalable shared-memory multiprocessor architectures, *IEEE Computer*, Vol. 23, No. 6, Junho 1990, pp. 71-73.
- [Tucker e Robertson, 1988] Tucker, L. W. e G. G. Robertson, Architecture and applications of the Connection Machine, *IEEE Computer*, Vol. 21, Agosto 1988, pp. 26-38.

- [Vanderpooten, 1990] Vanderpooten, D., "The construction of prescriptions in outranking methods" in C. A. Bana e Costa (ed.) *Readings in Multiple Criteria Decision Aid*, Springer-Verlag.
- [Vaughan-Nichols, 1991] Vaughan-Nichols, S. J., Catch as cache can, *BYTE*, Junho 1991, pp. 209-216.
- [Vincke, 1989] Vincke, Ph., *L'aide multicritère à la décision*, Éditions de l'Université Libre de Bruxelles, Bruxelles.
- [von Neumann e Morgenstern, 1947] von Neumann, J. e O. Morgenstern, *Theory of games and economic behavior*, Princeton University Press, Princeton, NJ.
- [Wayner, 1989] Wayner, P., VLIW: heir to RISC?, *BYTE*, Agosto 1989, pp. 259-262.
- [Yu, 1992] Yu, W., ELECTRE TRI. Aspects méthodologiques et guide d'utilisation, *Document du LAMSADE*, No. 74, Université Paris-Dauphine.
- [Zenios, 1992] Zenios, S. A., Parallel and super-computing in the practice of management science, apresentado em Commercial parallel processing, Londres, Inglaterra, 4-5 Fevereiro 1992.