

Instituto de Engenharia de Sistemas e Computadores de Coimbra
Institute of Systems Engineering and Computers
INESC – Coimbra

Jorge Maria

Escola Superior de Gestão – Instituto Politécnico de Santarém
jorgemaria@esgs.pt

João Coutinho-Rodrigues

Depº Engenharia Civil – Universidade de Coimbra e INESC-Coimbra
coutinho@dec.uc.pt

**Um Sistema Espacial de Apoio à Decisão Bicritério para
Planeamento de Rotas Turísticas**

No. 13

2003

ISSN: 1645-2631

Instituto de Engenharia de Sistemas e Computadores de Coimbra

INESC – Coimbra

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

www.inescc.pt

Resumo

No âmbito do turismo, a existência de informação georeferenciada sobre países ou locais a visitar (limites político-administrativos, localização de diversos equipamentos, rede viária, etc.), bem como a determinação de rotas “óptimas” entre eles, pode ser fomentada através da implementação de adequadas técnicas matemáticas, computacionais e multimédia, agrupadas num sistema de informação baseado em tecnologia SIG (Sistema de Informação Geográfica).

O sistema espacial de apoio à decisão apresentado neste trabalho tem como base o mapa digitalizado de uma cidade e da respectiva rede de ruas, bem como o conjunto de monumentos e pontos de interesse turístico, permite construir interactivamente percursos (rotas) apresentando as soluções mais interessantes quando, como pode acontecer na prática, se pretende a minimização simultânea de mais que um objectivo, como a extensão a percorrer e o tempo (ou outra função de custo) a despende para tal. Em geral esses objectivos são conflituosos (o percurso de comprimento mínimo entre dois pontos pode não ser o que corresponde ao menor tempo).

Palavras chave: Sistema de Apoio à Decisão, Análise Multicritério, Redes, Sistema de Informação Geográfica.

Abstract

Geo-referenced spatial information related to countries and cities, together with the location of interesting sites to be visited by tourists, are very important issues in tourism activities. Moreover, the evaluation of “optimal” paths linking interesting locations may be accomplished by means of adequate mathematical, computational and multimedia techniques integrated within a GIS environment.

We have developed an interactive decision support system (DSS) to make one city more competitive in attracting tourists (an application to a Portuguese city is presented). To accomplish this, we designed and implemented an interactive, user-friendly DSS that provides information about the city’s tourist attractions and allows tourists to use their time in the city efficiently. The system utilizes a bicriterion approach for evaluating compromise time/length solutions, within a geographical information system framework.

Keywords: Decision Support Systems, Multi-criteria Analysis, Networks, Geographical Information Systems

1. Introdução

As técnicas utilizadas neste trabalho, apoiam-se na teoria de grafos e optimização em redes. Desenvolveu-se um sistema espacial de apoio à decisão (ainda e apenas um protótipo) que, conjugando uma interface amigável com a implementação de complexos algoritmos, auxilia o utilizador na tomada de decisão. Ao apresentar os resultados sobre um mapa, o utilizador tem uma perspectiva mais realista daquilo que se propõe analisar ou decidir.

Iremos descrever o planeamento de percursos turísticos determinando rotas “optimizadas” tendo em conta, ou não, o binómio tempo/distância, para visitar os monumentos do centro histórico da cidade de Santarém.

A facilidade de utilização foi algo que esteve sempre presente. Assim o utilizador tem acesso a todas as potencialidades da aplicação através de simples botões colocados nas barras de ferramentas do *ArcView*[®] (vd Figura 1).



Figura 1 – Botões da aplicação

O utilizador não necessita de conhecimentos técnico-científicos da área em estudo, mas apenas de alguma destreza para utilização do rato em ambiente gráfico.

Os diálogos e/ou janelas disponibilizados pela aplicação são passíveis de ser mudados de posição, sobrepostos ou redimensionados a belo prazer do utilizador.

Ao passar com o rato sobre um dado botão, como por exemplo os da figura 1 surge uma curta indicação da acção que proporciona e uma descrição um pouco mais detalhada na barra de *status* do *ArcView*[®].

Ao entrar no projecto “Scalabis” é automaticamente activada a *View* com o mesmo nome e carregados os dados necessários ao normal funcionamento da aplicação. Sem qualquer acção directa por parte do utilizador, este é logo apresentado com o mapa da cidade com a respectiva localização dos monumentos (vd Figura 2).

Uma particularização dos problemas multiobjectivo é o caso dos problemas bi-objectivo, sendo um problema deste tipo que é tratado neste trabalho. Ou seja aqueles em que apenas dois objectivos concorrem para a determinação da solução.

Podemos ver em [Hansen 1980] que a determinação de todas as soluções não dominadas num problema de caminho mais curto bi-objectivo é, no pior dos casos, intratável porque o número de soluções cresce exponencialmente com o número de nós. Assim, perante um qualquer caso, o decisor seria apresentado com a enumeração exaustiva de todas as hipóteses possíveis de solução, inviabilizando uma escolha eficiente.

Para obviar este problema, a metodologia usada é a apresentada em [Coutinho-Rodrigues *et al* 1994 e 1998] que, utilizando um procedimento interactivo, possibilita ao decisor a orientação da sua pesquisa em função das suas preferências.

Este método começa por identificar um subconjunto de soluções não dominadas suportadas (i.e. pertence ao contorno convexo do conjunto das soluções). A busca de soluções eficientes não suportadas, que se encontram dentro do *gap* (vd figura 4), é realizada recorrendo a um algoritmo para determinação dos k caminhos mais curtos (kCMC), no nosso caso o algoritmo de Dreyfus.

Para calcular as soluções eficientes suportadas, do problema de caminho mais curto bi-objectivo, é utilizada uma soma ponderada das duas funções objectivo (ambas a minimizar), resolvendo o seguinte problema de caminho mais curto com um só objectivo [CMC1].

$$\text{[CMC1]} \quad \text{Min } \hat{Z} = w_1 Z_1 + w_2 Z_2 \quad (1)$$

sujeito a

$$\sum_{j \in N} X_{1j} = 1 \quad (2)$$

$$\sum_{i \in N} X_{ij} - \sum_{k \in N} X_{jk} = 0, \text{ (para todo o } j \in N \mid j \neq 1, n) \quad (3)$$

$$\sum_{i \in N} X_{in} = 1 \quad (4)$$

$$X_{ij} \in (0,1), \text{ (para todo o par } (i,j)) \quad (5)$$

onde

$$Z_1 = \sum_{i \in N} \sum_{j \in N} c_{ij} X_{ij} \quad (6)$$

$$Z_2 = \sum_{i \in N} \sum_{j \in N} d_{ij} X_{ij} \quad (7)$$

$$w_1 = \frac{|Z_{2\alpha} - Z_{2\beta}|}{|Z_{1\alpha} - Z_{1\beta}|} \quad (8)$$

$$w_2 = 1 \quad (9)$$

$$Z_{p\alpha} = \text{ao valor da } p^{\text{ésima}} \text{ função objectivo da solução eficiente } \alpha \quad (10)$$

(α e β são soluções adjacentes às soluções não dominadas)

$$X_{ij} = \begin{cases} 1, & \text{se o arco (i, j) pertence ao caminho} \\ 0, & \text{caso contrário} \end{cases}$$

c_{ij}, d_{ij} = "custos" (não negativos) associados aos arcos

N = conjunto dos nós da rede

nó 1 é o nó origem

nó n é o nó de destino

α, β - soluções não dominadas suportadas.

Assumimos que o primeiro objectivo Z_1 , é minimizar a distância total do caminho, enquanto que o segundo objectivo, Z_2 , é minimizar o tempo total que leva a percorrer esse mesmo caminho.

Os pontos A, B, C, S e R na figura 3 representam as soluções das funções objectivo para 5 caminhos não dominados.

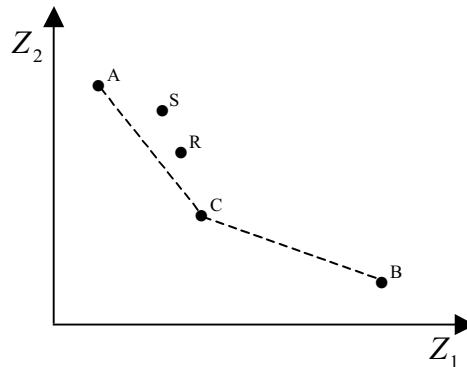


Figura 3 – Soluções não dominadas para um problema de caminho mais curto bi-objectivo (Z_1, Z_2)

Os pontos A, B e C (obtidos com o algoritmo de Dijkstra) representam a “fronteira eficiente” das soluções para o problema bi-objectivo, sendo que o ponto A representa o caminho que minimiza a distância (caminho mais curto) e o ponto B o caminho que minimiza o tempo (caminho mais rápido). O ponto C é uma solução de compromisso obtida tendo por base os pesos calculados como indicado em (8) de CMC1, em que α e β representam os pontos A e B.

Se nenhuma destas soluções satisfizer o decisor, este poderá solicitar a determinação de novas soluções de compromisso, indicando dois pontos que definirão uma zona de pesquisa para determinação das mesmas. Fornecendo assim os novos valores de α e β que permitem calcular w_1 e w_2 de modo a encontrar (como realizado para a solução C), quando existir, uma eventual nova solução.

Infelizmente este procedimento não permite identificar as soluções eficientes não suportadas como as representadas pelos pontos S e R da figura 3 [Cohon 1978], [Henig 1985], [Coutinho-Rodrigues *et al* 1998]

O procedimento utilizado para determinar essas soluções - que caem dentro do *gap* (vd figura 4)-, como já referimos, engloba um algoritmo para determinação dos kCMC que terminará, no pior dos casos, quando atingir um dado limite superior. Este procedimento será explicado de seguida.

“GAP’s” apoia-se na resolução dos kCMC tendo por base as soluções iniciais determinadas em [CMC1]. α e β são soluções suportadas do problema original bi-objectivo representadas, por exemplo, pelos pontos A e C da figura 4. Neste caso “APC” é o *gap* onde irão ser pesquisadas novas soluções não dominadas. O ponto P e os valores Z_{1P} e Z_{2P} permitem-nos identificar o primeiro limite superior para o algoritmo de kCMC. Note-se que $Z_{1P} = Z_{1C}$ e que $Z_{2P} = Z_{2A}$.

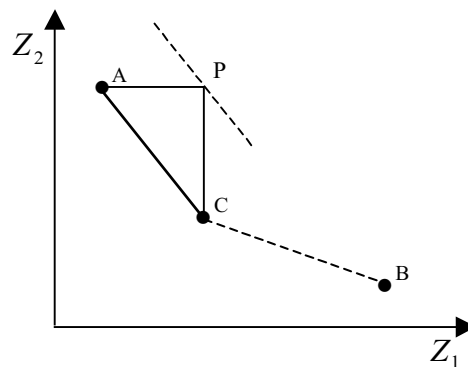


Figura 4 – O *gap* “APC”

w_1 e w_2 foram definidos de modo a obter \hat{Z} (em CMC1) representando a linha que passa por \overline{AC} . O algoritmo de kCMC começa a pesquisa dentro do *gap* e encontra a primeira solução, o ponto R na figura 5

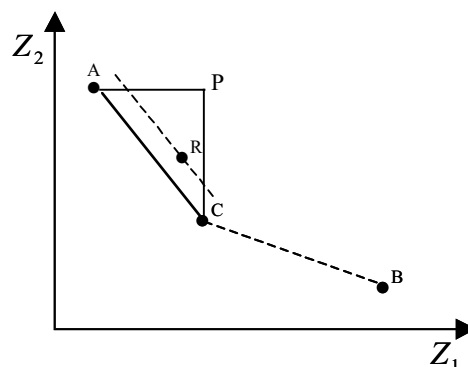


Figura 5 – A primeira solução encontrada (R) dentro do *gap*

Note-se que a primeira solução admissível deverá respeitar o limite superior do *gap*. Assim esta solução encontrando-se dentro do *gap* é, por definição de solução não dominada, não dominada.

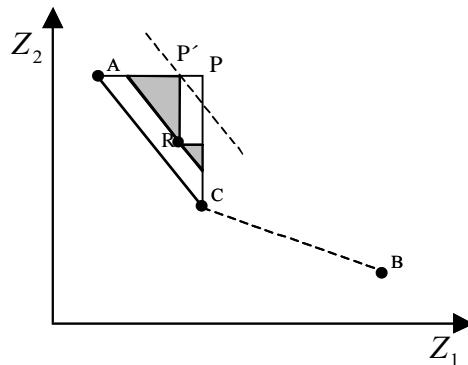


Figura 6 – Limite superior para \hat{Z} e área, no *gap*, para eventuais novas soluções não dominadas

Ainda por definição de solução não dominada, eventuais novas soluções não dominadas dentro do *gap* deverão situar-se nas zonas sombreadas (vd figura 6). Obtendo-se novos limites superiores para \hat{Z} , como é o caso do ponto P' na mesma figura.

Parece-nos agora evidente como continuaria a pesquisa passo a passo dentro do *gap*, diminuindo a zona com possíveis soluções não dominadas, cujo próximo ponto a encontrar poderia ser, por exemplo, o ponto S da figura 3. O processo terminará quando não existirem mais soluções não dominadas dentro do *gap* ou, em alternativa, quando o decisor, por se encontrar satisfeito com a solução encontrada, assim o desejar.

2.1. Resolução interactiva do Problema bi-objectivo

Para iniciar a determinação de um caminho mais curto bi-objectivo o utilizador só tem de “cliquear” no botão indicado na figura 7.



Figura 7 – Botão para problema bi-objectivo interactivo

Ao fazer isto surge no ecrã o diálogo “GAP’s” onde, por enquanto, só é possível especificar a identificação do nó de onde se pretende sair e, posteriormente, aquele que se pretende alcançar (vd Figuras 8 e 9). Esta selecção é realizada através de dois menus do tipo *drop-down* que disponibilizam uma listagem com a identificação dos vários monumentos.

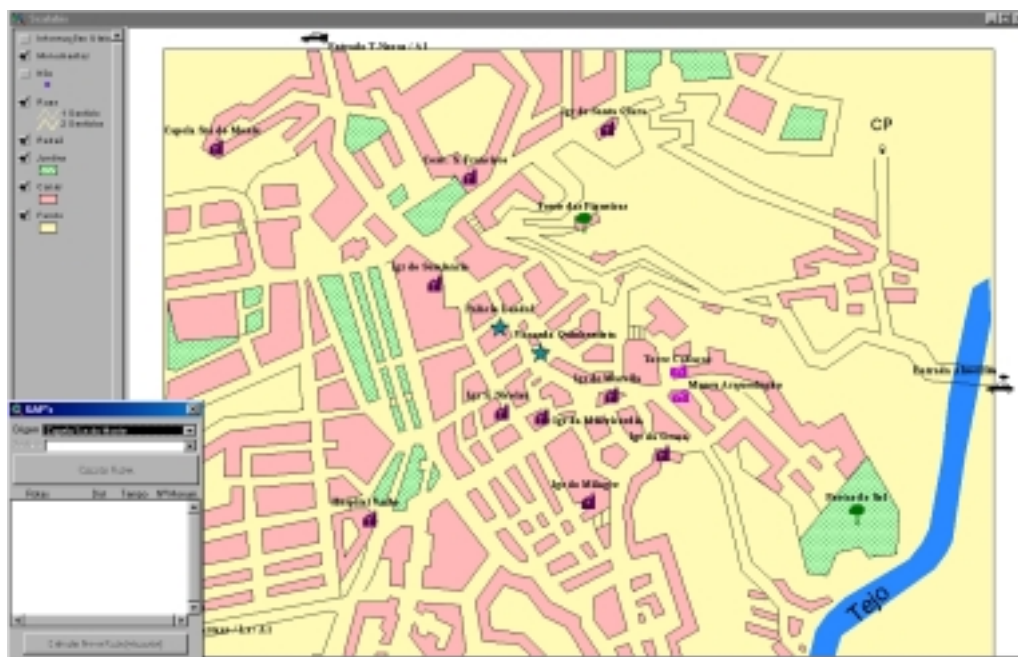


Figura 8 – Seleccionar origem

É de salientar que ao fazer a selecção dos monumentos, nos menus, aqueles são “marcados” no mapa, de modo a auxiliar o utilizador na certificação do monumento seleccionado (mesmo que o utilizador se “arrependa” e faça uma segunda ou mais escolhas), ● para a origem e ● para o destino. Note-se ainda que, para evitar “descuidos” por parte do utilizador, depois de seleccionar a origem, o nome deste monumento deixa de estar disponível na lista de selecção do destino.

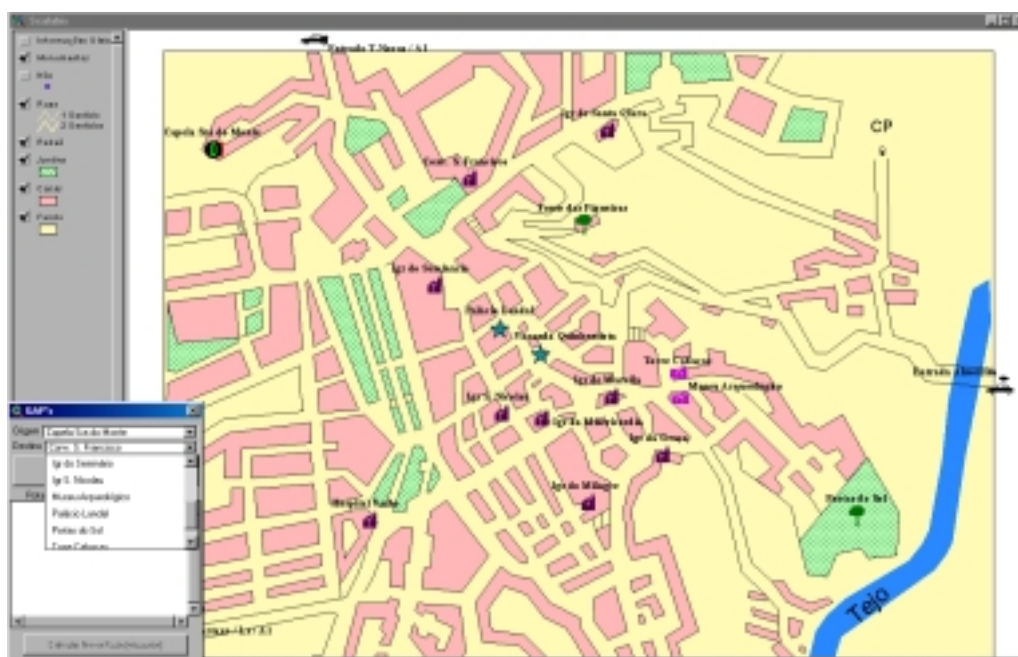


Figura 9 – Seleccionar destino

Feitas as seleções, fica disponível o botão “Calcular Rotas”. Ao “clique” neste botão a aplicação devolve as três primeiras soluções, que correspondem à rota mais curta, mais rápida e uma “ponderada” (vd Figura 10).

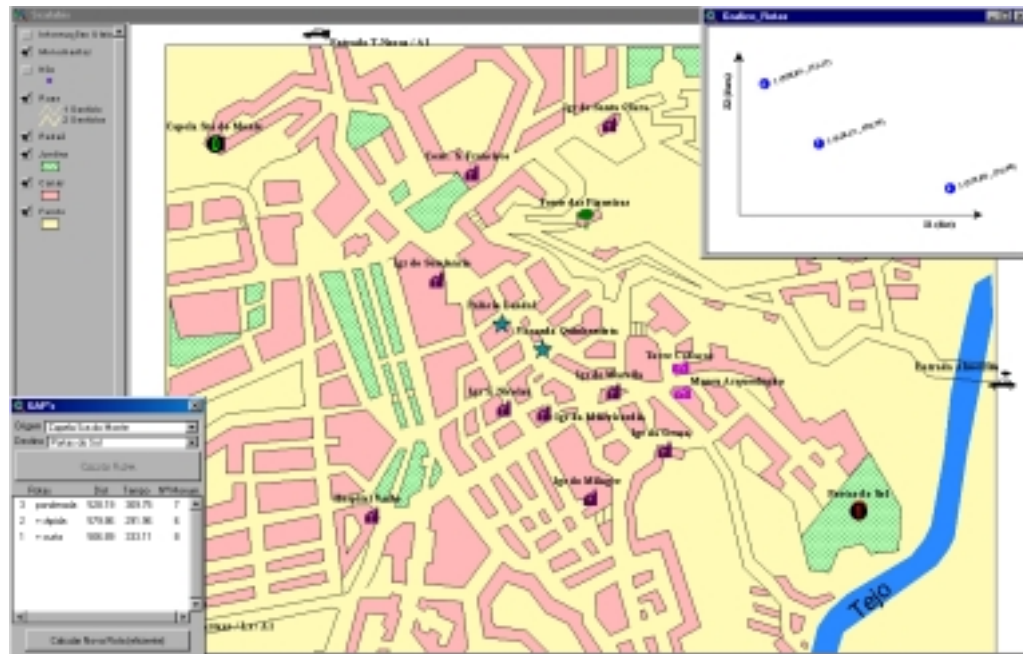


Figura 10 – Exibição das três primeiras rotas calculadas

As diversas soluções calculadas vão sendo apresentadas na *list box* do diálogo “GAP’s” e simultaneamente graficamente numa janela de nome “Gráfico Rotas” (vd Figura 10). Em relação a cada rota, ficamos a saber: a distância, o tempo de travessia e o número de monumentos que inclui. No gráfico, este último item, surge em forma de numeral sobre cada ponto representativo de uma rota (vd Figura 15).

Caso se pretenda visualizar, no mapa, uma das rotas calculadas basta fazer um “duplo clique”, na *list box*, sobre a rota em causa. Surgindo também no ecrã um novo diálogo que, além de indicar as ruas que a rota atravessa, possibilita a impressão do mapa com a rota marcada (vd Figura 11). Note-se ainda que o gráfico também reflecte a rota exibida no mapa.

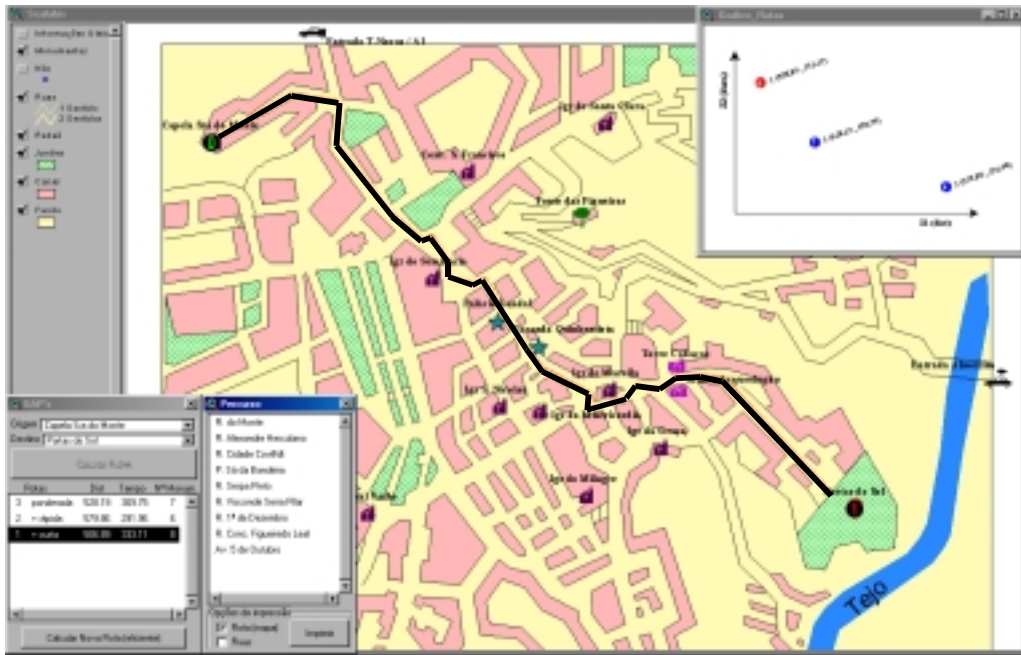


Figura 11 – Rota mais curta

Como se pode constatar na figura 12 ao pretendemos visualizar a rota correspondente à solução 2 (a mais rápida), o conteúdo do diálogo “Percurso” e do gráfico são actualizados em conformidade. Em relação a este último, podemos desactivá-lo ou voltar a activá-lo através do botão indicado na figura 13.

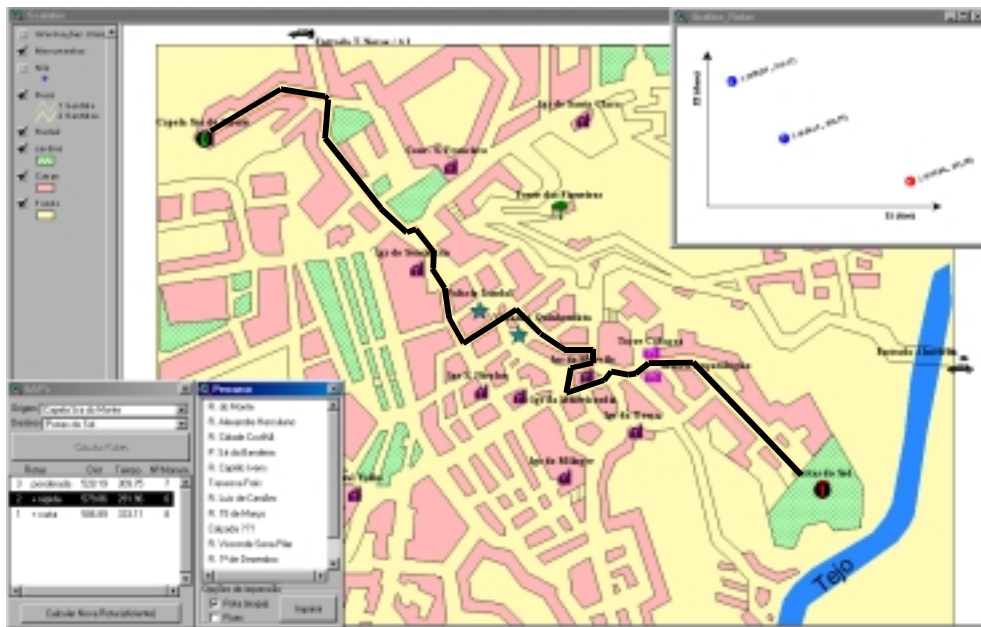


Figura 12 – Rota mais rápida



Figura 13 – Activa/desactiva gráfico

Caso se pretenda conhecer a localização de uma das ruas (no seu comprimento total e não apenas no segmento correspondente à rota), apresentadas no diálogo “percurso”, basta “clique” em cima do seu nome para que esta seja “iluminada” no mapa (vd Figura 14).

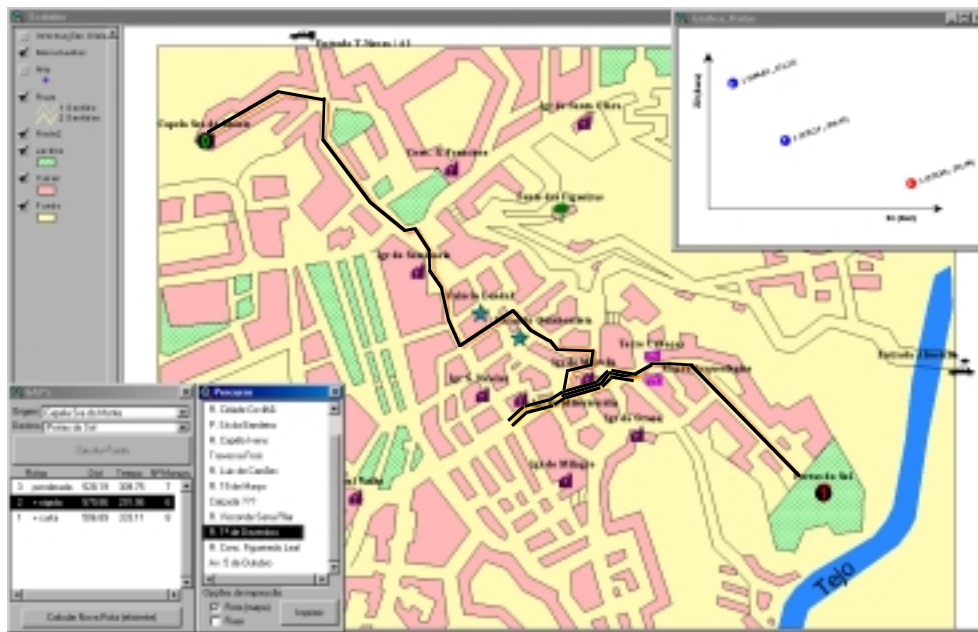


Figura 14 – Rua “iluminada”

Se o utilizador se considerar satisfeito com alguma das soluções apresentadas pode ficar por aqui. Podendo optar por imprimir o mapa com a rota seleccionada e/ou as ruas que a rota atravessa. Caso contrário o processo continuará como segue:

O utilizador deverá, com o auxílio da tecla “shift”, seleccionar, na *list box* do diálogo “GAP’s”, duas rotas e de seguida “clique” no botão “Calcular Nova Rota”, isto indica que pretende uma nova rota que se situe entre as duas seleccionadas.

Por exemplo, se se seleccionarem as rotas 2 e 3, significa que estamos dispostos, no nosso caso e em relação à solução 1, a “andar” mais desde que isso se traduza numa maior rapidez do percurso. Ou seja perdemos num dos objectivos para ganharmos no outro.

Ao “clique” no botão “Calcular Nova Rota”, a aplicação vai tentar encontrar uma nova solução que pertença à fronteira eficiente do conjunto das soluções (como já tinha acontecido para a determinação da solução número 3). Se tal rota existir (o que aconteceu no nosso exemplo) o *Script* é interrompido passando tal solução a figurar tanto no diálogo “GAP’s” como no gráfico (vd Figura 15). Este último permite-nos uma análise mais rápida sobre aquilo que ganhámos e perdemos em cada uma das funções objectivo.

No caso de não existir nenhuma solução sobre a fronteira eficiente do conjunto das soluções, a aplicação passa a “varrer” o gap, delimitado pelas rotas indicadas, com um algoritmo de *k* caminhos mais curtos na busca de novas soluções não dominadas.

As soluções, caso existam, são exibidas num novo diálogo (para evitar confusões quanto à delimitação de novos intervalos de pesquisa) de nome “Rotas Eficientes”. Este diálogo tem o mesmo tipo de comportamento que o diálogo “GAP’s” quanto à visualização das rotas.

No nosso exemplo apenas foi encontrada uma nova rota não dominada (vd Figura 16)

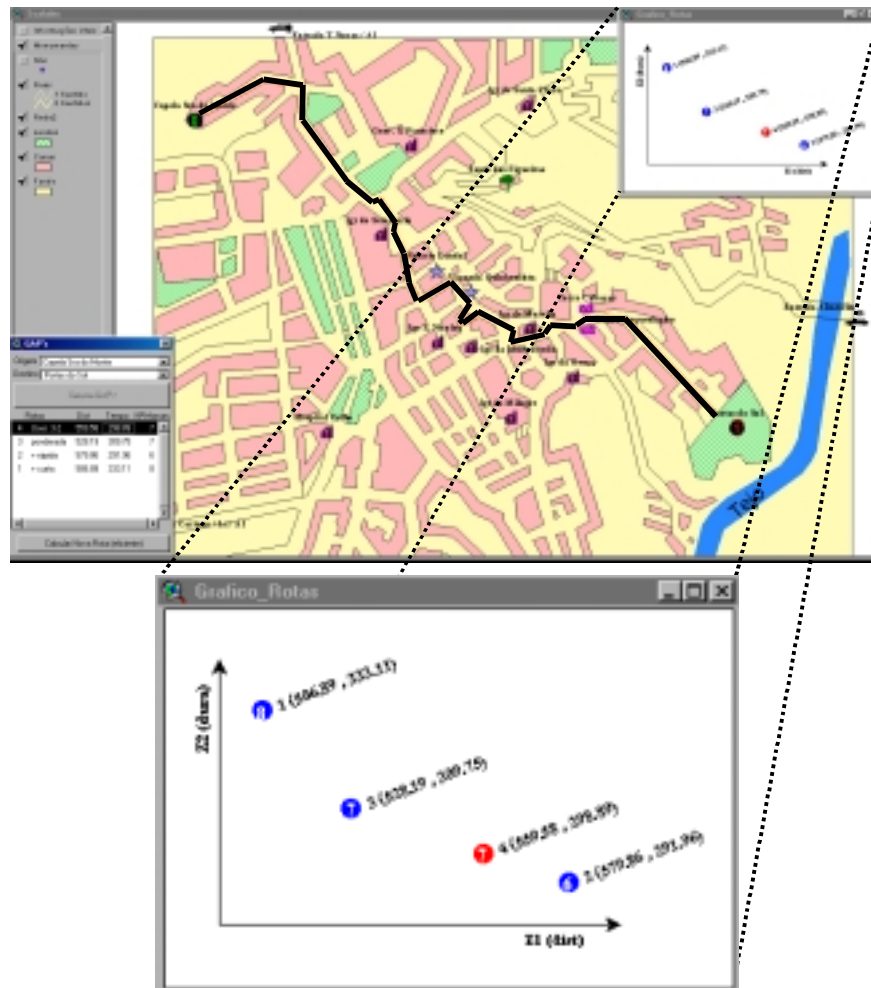


Figura 15 – Nova rota posicionada sobre a fronteira eficiente

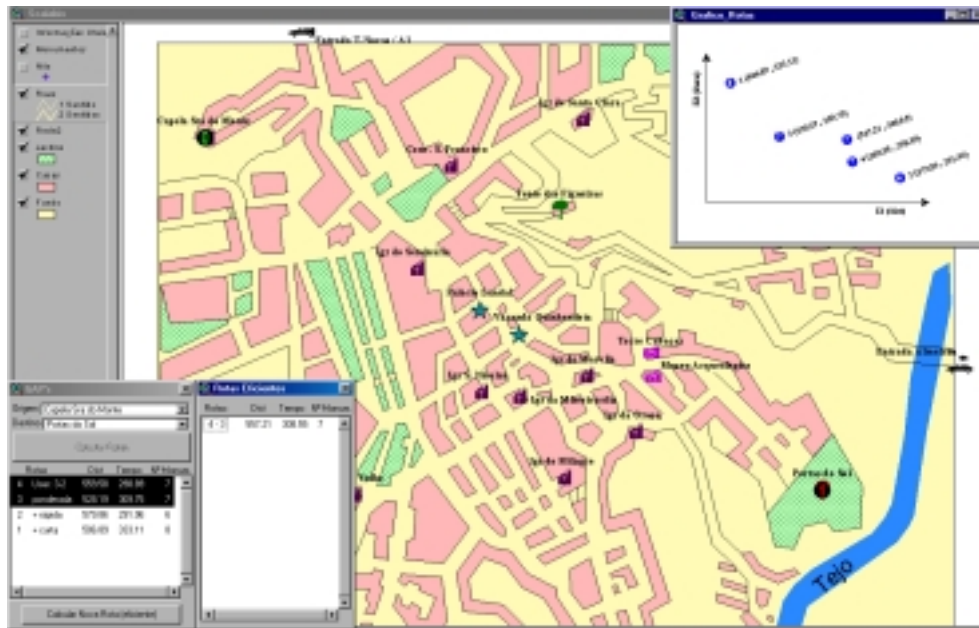


Figura 16 – Rota pesquisada dentro do gap

Se o decisor fosse inconsistente nas suas decisões (este, além da eventual perda de tempo por parte do decisor, é um dos problemas apontados aos métodos interactivos [Cohon 1978] [Zionts e Wallenius 1976]) e escolhesse agora procurar novas rotas entre as soluções 1 e 3 (desta feita abdicando do tempo em relação à distância), no nosso exemplo, seria apenas encontrada mais uma solução. Passando a ter à sua disposição todas as rotas não dominadas possíveis de determinar entre os dois monumentos indicados (vd Figura 17).

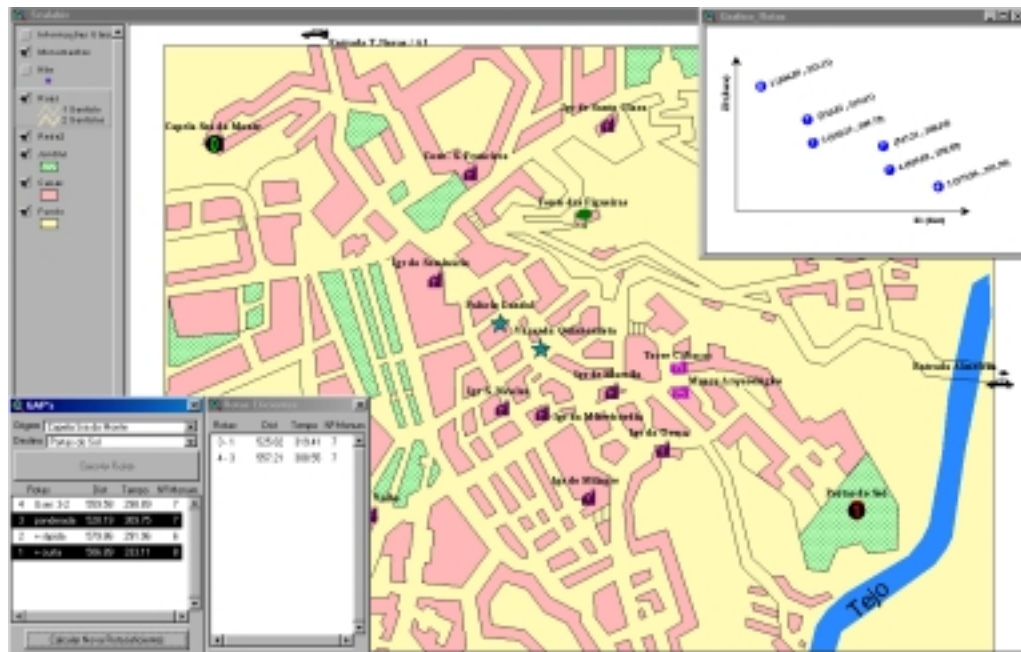


Figura 17 – Exemplo de problema com todas as rotas não dominadas possíveis

2.2. Outros métodos para o problema bi-objectivo

[Cohon 1978], classificou os métodos para resolução de problemas multi-objectivo em duas categorias: “métodos geradores” e “métodos agregadores”. O primeiro método premeia o decisor com todas as soluções não dominadas possíveis (ou uma aproximação destas), para que escolha, entre todas as possibilidades, aquela que melhor representa a sua vontade. Cohon é o próprio a reconhecer que o grande inconveniente desta abordagem é o eventual esforço computacional.

Quanto ao segundo método, são solicitadas à priori as preferências (prioridades, pesos relativos, etc.) do decisor em relação a cada um dos objectivos, para depois resolver o problema mono-objectivo que optimiza aquelas preferências. Contudo, Cohon sustenta que é irrealista pedir-se ao decisor que quantifique as suas preferências sem que este saiba quais os custos que isso envolve. Esta ideia também é defendida em [Zionts e Wallenius 1976] ao alegarem que parece mais fácil, para os decisores, responder a determinadas questões sobre os custos envolvidos, estando num contexto em que resultam de um caso concreto, em vez de situações abstractas.

Mesmo assim, aproveitando todo o trabalho já desenvolvido em termos de programação, optou-se por também disponibilizar, nesta aplicação, estes dois tipos de abordagem. Assim para aceder ao primeiro método basta “clique” no botão indicado na figura 18.



Figura 18 – Botão que permite aceder ao método gerador

Este botão permite aceder ao diálogo “GAP’s” (já visto na secção anterior: figuras 8, 9 e 10) mas tendo em conta que:

- ♦ o botão “Calcular Rotas” manda pesquisar toda a área delimitada pelas duas rotas que optimizam cada um dos objectivos individualmente; as soluções aparecem todas na *list box* do diálogo. Todas as outras acções (selecção, visualização, etc.) têm o mesmo procedimento;
- ♦ o botão “Calcular Nova Rota” não fica activo, pois como vimos no ponto anterior o utilizador não volta a interagir com a aplicação nesse domínio.
- ♦ o número de soluções admissíveis (não dominadas) e não admissíveis (dominadas) vão surgindo na barra de estado do *ArcView*[®], juntamente com um botão de “Stop” que permite ao utilizador parar a pesquisa antes desta ter chegado ao fim.

Para aceder ao segundo método basta “clique” no botão indicado na figura 19 que permite aceder ao diálogo apresentado na figura 20.



Figura 19 – Botão que permite indicar pesos antes de otimizar a rota



Figura 20 – Aplicação de pesos

O facto do nosso problema ser bi-objectivo facilita a definição dos pesos a aplicar a cada um dos objectivos. Como se assume que, se pretendemos melhorar um deles teremos de piorar o outro, a aplicação dos pesos é feita deslizando o “*slider*” para o lado do objectivo que pretendemos privilegiar. Isto fará decrescer proporcionalmente o peso a atribuir ao outro objectivo (menor grau de importância). Estas deslocações são possíveis de realizar em intervalos de 5 pontos percentuais.

É de salientar que se atribuirmos o peso todo a um dos objectivos, distância ou tempo, (vd figura 21 e 22) teremos respectivamente o caminho mais curto ou o caminho mais rápido.



Figura 21 – Calcular CMC

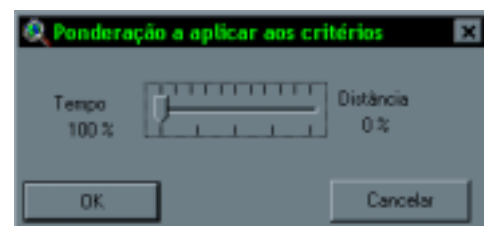


Figura 22 – Calcular caminho mais rápido

De seguida é solicitada a escolha da origem e do destino, através de duas caixas de listagem. Na figura 23 apresentamos a que permite seleccionar a origem. A única diferença em relação à segunda é esta não possuir o monumento origem previamente seleccionado.

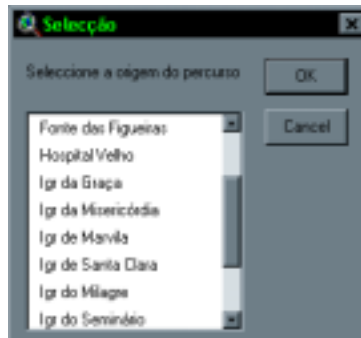


Figura 23 – Caixa listagem

Ao premir “Ok”, depois de seleccionado o destino, a rota é calculada e identificada no mapa (vd Figura 24). Surgindo também no ecrã o já nosso conhecido diálogo “percurso”, cujas potencialidades já foram descritas.

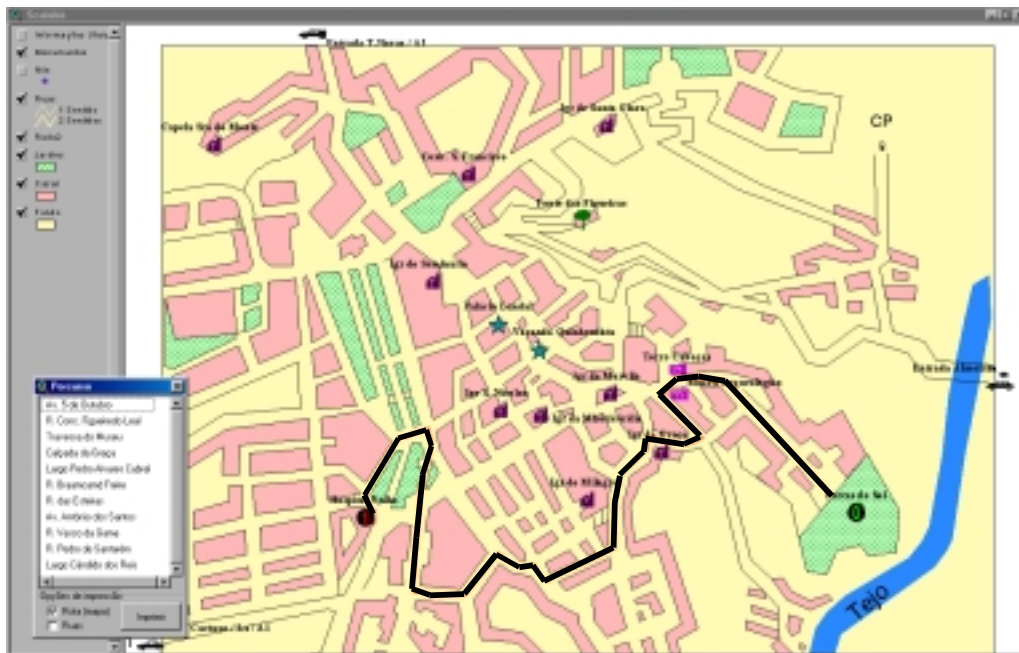


Figura 24 – Mapa com resolução de um problema mono-objectivo (Tempo = 100%)

3. Conclusões

No âmbito do turismo, a existência de informação georeferenciada sobre países ou locais a visitar (limites político-administrativos, localização de diversos equipamentos, rede viária, etc.), bem como a determinação de rotas “óptimas” entre eles, pode ser fomentada através da implementação de adequadas técnicas matemáticas, computacionais e multimédia, agrupadas num sistema de informação e promoção turística baseado em tecnologia SIG.

As técnicas utilizadas neste trabalho, apoiam-se na teoria de grafos, optimização em redes e suportam a resolução do problema do caminho mais curto bi-objectivo. Foi desenvolvido um protótipo que conjuga uma interface amigável (através da qual o utilizador interage facilmente sem necessidade de grandes conhecimentos técnicos) com a implementação de complexos algoritmos para resolução daquele problema (outros problemas poderão vir a ser incluídos, como por exemplo: “*Vertex Constrained Shortest Path*”, “*Constrained Shortest Path*” ou a criação de um “*meeting point*”).

Desenvolveu-se, um protótipo que auxilia o utilizador na tomada de decisão, acrescido do facto, dos resultados serem apresentados sobre um mapa, o que dá ao utilizador uma perspectiva mais realista sobre aquilo que se propõe analisar ou decidir.

4. Referências

Cohon, *Multiobjective Programming and Planning*, Academic Press, 1978.

Coutinho-Rodrigues, Clímaco, Current, “A PC-based interactive decision support system for two objective direct delivery problems”, *Journal of Business Logistics*, 15, nº1 (1994), pp. 305-322.

Coutinho-Rodrigues, Clímaco, Current, “An interactive bi-objective shortest path approach: searching for unsupported nondominated solutions”, *Computers and Operations Research*, 26, 789-798, 1999.

Hansen, “Bicriterion path problem”, *Lecture Notes in Economics and Mathematical Systems*, 177 (1980), pp. 109-127.

Henig, “The shortest path problem with two objective functions”, *European Journal of Operational Research* 2 (1985), pp. 281-291.

Zionts e Wallenius, “An interactive programming method for solving the multiple criteria problem”, *Management Science*, 22 (1976), pp. 652-663.