

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

Luis A. Santos e João M. Coutinho-Rodrigues

**Uma Nova Abordagem para o Cálculo do Caminho Mais Curto
com uma Restrição Adicional Baseada no Método NISE
Bi-objectivo**

No. 6

2004

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

Uma Nova Abordagem para o Cálculo do Caminho Mais Curto com uma Restrição Adicional Baseada no Método NISE Bi-objectivo

Luís Santos
Instituto Superior Bissaya Barreto, Bencanta, 3040 Coimbra, Portugal
lsantos@issb.pt

João Coutinho-Rodrigues
Departamento de Engenharia Civil, Faculdade de Ciências e Tecnologia,
PoloII, Universidade de Coimbra, 3030 Coimbra, Portugal
coutinho@dec.uc.pt

Abstract

The Shortest Path problem (SP) is one of the most important sub-problems related with real networks. This problem has been strongly studied and we can find in literature numerous algorithms that solve it in polynomial time. However, if an additional constraint is considered in the SP problem, the new problem, known as the Constrained Shortest Path problem (CSP), is at least NP-Complete [15]. To solve it, Handler and Zang [15] proposed an algorithm, based on Lagrangian relaxation, that has been accepted as the best one. In this paper we present a new technique to solve the CSP, based on the NonInferior Set Estimation method (NISE) [6,7]. When compared to Handler and Zang' algorithm [15], this approach is easier to understand and implement, and the results obtained are exactly the same, both in terms of number of SP solved and in number of iterations for k Shortest Path problem (k SP) performed.

Resumo

O problema do Caminho Mais Curto (CMC) constitui um dos sub-problemas mais resolvidos em problemas de optimização relacionados com redes reais. A sua complexidade computacional é polinomial e é possível encontrar na literatura diversos algoritmos, e diferentes implementações desses algoritmos, que o permitem resolver. Ao adicionar-se uma nova restrição ao CMC, o novo problema, designado por Caminho Mais Curto com Uma Restrição Adicional (CMCRA), torna-se pelo menos NP-Completo [15]. Um dos algoritmos mais referidos na literatura para tratar o CMCRA foi proposto por Handler e Zang [15] e baseia-se na relaxação Lagrangeana aplicada ao modelo de programação linear representativo do problema. Neste relatório de investigação apresenta-se uma nova técnica para resolver o CMCRA baseada no método designado por *NonInferior Set Estimation method* (NISE) [6,7]. Quando comparada com a abordagem apresentada por Handler e Zang [15], esta técnica é conceptualmente mais simples (portanto, mais fácil de entender) e mais fácil de implementar. Os resultados obtidos são

exactamente os mesmos, seja em termos do número de CMC que é necessário resolver, seja relativamente ao número de iterações do problema dos k Caminhos Mais Curtos (k CMC) que também há que determinar.

Palavras chave: Redes, Caminho óptimo, Restrição adicional, NISE

1. Introdução

O problema da determinação do CMC, no qual se procura o caminho óptimo entre dois nós ou entre um nó e todos os outros, constitui um dos sub-problemas mais resolvidos em redes, nomeadamente em problemas de transportes e circulação, existindo vários algoritmos de complexidade polinomial, bem como sucessivos melhoramentos de alguns deles (em geral baseados em variantes das estruturas de dados usadas) [5,22]. A introdução de uma restrição adicional neste problema (por exemplo, minimizar um comprimento total a percorrer impondo um tempo máximo de percurso que não deve ser ultrapassado) torna-o muito mais complexo sob o ponto de vista computacional. De facto, Handler e Zang [15] mostraram que o problema da determinação do CMCRA é, pelo menos, NP-Completo. Tal como no CMC, as aplicações reais do CMCRA são várias e estão, nomeadamente, associados a problemas de circulação em redes [18].

Intimamente relacionado com o problema da determinação do CMC aparece o da determinação dos k CMC entre dois nós. Neste caso interessa determinar os primeiros k caminhos óptimos, de modo a obter uma sucessão de caminhos ordenada de forma crescente segundo o valor da função a minimizar. Vários algoritmos têm sido propostos para este tratar este problema [2,12,17,20,21].

Uma das abordagens possíveis para tratar o problema do CMCRA consiste em utilizar um algoritmo que determine os k CMC e que termina assim que o caminho obtido satisfaça a condição adicional. A principal desvantagem deste processo é o elevado número de iterações do algoritmo em determinados problemas. Outras abordagens podem ser encontradas na literatura, tais como, a apresentada por Johsch [16], baseada em programação dinâmica, e a apresentada por Handler e Zang [15], baseada na relaxação Lagrangeana. Esta última abordagem é uma das mais referidas na literatura para o tratamento do problema.

Neste relatório de investigação apresentar-se-á uma nova abordagem para tratar o CMCRA, baseado no método NISE [6,7]. Numa primeira fase, determinam-se algumas das soluções situadas no contorno convexo do conjunto das soluções admissíveis para o CMC. Este contorno convexo, que passará a ser designado por φ , é constituído pela linha poligonal cujos vértices são os pontos que correspondem às soluções que se obtêm minimizando a função objectivo $c_{ij} + \mu t_{ij}$ (c_{ij} e t_{ij} serão definidos na secção 2), para qualquer valor não negativo do parâmetro real μ , considerando o sistema de eixos cartesiano bidimensional (f_1, f_2) (f_1 e f_2 serão definidos na secção 3). Posteriormente, no caso de não se ter

encontrado ainda a solução óptima do CMCRA, utiliza-se um algoritmo que determine os k CMC, considerando ainda a soma pesada da função objectivo original com a função que traduz a restrição adicional, para pesquisar no interior do conjunto limitado inferiormente pelo contorno convexo φ .

2. Formulação Matemática do Problema

Considere-se o grafo orientado $G = (N, A)$, onde $N = \{1, 2, \dots, n\}$ representa o conjunto dos nós e $A = \{(i, j) : i, j \in N, i < j\}$ representa o conjunto dos arcos orientados. A cada arco orientado (i, j) estão associados dois valores inteiros não negativos c_{ij} e t_{ij} que podem representar, por exemplo, o comprimento do arco e o tempo de atravessamento do arco (para nos referirmos genericamente ao valor associado a cada arco utilizar-se-á a designação “custo”).

O problema do CMCRA entre os nós 1 (nó que representa a origem) e n (nó que representa o destino) pode ser definido sob a forma de um modelo de programação linear do seguinte modo:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

sujeito a

$$\sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} 1 & \text{se } i = 1 \\ 0 & \text{se } i = 2, \dots, n \\ -1 & \text{se } i = n \end{cases} \quad (2)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq T \quad (3)$$

$$x_{ij} \in \{0, 1\}, (i, j) \in A, \quad (4)$$

onde a equação (3) representa a restrição adicional imposta ao problema do CMC (com T um parâmetro real), constituído pelas restrições (1), (2) e (4). As variáveis x_{ij} , associadas a cada um dos arcos, são definidas do seguinte modo: se o arco (i, j) pertencem ao caminho então $x_{ij} = 1$; caso contrário, $x_{ij} = 0$.

3. Descrição do Algoritmo de Handler e Zang

Seja $X = [x_{ij} : (i, j) \in A]$ um vector cujas componentes são x_{ij} , para todo $(i, j) \in A$, e Y o conjunto de vectores X que satisfazem as equações (2) e (4). Considerem-se as seguintes funções, para qualquer $X \in Y$:

$$f_1(X) = \sum_{(i,j) \in A} c_{ij} x_{ij}, \quad (5)$$

$$f_2(X) = \sum_{(i,j) \in A} t_{ij} x_{ij} \quad (6)$$

e

$$f_3(X) = \sum_{(i,j) \in A} t_{ij} x_{ij} - T. \quad (7)$$

No algoritmo proposto por Handler e Zang [15] a abordagem é feita com base na relaxação Lagrangeana. Assim, o problema do CMCRA, pode ser reescrito na seguinte forma:

$$f^* = f_1(X^*) = \min_{X \in Y} f_1(X) \quad (8)$$

sujeito a

$$f_3(X) \leq 0, \quad (9)$$

onde X^* representa a solução óptima e f^* o valor óptima da função objectivo.

A metodologia proposta consiste em relaxar a restrição (9). Para qualquer $u \in R$ e $X \in Y$ define-se a função Lagrangeana

$$L(u, X) = f_1(X) + u f_3(X) \quad (10)$$

e

$$L(u) = \min_{X \in Y} L(u, X). \quad (11)$$

Conforme referem os autores, é bem conhecida a propriedade

$$L(u) \leq f^* \quad \forall u \geq 0. \quad (12)$$

Isto significa que para se obter um limite inferior de boa qualidade para o valor da função objectivo do problema primal pode resolver-se o problema dual definido por:

$$L^* = L(u^*) = \max_{u \geq 0} L(u). \quad (13)$$

De referir que embora o dual seja um programa convexo, o mesmo não se passa relativamente ao programa primal, pelo que pode existir um intervalo de dualidade entre eles, ou seja, $L^* < f^*$.

Na abordagem utilizada pelos autores, determinam-se inicialmente duas soluções óptimas para a equação (11), considerando respectivamente $u=0$ e $u=+\infty$ (note-se que a resolução desta equação corresponde a determinar o caminho óptimo na rede original, considerando o custo em cada arco (i, j) igual a $c_{ij} + u t_{ij}$). Na fase seguinte, onde se resolve o problema dual, em cada iteração o valor do parâmetro u é actualizado, e posteriormente é determinado o caminho óptimo para este novo valor do parâmetro. Esta fase termina com a obtenção da solução óptima do dual. Na última fase, analisa-se o intervalo de dualidade, caso exista, através de um algoritmo que determine os k CMC.

4. Algoritmo baseado no método NISE

4.1 Descrição do método NISE

O método NISE, descrito por Cohon [6] e por Cohon *et. al.* [7], permite obter uma aproximação para o conjunto das soluções não-dominadas em problemas multiobjectivo (uma solução é “não-dominada” ou “eficiente” se não existir qualquer outra solução admissível que melhore um dos objectivos sem piorar, pelo menos, um dos outros), determinando as que se encontram em vértices do contorno convexo das soluções não-dominadas. Este processo apenas se pode aplicar quando as funções objectivo são lineares. Por uma questão de simplificação e também porque é a situação que se adequa ao tratamento do CMCRA, a descrição do método apenas será feita para o caso biobjectivo (para este tipo de problemas as soluções podem ser representadas por pontos num sistema de eixos cartesianos). Para determinar as soluções não-dominadas o método utiliza propriedades dos segmentos de recta que unem os pontos que as representam.

Suponhamos que se pretende a minimização de dois objectivos expressos pelas funções Z_1 e Z_2 . No método NISE as soluções não-dominadas são obtidas com base na soma pesada das funções objectivo, que constitui uma das técnicas para determinar soluções não-dominadas. Considerando o caso geral em que se multiplica cada uma das funções objectivo por um determinado coeficiente, que representa o peso desse objectivo relativamente aos restantes, a nova função objectivo a minimizar é expressa por:

$$\min Z(w_1, w_2) = w_1 Z_1 + w_2 Z_2. \quad (14)$$

Como a função objectivo (14) pode ser multiplicada por qualquer número positivo, sem que esta operação modifique a solução, então multiplicando a equação por $1/w_1$ tem-se:

$$\min Z(w_1, w_2) = Z_1 + \frac{w_2}{w_1} Z_2. \quad (15)$$

Considerando um novo peso $w = w_2/w_1$, conclui-se que qualquer equação do tipo (14) pode ser reescrita do seguinte modo:

$$\min Z(w) = Z_1 + wZ_2. \quad (16)$$

Para iniciar a aplicação do método de NISE, determinam-se as duas primeiras soluções não-dominadas, representadas na Figura 1 pelos pontos A e B (obtidos através da minimização respectivamente de Z_1 e Z_2 isoladamente).

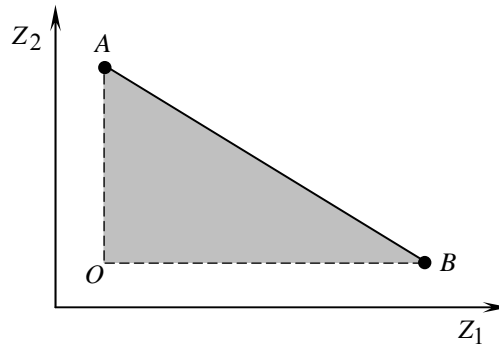


Figura 1 – Aplicação do NISE em problemas biobjectivo; representação das duas primeiras soluções não-dominadas.

A próxima solução não-dominada situada num vértice do contorno convexo das soluções não-dominadas, caso exista, situa-se no interior do triângulo $[AOB]$ e pode ser determinada com base da inclinação do segmento de recta \overline{AB} , através da expressão

$$\lambda = \frac{Z_2(A) - Z_2(B)}{Z_1(A) - Z_1(B)}. \quad (17)$$

A escolha do peso w é obtido através da expressão

$$w = -1/\lambda = \frac{Z_1(B) - Z_1(A)}{Z_2(A) - Z_2(B)}, \quad (18)$$

pelo que a função objectivo passa a ter a expressão

$$\min Z(A, B) = Z_1 + \frac{Z_1(B) - Z_1(A)}{Z_2(A) - Z_2(B)} Z_2. \quad (19)$$

Note-se que o valor do parâmetro λ é sempre negativo, por isso, o valor do peso w é positivo. Com este peso, a próxima solução não-dominada determinada pelo método (assumindo que é diferente de A e B), representada na Figura 2 pelo ponto C , situa-se dentro do triângulo $[AOB]$, abaixo do segmento de recta \overline{AB} e o mais afastado possível deste segmento de recta.

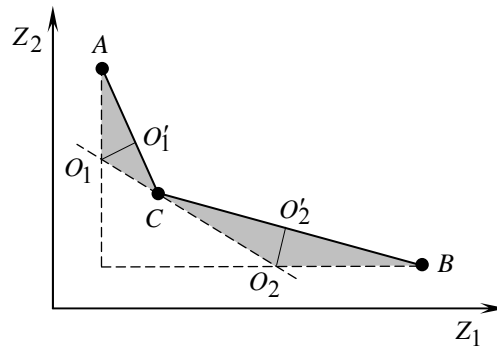


Figura 2 – Aplicação do NISE em problemas biobjectivo; situação após o cálculo da terceira solução não-dominada.

Com base nas três soluções não-dominadas já calculadas (A , B e C), é possível determinar duas novas soluções não-dominadas, caso existam, utilizando a mesma metodologia baseada agora respectivamente nas inclinações dos segmentos de recta \overline{AC} e \overline{BC} . A condição de paragem do método, consiste em definir o tamanho máximo das regiões que podem ficar por pesquisar, ou seja, impor um limite máximo para os comprimentos dos segmentos de recta $\overline{O_1O_1'}$ e $\overline{O_2O_2'}$ (onde O_1' e O_2' representam respectivamente a projecção perpendicular dos pontos O_1 e O_2 nos segmentos de recta \overline{AC} e \overline{BC}). Se, por exemplo, a solução não-dominada obtida com base na inclinação de \overline{BC} for igual à representada por A ou C , significa que no triângulo $[AO_1C]$ não existem não-dominadas situadas num vértice do contorno convexo das soluções não-dominadas.

4.2 Descrição do algoritmo

A abordagem proposta para determinar a solução óptima do CMCRA, consiste numa primeira fase em aplicar o método NISE para determinar soluções que se situam no contorno convexo φ e numa segunda fase aplicar um algoritmo para calcular os k CMC até obter a solução óptima, caso ainda não tenha sido obtida. O princípio subjacente à aplicação do método NISE ao CMCRA é semelhante ao descrito anteriormente para problemas biobjectivo. As diferenças situam-se em dois pontos: a redução do espaço de pesquisa e a condição de paragem do método.

A primeira fase do processo inicia-se com o cálculo das duas primeiras soluções, representadas na Figura 3 por A e B , que se obtêm minimizando respectivamente f_1 e f_2 (apenas a solução representada por B é admissível para o CMCRA). Com base na inclinação do segmento de recta \overline{AB} determina-se uma nova solução C_1 , podendo ocorrer duas situações distintas: a) se $f_2(C_1) \leq T$, ou seja, se a solução é admissível para o CMCRA, situação exemplificada na Figura 3, então a próxima solução é determinada com base na inclinação do segmento de recta $\overline{AC_1}$; b) se $f_2(C_1) > T$, ou seja, se a solução não é admissível para o CMCRA, então a próxima solução é determinada com base na inclinação do segmento de recta $\overline{BC_1}$. De facto, na situação a) não é possível, com base na inclinação do segmento de recta $\overline{BC_1}$, encontrar no interior do triângulo $[BO_2C_1]$ uma outra solução C_2 , para a qual $f_1(C_2) < f_1(C_1)$ e, por isso, a pesquisa é feita apenas no interior do triângulo $[AO_1C_1]$. Relativamente à situação b) com base na inclinação $\overline{AC_1}$ não é possível determinar nenhuma solução admissível para o CMCRA e, por isso, a pesquisa é feita apenas com base na inclinação $\overline{BC_1}$.

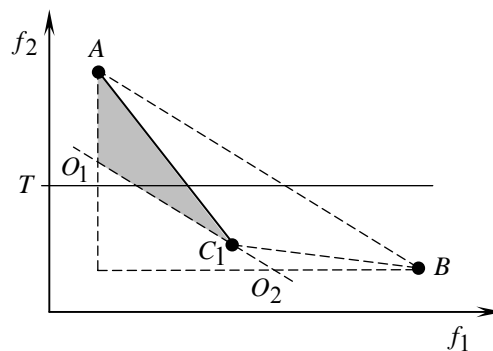


Figura 3 – Fase 1 do algoritmo; situação após o cálculo da terceira solução (C), usando o método NISE.

O processo é repetido até se obterem duas soluções C_i e C_{i+1} , tais que $f_2(C_i) \leq T$ e $f_2(C_{i+1}) > T$, conforme se exemplifica na Figura 4, e para as quais não é possível obter, com base na inclinação de $\overline{C_i C_{i+1}}$, outra solução diferente destas duas, ou seja, abaixo da recta que passa pelos pontos C_i e C_{i+1} não existe nenhuma outra solução.

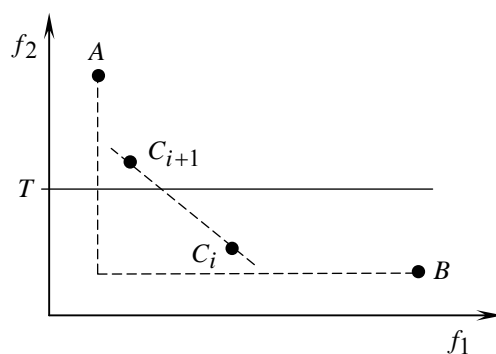


Figura 4 – Fase 1 do algoritmo; situação após o cálculo das duas últimas soluções, usando o método NISE.

Para pesquisar no interior do conjunto limitado inferiormente pelo contorno convexo φ não é possível continuar a utilizar o NISE. Inicia-se assim a segunda fase do processo, na qual a pesquisa é feita utilizando um algoritmo que determine os k CMC, com base na inclinação do segmento de recta $\overline{C_i C_{i+1}}$ (esta inclinação permanece fixa durante esta fase do processo). Nas duas primeiras iterações, as soluções determinadas são precisamente C_i e C_{i+1} , que designaremos por X_1 e X_2 , conforme se descreve na Figura 5 (note-se que X_1 é nesta fase a candidata a solução óptima). É fácil verificar que apenas dentro do triângulo representado a sombreado é possível existir uma outra solução X_i , para a qual $f_1(X_i) < f_1(X_1)$ e $f_2(X_i) \leq T$, ou seja, uma nova candidata a solução óptima do CMCRA. Note-se que o ponto X'_1 se situa na intersecção da recta horizontal $f_2 = T$ com a recta vertical que passa por X_1 .

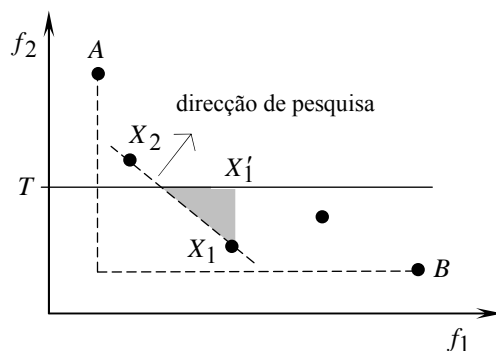


Figura 5 – Fase 2 do algoritmo; situação após o cálculo das duas primeiras soluções, com base nos k CMC.

A terceira solução a ser determinada nesta segunda fase é a representada por X_3 na Figura 6. Para definir a condição de paragem do processo considere-se uma recta r paralela ao segmento de recta

$\overline{C_i C_{i+1}}$ e que passa no ponto X_3 . Como esta recta passa acima do ponto X'_1 (note-se que X_1 é a actual candidata a solução óptima do problema), então não é possível encontrar outra solução X_i , para a qual $f_1(X_i) < f_1(X_1)$, por isso, X_1 é a solução óptima para o CMCRA. Ou seja, o processo termina quando se determina uma solução X_i para a qual a recta r (que passa por X_i) passe acima de X'_j (supondo que X_j representa a actual candidata a solução óptima para o problema).

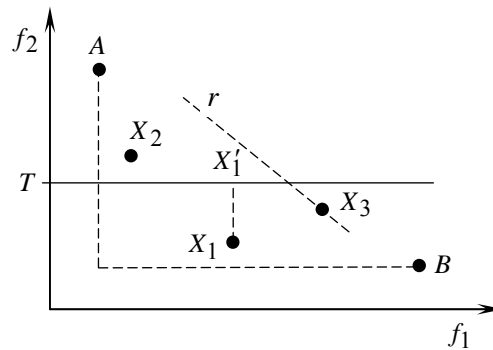


Figura 6 – Fase 2 do algoritmo; descrição da condição de paragem do algoritmo.

Considere-se a notação $CMC(a,b)$ e $kCMC(a,b)$ para designar respectivamente a determinação do CMC e dos $kCMC$ entre um nó inicial e um nó final de uma rede, para a qual os custos associados aos arcos são $ac_{ij} + bt_{ij}$. Assim, o algoritmo que descreve esta abordagem é o seguinte:

Fase1

$$C^+ \leftarrow CMC(1,0)$$

$$C^- \leftarrow CMC(0,1)$$

$$C \leftarrow CMC\left(1, \frac{f_1(C^-) - f_1(C^+)}{f_2(C^+) - f_2(C^-)}\right)$$

Enquanto $C \neq C^-$ e $C \neq C^+$

Se $f_2(C) \leq T$ **então**

$$C^- \leftarrow C$$

Caso contrário

$$C^+ \leftarrow C$$

Fim Se

$$C \leftarrow CMC\left(1, \frac{f_1(C^-) - f_1(C^+)}{f_2(C^+) - f_2(C^-)}\right)$$

Fim Enquanto

Fase 2

$CandOpt \leftarrow C^-$

$$\mu \leftarrow \frac{f_1(C^-) - f_1(C^+)}{f_2(C^+) - f_2(C^-)}$$

$k \leftarrow 3$

$X \leftarrow kCMC(1, \mu)$

Enquanto $(\mu(f_2(X) - T) + f_1(X)) < f_1(CandOpt)$

Se $f_2(X) \leq T$ e $f_1(X) < f_1(CandOpt)$ **então**

$CandOpt \leftarrow X$

Fim Se

$k \leftarrow k+1$

$X \leftarrow kCMC(1, \mu)$

Fim Enquanto

$SOpt \leftarrow CandOpt$

5. Implementação dos CMC e dos $kCMC$

O problema do CMC constitui um dos mais importantes e mais estudados em problemas de otimização. Desde finais da década de 50, vários artigos abordando este tema têm sido publicados, entre os quais se referem os mais recentes apresentados por Cherkassky *et al.* [5] e Zhan e Noon [22]. O primeiro artigo constitui um dos trabalhos mais abrangentes neste domínio, tendo os autores implementado e testado 17 algoritmos em várias redes experimentais de diferentes complexidades. Para além dos detalhes de implementação, são também referidos os diferentes graus de complexidade computacional. No segundo artigo os autores apresentam os desempenhos obtidos em redes reais por 15 dos 17 algoritmos abordados por Cherkassky *et al.* [5].

Quando o problema envolve custos associados aos arcos que podem tomar valores negativos, então o melhor algoritmo é o de Bellman-Ford-Moore [4,13,19], cuja complexidade computacional é $O(nm)$ (onde n representa o número total de nós e m o número total de arcos). No caso dos custos associados serem não negativos, o algoritmo que, de modo geral, obtém melhor desempenho é o de Dijkstra [11]. É possível encontrar na literatura diferentes estruturas utilizadas para a sua implementação e que podem ter complexidades computacionais bastante diferente, conforme se descreve na Tabela 1.

O algoritmo utilizado por nós para determinar o caminho mais curto entre dois nós ou entre um nó e todos os outros é semelhante ao de Dial [10] usando $C+1$ *buckets*. No entanto como os custos associados aos arcos podem ser reais foi necessário introduzir algumas modificações. Assim, para definir o número total de *buckets* fez-se $C = \max\{\text{trunc}(c_{ij}) : (i, j) \in A\}$. Em cada iteração, a identificação

dos *buckets* para colocar temporariamente os nós é feita com base na parte inteira dos seus rótulos temporários. Dentro de cada *bucket* os nós são ordenados por ordem crescente dos seus rótulos.

Tabela 1 – Complexidade das várias implementações do algoritmo de Dijkstra.

Implementação de Dijkstra	Complexidade
Implementação <i>naive</i> [11]	$O(n^2)$
Estrutura baseada em <i>buckets</i>	
Implementação básica [10]	$O(m + nC)$
<i>Overflow bag</i> [5]	$O(m + n(C/\alpha + \alpha))$
<i>Approximate bucket</i> [5]	$O(m\beta + n(C/\beta + \beta))$
<i>Double buckets</i> [9]	$O(m + n(C/\beta + \beta))$
Estrutura baseada em <i>heaps</i>	
<i>Fibonacci heap</i> [14]	$O(m + n \log n)$
<i>k-array heap</i> [8]	$O(m \log n)$
<i>R-heap</i> [1]	$O(m + n \log C)$

n – número total de nós;

m – número total de arcos;

C – comprimento máximo dos arcos ($C = \max\{c_{ij} : (i, j) \in A\}$);

α e β – parâmetros livres (Cherkassky *et al.* [5] consideram $\alpha = \min(50000, C/3)$ e

$$\beta = C/2^{11}).$$

O problema dos k CMC, no qual se determinam o primeiro, o segundo, ..., e o k -ésimo caminho mais curto entre dois pares de nós, tem várias aplicações em problemas de optimização. Vários algoritmos têm sido propostos para tratar este problema. Num estudo apresentado em Azevedo *et al.* [2], os autores comparam os algoritmos de Dreyfus [12], Martins [17] e um novo algoritmo por eles sugerido. Segundo o estudo, este novo algoritmo é bastante mais rápido que os restantes, embora a complexidade do algoritmo de Dreyfus [12] seja $O(kn \log n)$ e o do algoritmo de Azevedo *et al.* [2] seja $O(k^2m)$. O melhoramento introduzido neste algoritmo por Azevedo *et al.* [3] permite reduzir substancialmente a dimensão final da rede, uma vez que esta vai sendo aumentada em cada iteração, ou seja, sempre que se calcula o i -ésimo caminho mais curto, o número total de nós e arcos é incrementado.

O algoritmo por nós utilizado para calcular os k CMC foi o proposto por Azevedo *et al.* [2], com o melhoramento sugerido por Azevedo *et al.* [3].

6. Resultados computacionais

Para comparar os dois algoritmos referidos nas secções 3 e 4.2, que passaremos a designar por Alg1 e Alg2 respectivamente, geraram-se aleatoriamente 15 redes teste, com o número de nós a variar entre 10000 e 40000 e as densidades a variar entre 1.5 e 10. Para cada uma das redes testaram-se 4 valores

diferentes para o parâmetro T , contabilizando deste modo um total de 60 problemas resolvidos. Os resultados foram obtidos com um *Pentium III* de 1 Ghz e 512 RAM.

Tal como já foi referido, ambos os algoritmos são compostos por duas fases: numa primeira fase, calculam-se alguns CMC entre o nó inicial e o nó final, com diferentes direcções de pesquisa, até satisfazer uma determinada condição de paragem; e, numa segunda fase, calculam-se os k CMC entre o nó inicial e o nó final, sempre com a mesma direcção de pesquisa, até satisfazer outra condição de paragem. Para cada um dos 60 problemas de teste resolvidos, o número de CMC e de iterações dos k CMC foi exactamente o mesmo para os dois algoritmos. Relativamente ao tempos computacionais, praticamente os mesmos, variam entre 0.8 segundos, para problemas com 10000 nós e 100000 arcos, e 17.1 segundos, para problemas com 40000 nós e 400000 arcos.

Para melhor se perceber e comparar o desempenho dos dois algoritmos, apresenta-se de seguida na Tabela 2 e na Figura 7, as sucessivas iterações até se obter a solução óptima para o problema teórico descrito por Handler e Zang [15]. Relativamente à Tabela 2, os parâmetros a e b indicam que os custos associados aos arcos são calculados através de $ac_{ij} + bt_{ij}$ (é a mesma notação que foi usada na secção 4.2). Para obter mais informações sobre as condições de paragem do Alg1 deve consultar-se Handler e Zang [15] e sobre o Alg2 deve consultar-se a secção 4.2 deste relatório de investigação.

Tabela 2 – Comparação entre Alg1 e Alg2 para a rede de teste apresentada por Handler e Zang [15].

			Condições de paragem	
			Alg1	Alg2
	$CMC(a,b)$	Soluções obtidas	$L(\mu) \geq L$	$C = C^-$ ou $C = C^+$
Fase 1	$CMC(1,0)$	$A = (5,2)$	–	–
	$CMC(0,1)$	$B = (20,0.35)$	–	–
	$CMC(1,9.09)$	$C_1 = (9,1.2)$	$10.8 \geq 14.1$	$C_1 = A$ ou $C_1 = B$
	$CMC(1,12.94)$	$C_2 = (15,0.6)$	$9.8 \geq 11.6$	$C_2 = B$ ou $C_2 = C_1$
	$CMC(1,10)$	$C_3 = C_1 = (9,1.2)$	$11 \geq 11$	$C_3 = C_1$ ou $C_3 = C_2$
	$kCMC(a,b)$	Soluções obtidas	$LB \geq UB$	$(\mu(f_2(X) - T) + f_1(X)) < f_1(CandOpt)$
Fase 2	$kCMC(1,10)$	$X_1 = C_1 = (9,1.2)$	–	–
	”	$X_2 = C_2 = (15,0.6)$	–	–
	”	$X_3 = (6,1.6)$	$12 \geq 15$	$12 \geq 15$
	”	$X_4 = (14,0.9)$	$13 \geq 14$	$13 \geq 14$
	”	$X_5 = B = (20,0.35)$	$13.5 \geq 14$	$13.5 \geq 14$
	”	$X_6 = A = (5,2)$	$15 \geq 14$	$15 \geq 14$

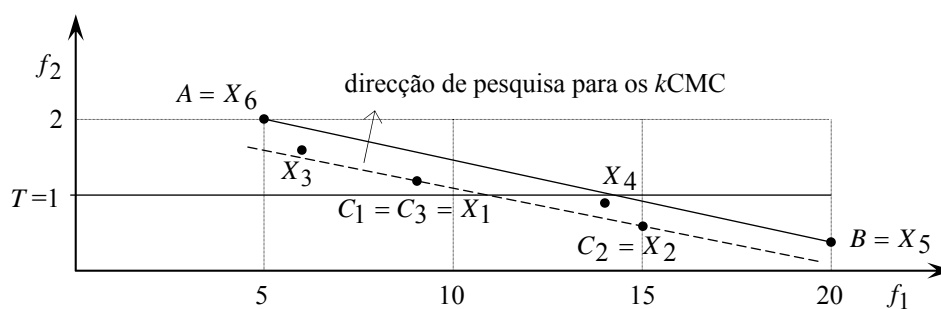


Figura 7 – Representação das soluções obtidas por Alg1 e Alg2 para a rede de teste apresentada por Handler e Zang [15].

Da análise dos resultados apresentados na Tabela 2 e na Figura 7, verifica-se que, embora as metodologias subjacentes aos dois algoritmos sejam completamente distintas (Alg1 baseia-se na relaxação Lagrangeana e Alg2 no método NISE), as condições de paragem, quer na fase 1 quer na fase 2, produzem exactamente os mesmos resultados, em termos das soluções calculadas até se obter a óptima.

7. Conclusões

O algoritmo de Handler e Zang [15], baseado na relaxação Lagrangeana, é um dos mais referidos na literatura para determinar a solução óptima de um CMCRA. O processo consiste basicamente em relaxar a restrição adicional e resolver o problema dual associado. No caso de existir um intervalo de dualidade é necessário utilizar um algoritmo que calcule os k CMC para pesquisar dentro desse intervalo.

O algoritmo aqui proposto baseia-se numa metodologia completamente diferente, conhecida por *NonInferior Set Estimation* [6,7]. Nesta metodologia determinam-se algumas soluções situadas em vértices do contorno convexo φ (definido na secção 1), com base nas propriedades dos segmentos de recta que unem os pontos que representam as soluções já calculadas. No caso de não se obter nesta primeira fase a solução óptima, utiliza-se um algoritmo que calcule os k CMC para pesquisar no interior do conjunto limitado inferiormente pelo contorno convexo φ .

Os resultados obtidos por estas duas abordagens distintas são exactamente os mesmos, em termos do número de CMC resolvidos na primeira fase e do número de iterações dos k CMC na segunda fase. Consequentemente, os tempos computacionais dos dois algoritmos são praticamente os mesmos, uma vez que as outras operações envolvidas não têm grande significado em termos de complexidade computacional.

A vantagem da nova abordagem proposta, está no facto dos conceitos envolvidos serem extremamente simples de assimilar e implementar. De facto, nesta abordagem, o modo como se calculam as soluções

em cada iteração é facilmente descrito através de um gráfico, pois baseia-se unicamente em considerações geométricas das soluções calculadas nas iterações anteriores. Já o algoritmo de Handler e Zang [15] utiliza outros conceitos não tão simplistas relacionados com relaxação Lagrangeana e dualidade linear.

8. Referências

- [1] Ahuja, R.K., Mehlhorn, K., Orlin, J.B. e Tarjan, R.E., “Faster Algorithms for the Shortest Path Problem”, *Journal of ACM* 37, 213-223 (1990).
- [2] Azevedo, J.A., Costa, M., Madeira, J. e Martins, E.Q.V., “An Algorithm for the Ranking of Shortest Paths”, *European Journal of Operational Research* 69, 97-106 (1993).
- [3] Azevedo, J.A., Madeira, J., Costa, M., Martins, E.Q.V. e Pires, F., “A Computational Improvement for a Shortest Paths Ranking Algorithm”, *European Journal of Operational Research* 73, 188-191 (1994).
- [4] Bellman, R.E., “On a Routing Problem”, *Quart. Appl. Math.* 16, 87-90 (1958).
- [5] Cherkassky, B.V., Goldberg, A.V. e Radzik, T., “Shortest Paths Algorithms: Theory and Experimental Evaluation”, *Mathematical Programming* 73, 129-174 (1996).
- [6] Cohon, J., *Multiobjective Programming and Planning*, Academic Press, 1978.
- [7] Cohon, J. Church, R. e Sheer, D., “Generating Multiobjective Trade-offs: An Algorithm for Bicriterion Problems”, *Water Resources Research* 15, 1001-1010 (1979).
- [8] Corman, T.H., Leiserson, C.E. e Riverst, R.L., *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.
- [9] Denardo, E.V. e Fox, B.L., “Shortest-Route Methods: Reaching, Pruning, and Buckets”, *Operations Research* 27, 161-186 (1979).
- [10] Dial, R.B., “Algorithm 360: Shortest Path Forest With Topological Ordering”, *Communications of the ACM* 12, 632-633 (1969).
- [11] Dijkstra, E.W., “A Note on Two Problems in Connection With Graphs”, *Numer. Math* 1, 269-271 (1959).
- [12] Dreyfus, S., “An Appraisal of Some Shortest Path Algorithms”, *Operations Research* 17, 345-412, (1969).
- [13] Ford Jr., L.R. e Fulkerson, D.R., *Flows in Networks*, Princeton Univ. Press, Princeton, NJ, 1962.
- [14] Fredman, M.L. e Tarjan, R.E., “Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms”, *Journal of ACM* 34, 596-615 (1987).
- [15] Handler, G.Y. e Zang, I., “A Dual Algorithm for the Constrained Shortest Path Problem”, *Network* 10, 293-310 (1980).
- [16] Johsch, H.C., “The Shortest Route Problem with Constraints”, *J. Math. Analysis Appl.* 14, 191-197 (1966).
- [17] Martins, E.Q.V., “An Algorithm for Ranking Paths that May Contain Cycles”, *European Journal of Operational Research* 18, 123-130 (1984).

- [18] Minoux, M., “Plus Courts Chemins avec Contraintes: Algorithmes et Applications”, *Ann. Télécommunications* 30, 383-394 (1975).
- [19] Moore, E.F., “The Shortest Path through a Maze”, *Proceedings of the Int. Symp. on the Theory of Switching*, Harvard University Press, 285-292 (1959).
- [20] Shier, D.A., “Iterative Methods for Determining k -Shortest Paths in a Network”, *Network* 6, 205-230 (1976).
- [21] Yen, J.Y., “Finding the k Shortest Loopless Paths in a Network”, *Management Science* 17, 712-716 (1971).
- [22] Zhan, F.B. e Noon C.E., “Shortest Path Algorithms on Real Road Networks”, *Transportation Science* 32, 65-73 (1998).