

IRIS: um SAD para problemas de classificação baseado em agregação multicritério¹

Luís C. Dias

INESC Coimbra, Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal, e
Faculdade de Economia, Universidade de Coimbra, Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal

LDias@inescc.pt

Vincent Mousseau

LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16,
France

mousseau@lamsade.dauphine.fr

Resumo

O sistema de apoio à decisão IRIS (Interactive Robustness analysis and parameters' Inference for multicriteria Sorting problems) foi concebido para classificar acções (candidatos, clientes, projectos, alternativas, etc.) descritas pelo seu desempenho em múltiplos critérios, segundo um conjunto ordenado de categorias definido antecipadamente. Baseia-se no método de classificação ELECTRE TRI, distinguindo-se por não exigir do decisor valores precisos para todos os parâmetros desse método. Mais realisticamente, procura apenas obter do decisor algumas restrições que esses parâmetros devem respeitar, nomeadamente exemplos de classificação que o método deverá reproduzir. Se as restrições indicadas pelo decisor não forem incompatíveis entre si, então o IRIS inferirá um conjunto de valores para os parâmetros capaz de reproduzir todos os exemplos, indicando ainda a gama de classificações possíveis face às restrições indicadas. Se as restrições forem incompatíveis, então o IRIS sugerirá valores para os parâmetros que minimizam uma medida de erro e permite identificar quais as restrições que, a ser removidas, conduzem a um sistema de restrições com solução.

Palavras chave: Sistemas de Apoio à Decisão, Decisão e conhecimento, Classificação, Informação imprecisa, Inferência de parâmetros.

1 Introdução

Este artigo apresenta o sistema de apoio à decisão IRIS 2.0 (Interactive Robustness analysis and parameters' Inference for multicriteria Sorting problems) para problemas de classificação ordinal multicritério. Nestes problemas, existe um conjunto de *acções* (candidatos, clientes, projectos, alternativas, consoante o contexto) descritas pelo seu desempenho em múltiplos critérios de avaliação. É também necessário definir um conjunto de categorias possíveis, ordenado por ordem de preferência. Por exemplo, pode-se pretender: dividir projectos de I&D pelas categorias “Mau”, “Razoável”, “Bom” e “Muito bom”; classificar pedidos de crédito

¹ Artigo apresentado na 3ª Conf. da Associação Portuguesa de Sistemas de Informação, Coimbra, 20-22 Novembro de 2002.

pelas categorias “Rejeitar”, “Aceitar”, “Aceitar com juro reduzidos”; separar os trabalhadores de uma empresa por categorias associadas a pacotes de incentivos; etc. A avaliação de cada acção é independente das restantes, não sendo as acções comparadas entre si.

A classificação multicritério, ao contrário de outras metodologias (p.ex. métodos da estatística), faz-se considerando os julgamentos de um decisor [Zopounidis e Doumpos 2002]. Será esse decisor a indicar a fronteira entre uma categoria e a seguinte, a determinar a importância de cada critério, etc. As preferências do decisor reflectem-se nos parâmetros do método de classificação, de uma forma que nem sempre é simples de quantificar de forma precisa. Além disso, as preferências do decisor são frequentemente mal conhecidas e evoluem com o tempo. Em muitos casos, não existirá um decisor isolado, mas antes um conjunto de decisores com opiniões e preferências que não são unânimes. Por conseguinte, poderá ser benéfico questionar indirectamente o decisor, pedindo-lhe que classifique algumas acções como exemplos (a partir dos quais se infere o valor de parâmetros [Mousseau e Slowinski 1998]), e/ou usando procedimentos que aceitem informação imprecisa [Dias e Clímaco 2000].

O *software* IRIS apoia a prática da metodologia proposta por [Dias et al. 2002], [Mousseau e Dias 2002], [Mousseau et al. 2002], baseada no método ELECTRE TRI [Yu 1992], [Roy e Bouyssou 1993]. Pretende constituir uma implementação de referência para o ELECTRE TRI, sucedendo aos programas propostos por [Yu 1992] e [Mousseau et al. 1999]. As principais características do IRIS 2.0 são:

- foi concebido para os sistemas operativos Windows 95/98/Me;
- implementa uma versão do método ELECTRE TRI (na sua variante pessimista), proposta por [Mousseau e Dias 2002], que toma em conta o conceito de discordância;
- não exige valores precisos para os coeficientes de importância dos critérios nem para o limiar de corte do método. O decisor pode limitar-se a indicar restrições (intervalos ou outras) que os parâmetros devem respeitar;
- permite que algumas restrições sejam indicadas através de exemplos de classificação que o método deve reproduzir;
- quando as restrições são incompatíveis entre si, sugere alternativas de remoção de restrições por forma a restaurar a consistência do conjunto, conforme [Mousseau et al. 2002];
- quando o conjunto de restrições define um sistema com solução, determina uma combinação “central” de valores para os parâmetros que satisfaz todas as restrições, e determina a gama de categorias a que cada acção podia ser afectada sem violar nenhuma restrição;
- adicionalmente, calcula indicadores acerca da precisão dos parâmetros e dos resultados.

Na secção seguinte recordamos brevemente o método ELECTRE TRI. Na Secção 3 resumimos a metodologia de construção interactiva de um modelo de classificação, inicialmente proposta por [Dias et al. 2002] e continuada em [Mousseau et al. 2002] e [Mousseau e Dias 2002]. A Secção 4 apresenta a estrutura do interface do programa IRIS, que é ilustrado através de uma “visita guiada” na Secção 5. A última secção apresenta os comentários finais.

2 Breve revisão do método ELECTRE TRI

A família de métodos ELECTRE para avaliação multicritério de acções foi criada por Bernard Roy e seus colaboradores ao longo das últimas três décadas [Roy 1991], [Roy e Bouyssou 1993]. Entre estes, o método ELECTRE TRI [Yu 1992], [Roy e Bouyssou 1993] foi especificamente concebido para o problema de classificar um conjunto de acções segundo categorias pré-definidas e ordenadas, com base em múltiplos critérios.

Seja $A = \{a_1, \dots, a_m\}$ um conjunto de acções avaliadas em n critérios g_1, \dots, g_n . Designaremos por $C = \{C^1, \dots, C^k\}$ um conjunto de k categorias por ordem de preferência, sendo C^1 a pior e C^k a melhor. Cada categoria C^h ($h=1, \dots, k$) é definida através de duas acções de referência: b^h delimita a categoria superiormente, enquanto b^{h-1} delimita a categoria inferiormente. Assim, é necessário definir as acções de referência b^0, \dots, b^k , tais que, exceptuando a primeira e a última, cada uma constitui simultaneamente o limite superior de uma categoria e o limite inferior da categoria seguinte (Fig. 1). [Yu, 1992] especifica uma série de condições que as acções de referência devem respeitar, nomeadamente: b^0 (b^k) deve ser em todos os critérios pior (melhor, resp.) do que qualquer das acções a classificar, e cada referência b^h deve ser melhor do que a anterior b^{h-1} em todos os critérios.

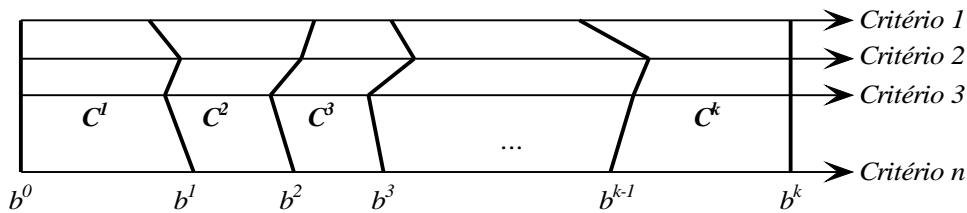


Figura 1 – Definição de categorias no ELECTRE TRI.

A afectação de acções a categorias baseia-se no conceito de *prevalência* (uma relação binária). Diremos que uma acção a_i prevalece sobre uma referência b^h (representando por $a_i S b^h$) quando há argumentos suficientes para sustentar que a_i não é inferior a b^h . Formalmente, determina-se um *índice de credibilidade* da prevalência $s(a_i, b^h)$ que atende ao “peso” dos critérios que apoiam aquela conclusão e à força da discordância oposta pelos restantes critérios (v. Secção 2.1), comparando esse índice com um limiar de corte λ :

$$a_i S b^h \text{ se e apenas se } s(a_i, b^h) \geq \lambda. \quad (1)$$

A afectação de cada acção a_i $\hat{I}A$ processa-se comparando-a às sucessivas acções de referência. Segundo a *variante pessimista* (a mais utilizada na prática) afecta-se cada acção a_i $\hat{I}A$ à categoria mais elevada C^h tal que a_i prevaleça sobre o seu limite inferior (b^{h-1}):

$$a_i @ C^h \text{ se e apenas se } s(a_i, b^{h-1}) \geq \lambda \text{ e } s(a_i, b^h) < \lambda. \quad (2)$$

2.1 Cálculo de um índice de credibilidade $s(a, b)$

a) Determinar um índice de concordância para cada critério

Designemos por D_j a *vantagem* de uma acção a sobre outra acção b no critério de índice j :

$$D_j = \begin{cases} g_j(a) - g_j(b), & \text{se o critério } g_j \text{ for a maximizar} \\ g_j(a) - g_j(b), & \text{se o critério } g_j \text{ for a minimizar} \end{cases}$$

Para cada critério, calcula-se um *índice de concordância para o critério* g_j em relação à conclusão $a S b$:

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

A concordância é total se a vantagem D_j for positiva ou, no caso ser negativa, se a desvantagem (i.e., $-D_j$) não exceder um *limiar de indiferença* q_j . A concordância é nula se a desvantagem

exceder um *limiar de preferência* p_j . A concordância varia linearmente quando a desvantagem se situa entre aqueles dois limiares.

b) Determinar o índice de concordância global

Os índices de concordância para os vários critérios são agregados num *índice de concordância global (multicritério)* através da expressão

$$c(a,b) = \sum_{j=1}^n k_j \cdot c_j(a,b)$$

onde k_j (valor não negativo) representa o *coeficiente de importância* (“peso”) do critério de índice j ($j=1, \dots, n$). Os pesos k_1, \dots, k_n devem ser indicados por forma a que a sua soma seja igual a 1.

c) Determinar um índice de discordância para cada critério

Para cada critério, calcula-se igualmente um *índice de discordância para o critério* g_j em relação à conclusão $a S b$:

$$d_j(a,b) = \begin{cases} 0, & \text{se } -D_j \leq u_j \\ (-D_j - u_j)/(v_j - u_j), & \text{se } u_j < -D_j \leq v_j \\ 1, & \text{se } -D_j > v_j \end{cases}$$

A discordância é nula se a vantagem D_j for positiva ou, no caso ser negativa, se a desvantagem (i.e., $-D_j$) não exceder o *limiar de discordância* u_j . A discordância é total (“veta” a conclusão $a S b$) se a desvantagem exceder um *limiar de veto* v_j . A discordância varia linearmente quando a desvantagem se situa entre aqueles dois limiares. Trata-se de uma variante proposta por [Mousseau e Dias, 2002], que sugerem considerar $u_j = 0.25 p_j + 0.75 v_j$ nos casos em que não se deseja introduzir explicitamente o parâmetro u_j .

d) Determinar o índice de credibilidade

O *índice de credibilidade* da afirmação $a S b$ obtém-se dos índices anteriores através da expressão seguinte [Mousseau e Dias, 2002]:

$$s(a,b) = c(a,b) \cdot [1 - d^{\max}(a,b)], \text{ com } d^{\max}(a,b) = \max_{j \in \{1, \dots, n\}} d_j(a,b).$$

Se existir algum critério para o qual $D_j \leq -v_j$ (i.e. se existir algum critério que “oponha um veto”), então o índice $s(a,b)$ torna-se nulo.

3 Construção interactiva de um modelo de classificação

[Dias et al. 2002] propuseram um processo interactivo para ajudar um decisor a construir progressivamente um modelo de classificação para o ELECTRE TRI; mais concretamente, para ajudar o decisor a estabelecer o valor dos pesos k_1, \dots, k_n e do limiar de corte λ . Embora todos os restantes parâmetros tenham de estar fixos, consideram-se como sujeitos a variar aqueles que são mais difíceis de fixar, dada a interdependência existente entre eles. Naquele artigo propôs-se a combinação da inferência de parâmetros de [Mousseau e Slowinski 1998] com a análise de robustez de [Dias e Clímaco 2000], embora o artigo considerasse que não havia discordância (v_j infinitos). [Mousseau e Dias 2002] propõem uma alteração que permite incorporar a discordância, de acordo com a variante apresentada na secção anterior.

Em cada iteração considera-se um conjunto R de restrições ao valor de (I, k_1, \dots, k_n) , que define um conjunto T de combinações admissíveis nessa iteração. As restrições podem ser quaisquer restrições lineares, incluindo:

- um intervalo de valores para λ ;
- um intervalo de valores para cada peso k_j ;
- comparações entre o peso de coligações de critérios, p.ex. $k_1 \geq k_3$, $k_1 \geq k_2 + k_4$;
- limites à classificação de acções (exemplos de afectação), p.ex. dizer que a_1 pertence à categoria C^3 origina (pela equivalência (2)) as restrições $s(a_1, b^2) - \lambda \geq 0$ e $\lambda - s(a_1, b^3) \geq \epsilon$ (ϵ é um valor muito pequeno, necessário por a desigualdade em (2) ser estrita); dizer que a_2 pertence à categoria C^3 ou à C^4 origina as restrições $s(a_2, b^2) - \lambda \geq 0$ e $\lambda - s(a_2, b^4) \geq \epsilon$; dizer que a_3 pertence à categoria C^1 origina apenas a restrição $\lambda - s(a_3, b^1) \geq \epsilon$; etc. Cada uma destas restrições é linear, dado que apenas os pesos e o limiar λ são variáveis [Mousseau e Dias 2002].

Recomenda-se começar o processo interactivo com poucas restrições, ou mesmo nenhuma, acrescentando-as apenas à medida que o decisor vai aprendendo acerca do problema e do modelo, e vai ganhando confiança nos seus julgamentos. Se assim for, as restrições deverão formar um sistema que admite pelo menos uma solução. Porém, também se prevê a possibilidade de as restrições serem incompatíveis entre si, de tal forma que não exista nenhuma combinação de valores para (I, k_1, \dots, k_n) capaz de respeitar todas as restrições. Os resultados e as análises possíveis variarão consoante a situação seja uma ou outra.

3.1 Quando as restrições definem um sistema com solução

Nestes casos, haverá pelo menos uma combinação de valores para (I, k_1, \dots, k_n) que satisfaz todas as restrições de R . Seja T o conjunto de todas essas combinações. O programa IRIS permite inferir uma combinação de valores para os parâmetros, para a qual determina a classificação correspondente. Porém, acrescenta informação que permita aferir quão diferentes poderiam ser as classificações sem violar nenhuma restrição. Assim, o decisor poderá aceitar os resultados com alguma confiança, ou optar por prosseguir de forma a reduzir o conjunto T . Para tal, o IRIS também oferece alguma orientação. Os resultados fornecidos são listados a seguir.

- Através da resolução de um programa linear (ver detalhes em [Mousseau e Dias 2002]), obtém-se uma combinação de valores para os parâmetros que maximiza a menor folga com que as restrições de R são satisfeitas.
- Para cada acção, o IRIS mostra a categoria que lhe corresponde utilizando a combinação inferida.
- Para cada acção, o IRIS mostra a gama de categorias em que esta poderia ser colocada sem violar nenhuma restrição (método de [Dias e Clímaco 2000]). Tal permite observar quais as classificações que mais são afectadas pela imprecisão existente nessa altura. Permite ainda ajudar o decisor a escolher exemplos de classificação que não sejam incompatíveis com as restrições anteriormente introduzidas. Não menos importante, é a possibilidade de extrair conclusões robustas (i.e. válidas para todas as combinações aceitáveis), do tipo “ a_1 pertence à categoria C^3 ou superior”, ou “ a_1 atinge, quando muito, a categoria C^4 ”.
- Para cada classificação possível, o IRIS pode determinar (inferir) uma combinação de valores para os parâmetros que a tornasse realidade. Tal pode ajudar o decisor a fornecer restrições explícitas ao valor dos parâmetros, em particular quando se procuram eliminar

combinações que conduzam a classificações extremas (a pior e a melhor possíveis para cada acção), caso sejam vistas como inadequadas pelo decisor.

- Utilizando simulação Monte-Carlo, o IRIS calcula aproximadamente o volume relativo do poliedro T , o que é um indicador da precisão dos parâmetros (indica a proporção de combinações que é admissível).
- O IRIS também calcula a média geométrica do número de categorias em que cada acção pode ser colocada, o que é um indicador da precisão dos resultados.

3.2 Quando as restrições definem um sistema sem solução

Nestes casos, não haverá nenhuma combinação de valores para (I, k_1, \dots, k_n) que satisfaça todas as restrições de R (i.e., $T = \emptyset$). O programa IRIS infere uma combinação de valores para os parâmetros, para a qual determina a classificação correspondente, que viola algumas restrições. A análise deverá centrar-se na remoção da incompatibilidade existente entre as restrições. Os resultados fornecidos são listados a seguir.

- Através da resolução de um programa linear (ver detalhes em [Mousseau e Dias 2002]), obtém-se uma combinação de valores para os parâmetros que minimiza o maior desvio em relação às restrições de R não satisfeitas.
- Para cada acção, o IRIS mostra a categoria que lhe corresponde utilizando a combinação inferida, assinalando os exemplos de classificação que não tenham sido reproduzidos. Trata-se de uma sugestão que o decisor poderá aceitar ou não.
- Para cada restrição, indica se a combinação inferida a respeita e, em caso negativo, indica qual a magnitude da violação.
- O IRIS inclui um módulo para encontrar, por ordem de cardinalidade, conjuntos de restrições que, a ser removidas, tornariam as restantes compatíveis entre si. Procura indicar primeiro conjuntos de uma restrição (se existirem), seguidos de conjuntos com duas restrições (se existirem), etc. Para tal resolve uma série de programas lineares com variáveis binárias (ver detalhes em [Mousseau et al. 2002]). De entre os conjuntos de restrições propostos (subconjuntos de R), o decisor deverá escolher aquele de que mais facilmente abdicaria, sabendo que após retirar as restrições desse conjunto, as restantes restrições serão compatíveis entre si. O facto de este módulo não se limitar a encontrar uma proposta (que minimizasse o número de restrições a retirar), tem em conta o facto de o decisor poder atribuir prioridades distintas às diferentes restrições.

3.3 Processo interactivo

O processo que o IRIS propõe para construção de um modelo de classificação para o ELECTRE TRI é interactivo, na medida em que os resultados de uma determinada iteração são utilizados para orientar o decisor quanto à revisão dos *inputs* para a iteração seguinte. O processo pode-se iniciar com muito pouca informação (intervalos amplos para todas as variáveis (I, k_1, \dots, k_n) , poucas ou nenhuma restrições adicionais, que incluem os exemplos de classificação). Em cada iteração, o decisor deverá alterar a informação anterior de forma mínima, apagando, acrescentando, ou alterando poucas restrições. A rapidez dos cálculos do IRIS permite observar de imediato os efeitos da alteração, que serão melhor compreendidos procedendo desta forma. A aprendizagem contínua acerca do problema e do funcionamento do método, e a progressiva formação das suas preferências, permitem ao decisor ir progredindo de iteração para iteração.

O processo deverá ter como objectivo a progressiva redução do conjunto de combinações aceitáveis para (I, k_1, \dots, k_n) . O decisor poderá terminar o processo quando sinta que a precisão dos resultados é satisfatória para os seus fins (conforme se afere pelas gamas de classificação), ao tempo que se sinta confortável e confiante acerca das restrições que introduziu.

Os produtos tangíveis deste processo interactivo serão:

- um conjunto de exemplos de classificação e de outras restrições que define quais são as combinações de valores aceitáveis para (I, k_1, \dots, k_n) ;
- uma combinação precisa $(I^*, k_1^*, \dots, k_n^*)$, resultante do programa de inferência, que define o modelo de classificação;
- uma categoria precisa e/ou uma gama de categorias para cada acção a classificar, que é robusta em relação à informação fornecida.

Porém, o resultado mais importante do processo interactivo poderá ser um maior discernimento, por parte do decisor, em relação ao problema de decisão com que se deparava, e um melhor conhecimento das suas preferências, que possivelmente se alteram durante esse processo.

4 Estrutura do interface do IRIS

À semelhança de programas como o Internet Explorer ou o Access, o IRIS é do tipo SDI (Single Document Interface), sendo necessário correr várias instâncias do mesmo para trabalhar em mais do que um “documento” em simultâneo. A parte esquerda da janela está associada aos *inputs*, enquanto a parte direita é usada para os *outputs*, podendo o utilizador deslocar a linha que divide aquelas áreas (Fig. 2). Cada uma das áreas está organizada de acordo com a metáfora do “organizer” com múltiplas páginas.

A área da esquerda permite editar os *inputs*, a saber:

- desempenhos das acções a classificar e exemplos de afectação (Página *Actions*);
- valor dos parâmetros fixos, que são os limites das categorias e os limiares de indiferença de preferência, de discordância e de veto (Página *Fixed Par.*);
- limites superiores e inferiores para (I, k_1, \dots, k_n) (Página *Bounds*); e
- restrições adicionais àquelas variáveis (Página *Constraints*) (v. Fig. 3).

Os resultados só reflectem as alterações nos *inputs* após o utilizador ordenar ao IRIS que os calcule de novo. A área da direita permite visualizar resultados, a saber:

- gamas de categorias, classificação inferida e valores inferidos para os parâmetros (Página *Results*);
- programa linear para inferência dos parâmetros (Página *Infer. Prog.*); e
- média geométrica de categorias possíveis por cada acção (Página *Indices*).

4.1 Organização dos menus

As opções de menu são acessíveis a partir da barra no topo, mas também em menus “pop up” que variam consoante o local em que são chamados (com o botão direito do rato):

Menu *FILE*:

New: Cria um novo problema.

Open: abre um problema anteriormente guardado; como se trata de ficheiros de texto, é fácil transferir tabelas de dados de processadores de texto ou de folhas de cálculo.

Report: cria um ficheiro de texto com um relatório dos resultados actuais.

Save Data, Save Data As...: guarda as alterações nos *inputs*.

Print: imprime uma cópia do écran; para imprimir resultados deve-se partir do ficheiro criado pela opção *Report*.

Print Setup: configuração da impressora.

Exit: termina o programa.

Menu *CATEGORIES*:

Split: divide uma categoria em duas, pedindo ao utilizador que estabeleça uma referência que sirva de fronteira entre elas.

Merge: funde duas categorias consecutivas numa só.

Use u_j : mostra e utiliza de forma explícita os parâmetros u_j ($j=1, \dots, n$); caso contrário, considera $u_j = 0.25 p_j + 0.75 v_j$.

Menu *CRITERIA*:

Insert: insere novo critério.

Delete: remove um critério.

Menu *ACTIONS*:

Insert: insere nova acção a classificar.

Delete: remove uma acção.

Erase Examples: remove todas as restrições associadas a exemplos de classificação.

Menu *CONSTRAINTS*:

Insert: insere nova restrição (qualquer restrição linear).

Delete: remove restrição.

Menu *RESULTS*:

Volume Computation: calcula aproximadamente, por simulação Monte Carlo, a proporção de combinações de valores para as variáveis que respeita todas as restrições.

Robust Assignments: actualiza os *outputs*, resolvendo o problema de inferência e determinando as gamas de categorias para cada acção.

by Input Order: ordena os resultados pelo número associado a cada acção.

by Variability Order: ordena os resultados por ordem decrescente da diferença entre a melhor e a pior categoria a que cada acção pode ser classificada.

Menu **INCONSISTENCY**:

Activa o módulo de análise de restrições incompatíveis (quando existam).

Menu **HELP**:

Online Manual: abre o manual do programa.

How to Get Help: descreve como obter ajuda.

About...: informações acerca do IRIS.

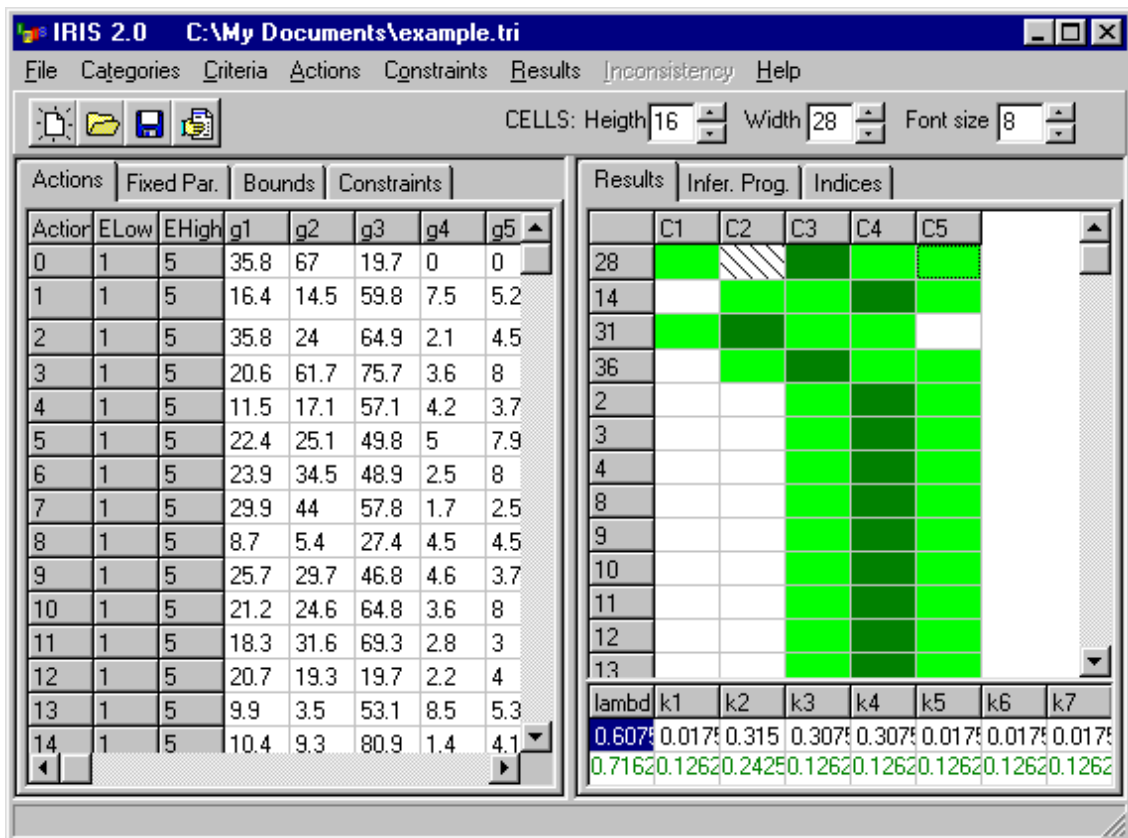


Figura 2 – Resultados iniciais do IRIS 2.0.

5 Visita guiada através de um exemplo

Com o intuito de ilustrar a utilização do programa, iremos seguir os passos de um hipotético decisor, a partir dos dados iniciais de um problema de classificação apresentado detalhadamente em [Dias et al. 2000] (adaptados de uma aplicação do ELECTRE TRI no sector da banca, analisada em [Dimitras et al. 1995]). Embora a versão 2 do IRIS não esteja limitada a problemas sem discordância, nos dados que ora retomamos não se consideravam limiares de veto.

Neste exemplo existem 40 acções representando empresas que devem ser classificadas segundo o seu risco de falência: muito elevado (C^1), elevado (C^2), médio (C^3), baixo (C^4), ou muito baixo (C^5). Estas cinco categorias estão ordenadas da pior (C^1) para a melhor (C^5). Cada acção foi avaliada em sete critérios, introduzindo-se esses dados no IRIS (parte esquerda da Fig. 2).

Iteração inicial:

Para cada variável foram estabelecidos intervalos muito amplos na página *Bounds*: cada peso k_j ($j=1, \dots, 7$) tem que ser positivo e não pode pesar tanto como a reunião de todos os outros, pelo que se colocou $k_j \in [0.01, 0.49]$; para o limiar de corte definiu-se $\lambda \in [0.6, 0.99]$ (teoricamente pode variar entre 0.5 e 1). Adicionalmente, o decisor informou que o segundo critério era aquele com maior importância, i.e., qualquer coligação que o inclua é mais forte do que a mesma coligação com outro critério no seu lugar. Por conseguinte, as restrições $k_2 \geq k_1$ (*ad1*), $k_2 \geq k_3$ (*ad2*), $k_2 \geq k_4$ (*ad3*), $k_2 \geq k_5$ (*ad4*), $k_2 \geq k_6$ (*ad5*) e $k_2 \geq k_7$ (*ad7*) foram introduzidas na página *Constraints* (Fig. 3). Não foi inserido nenhum exemplo de classificação.

Actions	Fixed Par.	Bounds							Constraints	
	lambda	k1	k2	k3	k4	k5	k6	k7	<.=.>	RHS
Norm	0	1	1	1	1	1	1	1	=	1
ad1		-1	1						>=	0
ad2			1	-1					>=	0
ad3			1		-1				>=	0
ad4			1			-1			>=	0
ad5			1				-1		>=	0
ad6			1					-1	>=	0

Figura 3 – Restrições que tornam k_2 o maior dos pesos.

Na parte direita da Fig. 2 mostra-se o tipo de resultados que é possível obter com esta informação apenas. Como há múltiplas combinações de valores para (I, k_1, \dots, k_n) que satisfazem as restrições introduzidas, o programa mostra a gama de categorias em que cada acção pode ser classificada sem violar essas restrições, por ordem decrescente da diferença entre a melhor e a pior categoria (também se pode optar pela ordenação original). Este exemplo apresenta a curiosidade de mostrar que a classificação de a_{28} na categoria C^2 não é possível, apesar da mesma acção poder ser classificada em C^1 ou C^3 (para uma caracterização destas situações veja-se [Dias et al. 2000]).

Em cada gama de categorias, identifica-se através de uma cor mais escura qual a classificação proposta pelo IRIS, com base nos valores inferidos para os parâmetros, que são mostrados na última linha da página *Results*. Na linha imediatamente acima, mostra-se uma combinação de valores para (I, k_1, \dots, k_n) correspondente à célula seleccionada, i.e., à classificação de a_{28} em C^5 (v. Fig. 2). Se o utilizador seleccionar outra célula numa qualquer gama de classificações, estes valores serão automaticamente actualizados (sem necessidade de escolher nenhuma opção dos menus).

Na situação actual, a página *Indices* indicaria que a média (geométrica) de categorias em cada gama é de 2.757. Se o utilizador calculasse a proporção das combinações que respeitam todas as restrições (opção *Volume computation* do menu *Results*), ficaria a saber que esta era cerca de 45 por milhão, e ainda que de entre todas as combinações que respeitam os limites da página *Bounds*, apenas cerca de 14% respeita as restantes restrições. Estes números não são muito informativos em termos absolutos, mas permitirão ao decisor avaliar como mudam de uma iteração para as seguintes.

Segunda e terceira iterações:

Admitamos agora que o decisor se sentia suficientemente confiante para afirmar que a acção a_{28} não era pior do que a categoria C^4 . Podia-se então introduzir facilmente na página *Actions* este exemplo de classificação, embora seja impreciso por admitir duas classificações para essa acção (C^4 e C^5). Após recalculer os resultados, o decisor podia analisá-los tal como anteriormente. Suponhamos que a classificação de a_{31} na categoria C^1 continuava a ser

admissível, e que o decisor conhecia suficientemente aquela acção para afirmar que era um bom exemplo dessa categoria. Nesta terceira iteração, o decisor acrescentaria este exemplo (parte esquerda da Fig. 4) e chegaria aos resultados apresentados na parte direita da Fig. 4 (nesta figura, as acções estão ordenadas pelo seu número identificador).

Pode-se observar que todos os exemplos estão a ser reproduzidos, sendo os valores inferidos para os parâmetros diferentes dos da Fig. 2. Em relação à Fig. 2, observa-se que a acção a_{36} apenas pode ser classificada em duas categorias, e não em quatro, como anteriormente.

Actions							Results								
Fixed Par.		Bounds		Constraints			Inferred constraints					Infer. Prog.		Indices	
Action	ELow	EHigh	g1	g2	g3	g4		C1	C2	C3	C4	C5			
21	1	5	13.2	9.1	74.1	6.1				Green	Dark Green	Green			
22	1	5	9.1	4.1	44.8	3.1					Dark Green	Green			
23	1	5	12.9	1.9	65	14				Green	Dark Green				
24	1	5	5.9	-27.7	77.4	16			Green	Dark Green	Green				
25	1	5	16.9	12.4	60.1	5.1					Dark Green	Green			
26	1	5	16.7	13.1	73.5	11					Dark Green	Green			
27	1	5	14.6	9.7	59.5	6.1					Dark Green	Green			
28	4	5	5.1	4.9	28.9	2.1					Dark Green	Green			
29	1	5	24.4	22.3	32.8	3.1					Dark Green	Green			
30	1	5	29.5	8.6	41.8	5.1					Dark Green	Green			
31	1	1	7.3	-64.5	67.5	30		Dark Green							
32	1	5	23.7	31.9	63.6	12				Green	Dark Green	Green			
33	1	5	18.9	13.5	74.5	12				Green	Dark Green	Green			
34	1	5	13.9	3.3	78.7	14			Green	Dark Green	Green				
35	1	5	-13.3	-31.1	63	21				Dark Green	Green				
36	1	5	6.2	-3.2	46.1	4.1					Dark Green	Green			
37	1	5	4.8	-3.3	71.1	8.1					Dark Green	Green			
38	1	5	0.1	-9.6	42.5	12					Dark Green	Green			
lambda							k1	k2	k3	k4	k5	k6	k7		
							0.6685	0.0682	0.3584	0.0682	0.3001	0.0682	0.0682	0.0682	

Figura 4 – Resultados na terceira iteração.

Quarta iteração:

Imaginemos que o decisor reparava neste momento que a acção a_{24} não pode atingir a categoria C^5 . Para saber o que impede que tal aconteça, o decisor podia inserir como exemplo a classificação daquela acção nesta categoria (Fig. 5, à esquerda).

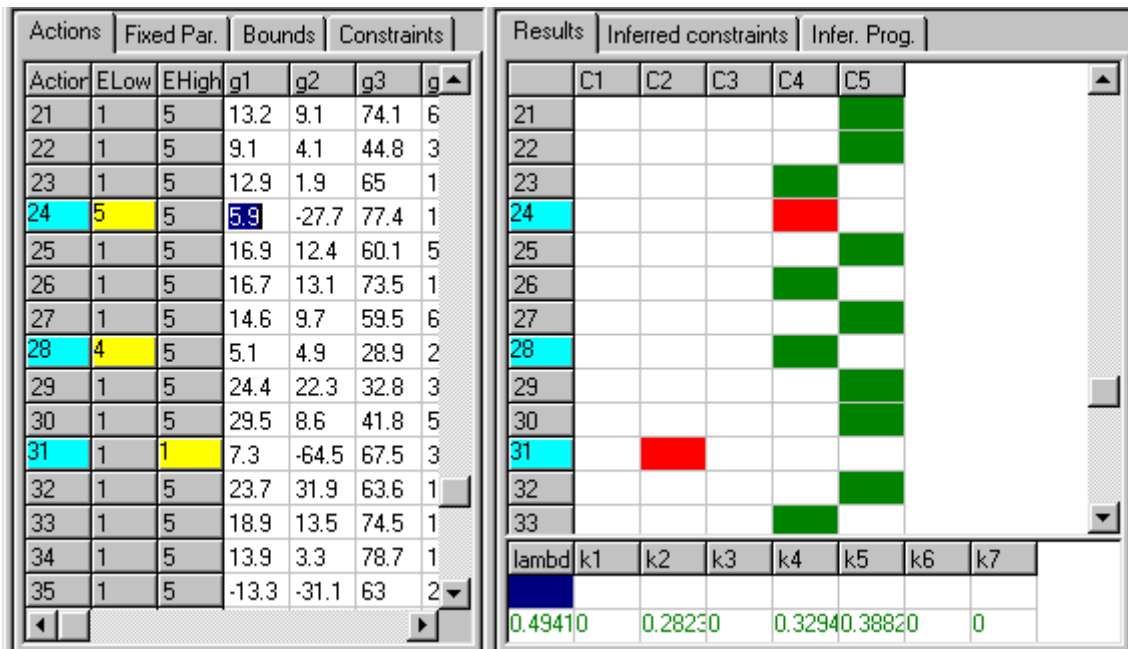


Figura 5 – Resultados na quarta iteração: existem restrições incompatíveis.

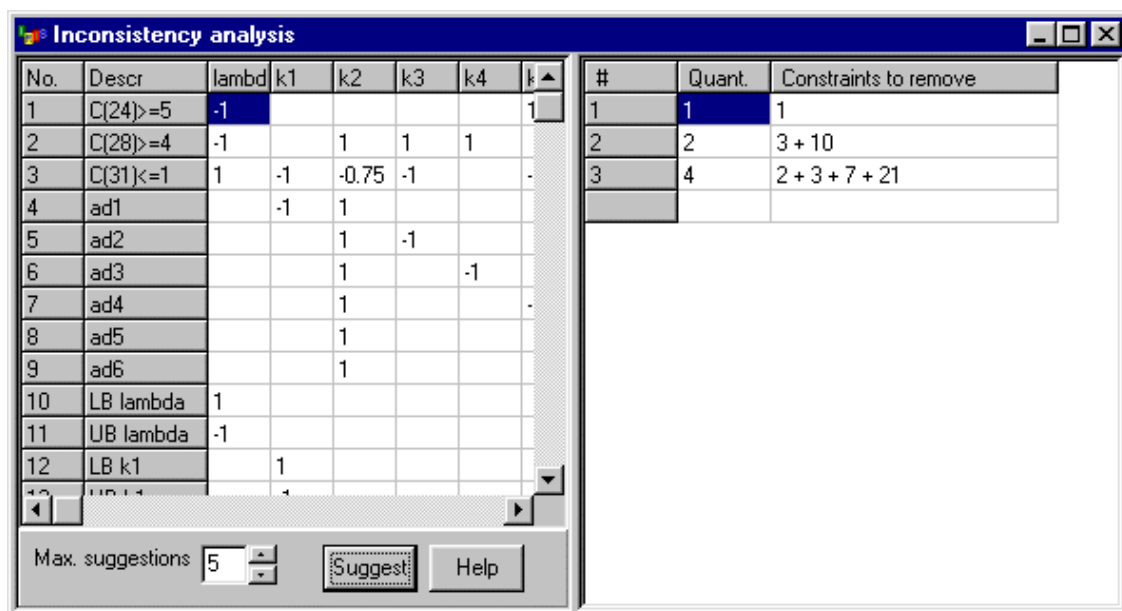


Figura 6 – Análise de restrições incompatíveis.

A nova restrição será incompatível com as anteriores, pelo que os resultados (Fig. 5, à direita) mostram que nem todos os exemplos puderam ser reproduzidos. A combinação de valores inferida (última linha da página *Results*) violará algumas das restrições, mas é aquela em que o maior dos desvios é minimizado.

Nestas circunstâncias, é necessário remover pelo menos uma restrição. O módulo de análise de incompatibilidades permite saber quais são as alternativas para remover a incompatibilidade. Na parte esquerda da Fig. 6, surge uma lista de todas as restrições e um campo onde o utilizador pode indicar um limite para o número máximo de sugestões. Apesar de o limite neste exemplo ser de cinco, apenas existem três formas alternativas de remover a incompatibilidade (parte direita da Fig. 6): ou remover a restrição número 1, que corresponde precisamente ao exemplo que se acabou de introduzir (classificar a_{24} como C^5), ou remover duas restrições, a número 3

(exemplo de classificação de a_{31}) e a número 10 (limite inferior de λ), ou remover quatro restrições (números 2, 3, 7 e 21). As sugestões surgem por ordem do número de restrições que implicam remover.

Conhecidas as alternativas, suponhamos que o decisor optava por recuar na classificação de a_{24} como C^5 , por não desejar abdicar das outras restrições. Regressávamos, assim, à situação da Fig. 4.

Quinta iteração:

Enfim, imaginemos que o decisor indicava que a_{10} devia ser classificada como C^3 , chegando aos resultados da Fig. 7 (onde as acções surgem ordenadas pela variabilidade das classificações). Nesta altura, trinta acções estão classificadas, com precisão, numa categoria, e existem dez acções que possuem duas categorias possíveis cada. A página *Indices* indica que a média (geométrica) de categorias em cada gama é de 1.189. Se o utilizador calculasse de novo a proporção das combinações que respeitam todas as restrições, ficaria a saber que esta era cerca de 0.1 por milhão, e ainda que de entre todas as combinações que respeitam os limites da página *Bounds*, apenas cerca de 0.04% respeita as restantes restrições.

Terminamos aqui a ilustração do programa. O decisor poderia aceitar como finais estes resultados, ou então poderia prosseguir acrescentando nova informação ou revendo a informação já introduzida.

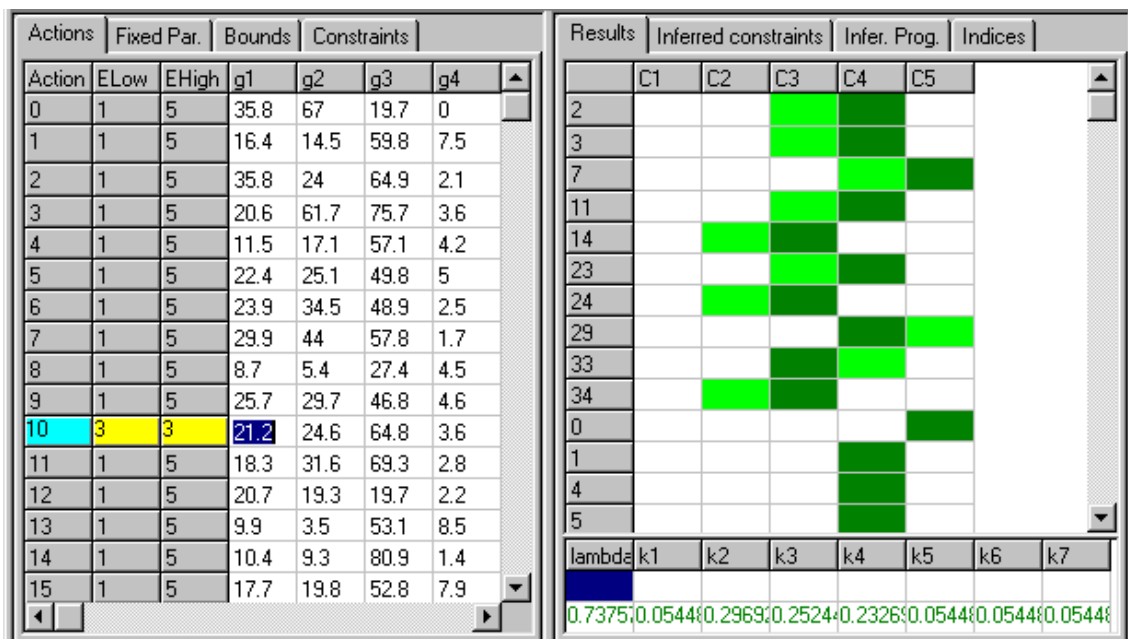


Figura 7 – Resultados na quinta iteração.

6 Conclusões

O programa IRIS destina-se ao apoio à decisão em problemas de classificação. Não trata de situações em que se pretende classificar acções de forma automática, mas de situações em que um decisor procura avaliar acções de forma absoluta, atribuindo-lhes uma nota ou, o mesmo é dizer, separando-as por categorias ordenadas. As preferências subjectivas desse decisor serão, pois, determinantes.

A principal vantagem do programa é o apoio que pode dar a decisores que não conheçam bem as suas preferências, ou que não as saibam quantificar de forma precisa e de acordo com o significado dos parâmetros do ELECTRE TRI. Aceitando informação imprecisa (restrições sobre os parâmetros, em lugar de valores precisos), o IRIS apresenta de forma integrada uma metodologia de inferência de parâmetros e de procura das conclusões lícitas face à informação imprecisa, as conclusões robustas. Trata-se de um processo que fomenta a aprendizagem e a progressiva delimitação da variação dos *inputs* e dos *outputs*.

A principal desvantagem será o não considerar todos os parâmetros como variáveis, tendo o decisor que fixar o valor dos limites das categorias e dos limiares de indiferença, preferência e veto. Porém, já considera como variáveis os parâmetros porventura mais difíceis de fixar: os pesos dos critérios, além do limiar de corte da relação de prevalência.

Uma das vias para investigação futura consiste precisamente em procurar resolver os problemas de inferência quando há outros parâmetros variáveis, tais como os limiares de veto. Outra via interessante, já em andamento, é a análise de restrições incompatíveis por forma a sugerir ao decisor relaxações das mesmas, em lugar da sua remoção.

Um aspecto a que atribuímos grande importância é o da interactividade, que o interface do IRIS favorece. A interactividade, conforme sublinha Courbon [Courbon et al. 1994], não deve ser uma preocupação de última hora, mas antes um utensílio de concepção. No processo que o IRIS vem apoiar, o modelo que o SAD usa vai sendo afinado de forma interactiva ao longo do processo de decisão. De certa forma, como sugere Courbon no mesmo artigo, um SAD tanto pode ajudar um decisor (porque aquele, no nosso caso, explora a imprecisão do modelo e extrai conclusões), como pode ser ajudado pelo decisor (porque este, no nosso caso, vai delimitando a imprecisão do modelo).

Agradecimentos

A metodologia na qual se baseia o programa IRIS foi construída, ao longo dos últimos anos, por Luís Dias, Vincent Mousseau, José Figueira, João Clímaco e Carlos Gomes da Silva, no quadro de dois projectos de colaboração Luso-Francesa (ref. 328J4 e 500B4) financiados pelo ICCTI e pela Embaixada de França em Portugal. Carlos Gomes da Silva programou o módulo da versão 1 do IRIS destinado à análise de restrições incompatíveis, reformulado na versão 2 por Rui Lourenço, que acrescentou um módulo de programação inteira mista.

Referências

- Courbon, J.C., D. Dubois, B. Roy, J.C. Pomerol (1994), Autour de l'aide à la décision et de l'intelligence artificielle, *Cahiers du LAFORIA* 94/01, Institut Blaise Pascal, Paris.
- Dias, L.C., J.N. Clímaco (2000), ELECTRE TRI for Groups with Imprecise Information on Parameter Values, *Group Decision and Negotiation* 9, 355-377.
- Dias, L., V. Mousseau, J. Figueira, J. Clímaco (2000), An Aggregation/Disaggregation Approach to obtain Robust Conclusions with ELECTRE TRI, *Cahier du LAMSADE*, No. 174, Université Paris-Dauphine.
- Dias, L., V. Mousseau, J. Figueira, J. Clímaco (2002), An Aggregation/Disaggregation Approach to Obtain Robust Conclusions with ELECTRE TRI, *European Journal of Operational Research* 138, 332-348.
- Dimitras, A., C. Zopounidis, C. Hurson (1995), Multicriteria decision aid method for the assessment of business failure risk, *Foundations of Computing and Decision Sciences*, 20(2), 99-112.
- Mousseau, V., L.C. Dias (2002), Valued outranking relations in Electre providing manageable disaggregation procedures, *Cahier du LAMSADE*, No. 189, Université Paris-Dauphine.
- Mousseau, V., J. Figueira, L. Dias, J. Clímaco, C. Gomes da Silva (2002), Resolving inconsistencies among constraints on the parameters of an MCDA model, a publicar por *European Journal of Operational Research*.
- Mousseau V., R. Slowinski (1998), Inferring an ELECTRE TRI Model from Assignment Examples, *Journal of Global Optimization*, 12, 157-174.
- Mousseau, V., R. Slowinski, P. Zielniewicz (1999), A User-oriented Implementation of the ELECTRE-TRI Method Integrating Preference Elicitation Support, *Computers & Operations Research*, 27, 757-777.
- Roy, B. (1991), The outranking approach and the foundations of ELECTRE methods, *Theory and Decision* 31, 49-73.
- Roy B., D. Bouyssou (1993), *Aide multicritère à la décision: méthodes et cas*, Economica, Paris.
- Yu, W. (1992), ELECTRE TRI. Aspects méthodologiques et guide d'utilisation, *Document du LAMSADE*, No. 74, Université Paris-Dauphine.
- Zopounidis, C., D. Doumpos (2002), Multicriteria classification and sorting problems: a literature review, *European Journal of Operational Research* 138, 229-246.