

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

Luis A. Santos
lsantos@issb.pt

João M. Coutinho-Rodrigues
coutinho@dec.uc.pt

Um Novo Algoritmo para o Cálculo de Caminhos Óptimos
Baseado numa Direcção de Pesquisa Definida
por uma Restrição Adicional

No. 10

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

Um Novo Algoritmo para o Cálculo de Caminhos Óptimos

Baseado numa Direcção de Pesquisa Definida por uma Restrição Adicional

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 an applied sub-problems in real networks. The objective of this problem is to identify the least cost path through a network from a pre-determined starting node to a pre-determined terminus node. This problem has been strongly studied and we can find in literature numerous algorithms that solve the problem in polynomial time. If we add another constraint to SP (for example, limiting the travel time), the new problem, known as Constrained Shortest Path problem (CSP), is at least NP-Complete [8]. In this paper, a new optimal algorithm for the CSP is introduced. This algorithm based on the use of the k Shortest Path problem (k SP) was tested on several experimental network and compared to the two most commonly used algorithms to solve CSP, introduced by Handler e Zang [8]. The results indicate that the new algorithm is superior to the previous ones in terms of solution time and/or computer memory requirements.

Resumo

O problema do Caminho Mais Curto (CMC) constitui um dos sub-problemas mais resolvidos em problemas de redes. O objectivo deste problema é calcular o caminho óptimo (por exemplo em termos de comprimento total a percorrer) entre dois nós pré-definidos de uma rede ou entre um nó pré-definido e todos os outros. A sua complexidade computacional é polinomial e existem diversos algoritmos e diferentes implementações desses algoritmos na literatura. Se introduzirmos uma nova restrição no problema do CMC (por exemplo, impondo um limite para o tempo de percurso), obtém-se o denominado problema do Caminho Mais Curto com uma Restrição adicional (CMCRA) que é, pelo menos, NP-Completo [8]. Neste relatório de investigação, é apresentado um novo algoritmo baseado na resolução do problema dos k Caminhos Mais Curtos (k CMC) para tratar o CMCRA. Este algoritmo foi

testado em várias redes experimentais e comparado com dois dos algoritmos mais usados para tratar o problema, referidos por Handler e Zang [8]. Os resultados obtidos indicam que este novo algoritmo é melhor que os outros dois em termos do tempo computacional requerido e da memória necessária para obter a solução ótima.

Palavras chave: Redes, Caminho ótimo, Restrição adicional, k caminhos ótimos

1. Introdução

O problema da determinação do CMC, cujo objectivo é calcular o caminho ótimo entre dois nós pré-definidos de uma rede ou entre um nó pré-definido e todos os outros, constitui um dos sub-problemas mais resolvidos em redes, nomeadamente em problemas de transportes e circulação. Este problema tem sido bastante estudado desde sensivelmente finais da década de 50 e é possível encontrar na literatura vários algoritmos de complexidade polinomial, bem como sucessivos melhoramentos de alguns deles [3,15], que permitem tratar o problema de forma eficiente. Um problema fortemente relacionado com o do CMC é o da determinação dos k CMC, que consiste em calcular os primeiros k caminhos ótimos entre dois nós pré-definidos de uma rede (para $k=1$ obtém-se o 1º CMC, para $k=2$ obtém-se o 2º CMC, ...). Para tratar este problema, igualmente de complexidade polinomial, vários algoritmos têm sido apresentados na literatura [1,7,11,13,14].

O problema da determinação do CMCRA, que se obtém quando se adiciona uma nova restrição ao problema do CMC (por exemplo, minimizar um comprimento total a percorrer impondo um limite máximo para o tempo de percurso), tem uma complexidade computacional substancialmente mais elevada em comparação com o problema do CMC. De facto, Handler e Zang [8] mostraram que o CMCRA é, pelo menos, NP-Completo. Tal como acontece para o CMC e os k CMC, as aplicações reais do CMCRA são várias e estão normalmente associados a problemas de circulação em redes [12].

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 restrição adicional [8]. No entanto, grande parte dos problemas (independentemente do seu tamanho ou dos valores numéricos associados à restrição adicional) requer, para obter a solução ótima, um número de iterações extremamente elevado.

Para colmatar este óbice, outras abordagens têm sido apresentadas, baseadas em programação dinâmica [9], na relaxação Lagrangeana da restrição adicional [8] ou no método NISE (*NonInferior Set Estimation*) [10]. O algoritmo proposto por Handler e Zang [8], baseado na relaxação Lagrangeana, tem constituído ao longo dos anos uma das principais referências para tratar o CMCRA, sendo um dos mais citados na literatura especializada. O algoritmo apresentado por Santos e Coutinho [10], baseado numa metodologia extremamente simples de assimilar e implementar (método NISE [4,5]) obtém exactamente os mesmos resultados do algoritmo de Handler e Zang [8] baseado na relaxação Lagrangeana, sendo

equivalentes quer em termos do número de CMC resolvidos, quer no número de iterações do algoritmo que calcula os k CMC.

Neste relatório de investigação, apresenta-se um novo algoritmo para tratar o CMCRA, que foi testado em várias redes experimentais e comparado com dois dos algoritmos mais usados para tratar o problema, propostos por Handler e Zang [8]. Este algoritmo é baseado unicamente na determinação dos k CMC e utiliza uma nova direcção de pesquisa, diferente da clássica, que tem em conta uma característica simples de cada problema em particular, independente da rede, relacionada com o valor relativo imposto para a restrição adicional.

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 respectivo tempo de atravessamento do arco (para nos referirmos genericamente a qualquer desses parâmetros associados 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 inteira (e binária) 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 (com T um parâmetro real) imposta ao problema do CMC de base contemplando as 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) pertence ao caminho a determinar pelo modelo como solução óptima, então $x_{ij} = 1$; caso contrário, $x_{ij} = 0$.

3. Métodos Existentes para Resolução do CMCRA

3.1 Abordagem Clássica Baseada nos k CMC

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 $X \in Y$:

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

e

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

Um dos processos que pode ser utilizado para obter a solução óptima do CMCRA consiste em aplicar um algoritmo que determine os k CMC até obter o primeiro caminho $X_k \in Y$ para o qual $f_2(X_k) \leq T$ [8]. Para descrever graficamente o processo, os caminhos $X \in Y$ serão representados num sistema de eixos cartesianos por um ponto com a mesma designação e de coordenadas $(f_1(X), f_2(X))$. Assim, esta abordagem consiste em percorrer o conjunto Y no sentido do vector $c = (1, 0)$, ou seja, da esquerda para a direita, até encontrar o primeiro caminho situado abaixo ou coincidente com a recta horizontal $f_2 = T$. No exemplo apresentado na Figura 1, a solução óptima do problema é obtida na 4ª iteração do algoritmo, com a determinação do 4º caminho mais curto (X_4).

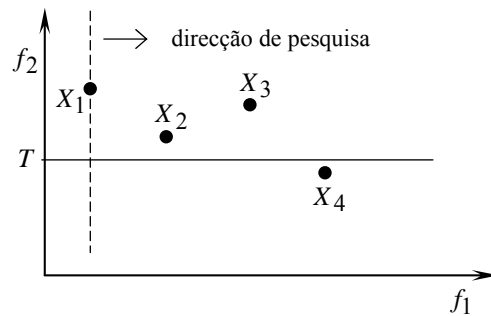


Figura 1 – Abordagem clássica baseada nos k CMC.

Embora esta abordagem seja extremamente simples, uma vez que envolve apenas a implementação de um algoritmo para determinar os k CMC, tem como principal inconveniente o facto de, em determinadas situações, o número de iterações do algoritmo ser extremamente elevado, até se encontrar a solução óptima, conduzindo a tempos de cálculo e memória requerida incomportáveis. De facto, retomando ainda o exemplo da Figura 1, a região delimitada pelas rectas verticais que passam pelos pontos X_1 e X_4 não é limitada superiormente, o que significa que, embora neste caso hipotético tal não aconteça, poderia existir nesta região um número muito elevado de caminhos a calcular.

3.2 Algoritmo de Handler e Zang

O algoritmo proposto por Handler e Zang [8], baseado na relaxação Lagrangeana da restrição adicional, é um dos mais referidos na literatura para tratar o CMCRA. Este algoritmo divide-se em duas fases: numa primeira fase, resolve-se o problema dual associado ao problema que se obtém após relaxar a restrição adicional, o que permite alcançar um limite inferior de muito boa qualidade para o CMCRA e, numa segunda fase, inspecciona-se o intervalo de dualidade, no caso de existir (este intervalo pode existir porque o dual é um programa convexo mas o primal não), através de um algoritmo que determine os k CMC.

A primeira fase, inicia-se com a determinação dos caminhos A e B , que se obtém minimizando isoladamente f_1 e f_2 respectivamente. O seguinte (C) é calculado através da determinação do CMC

para os novos custos associados aos arcos e definidos por $c_{ij} + \mu t_{ij}$, com $\mu = \frac{f_1(B) - f_1(A)}{f_2(A) - f_2(B)}$. Se

$f_2(C) > T$, então o parâmetro μ é actualizado substituindo o caminho A pelo caminho C , se $f_2(C) \leq T$,

então o parâmetro μ é actualizado substituindo o caminho B pelo caminho C . Esta fase termina quando para dois caminhos C_i e C_{i+1} não for possível obter qualquer outro diferente destes dois, através da

determinação do CMC usando como custos os valores dados por $c_{ij} + \mu t_{ij}$. Conforme se exemplifica

na Figura 2, em termos geométricos, esta fase termina quando abaixo da recta que une os pontos C_i e C_{i+1} não existir qualquer outra solução para o CMC sem a restrição adicional (note-se que o segmento de recta que une os pontos C_i e C_{i+1} constitui uma aresta do 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 $c_{ij} + \mu t_{ij}$, para qualquer valor não negativo do parâmetro real μ , considerando o sistema de eixos cartesiano bidimensional (f_1, f_2) .

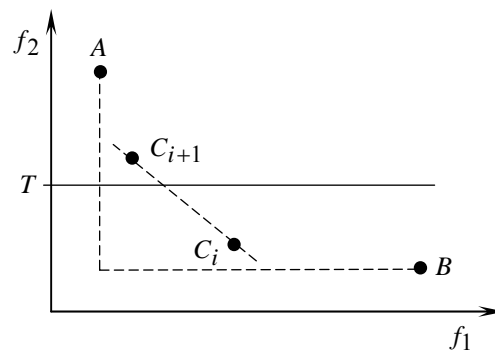


Figura 2 – Condição de paragem da primeira fase do algoritmo de Handler e Zang [8], baseado na relaxação Lagrangeana.

Na segunda fase, são determinados alguns k CMC, utilizando uma direcção de pesquisa fixa e perpendicular ao segmento de recta que une os pontos C_i e C_{i+1} , até verificar a condição de paragem do processo. Esta condição de paragem, baseia-se nos limites inferior e superior associados aos problemas dual e primal, tornando mais difícil de exemplificar geometricamente (para mais detalhes consultar [8]).

4. Uma Nova Abordagem Baseada nos k CMC

4.1 Definição da Direcção de Pesquisa

Depois de avaliar, em algumas redes de teste, o desempenho das duas abordagens descritas anteriormente concluiu-se que:

- a) a abordagem clássica baseada nos k CMC, tem a vantagem de necessitar de calcular apenas um CMC do nó inicial para todos os outros (um CMC é muitíssimo mais dispendioso, em termos computacionais, do que um k CMC para $k > 1$ no algoritmo que usamos), mas tem a desvantagem de, para determinados problemas, necessitar de um número incomportavelmente elevado de iterações (k) para obter a solução óptima;

- b) o algoritmo de Handler e Zang [8], baseado na relaxação Lagrangeana, tem a vantagem de o número de iterações dos k CMC ser extremamente reduzido, mas necessita de um CMC do nó inicial para todos os outros e pelo menos dois CMC do nó inicial para o nó final.

A ideia base associada a esta nova abordagem foi tentar encontrar um procedimento que permitisse tratar o CMCRA, com o menor número possível de CMC e iterações dos k CMC. Utilizando uma abordagem baseada unicamente no cálculo dos k CMC, o que também acontece na abordagem clássica, propõe-se uma nova direcção de pesquisa no conjunto Y que reduza o número de iterações do algoritmo. Quer na abordagem clássica quer nesta nova abordagem a direcção de pesquisa é sempre a mesma durante todo o processo.

O objectivo principal foi encontrar uma direcção de pesquisa que tenha em conta alguma característica de cada problema (identificável a partir dos respectivos parâmetros) e que possibilite a determinação do menor número possível de pontos, representados no sistema de eixos cartesiano, até validar uma determinada condição de paragem. Neste caso, o vector c , que representa a direcção de pesquisa, é definido por uma função expressa com base nas coordenadas de dois pontos A e B , que se obtêm minimizando respectivamente f_1 e f_2 , e do valor do parâmetro T , do seguinte modo:

$$c = \left(1, \frac{f_1(B) - f_1(A)}{7p^2 [f_2(A) - f_2(B)]} \right), \quad (7)$$

onde $p = \frac{T - f_2(B)}{f_2(A) - f_2(B)}$.

Esta nova direcção de pesquisa foi obtida de modo que seja aproximadamente tangente ao contorno convexo φ , a recta perpendicular a c que passa no ponto de intersecção da recta $f_2 = T$ com o contorno convexo φ . Esta situação será exemplificada de seguida na Figura 3, que descreve graficamente como varia a inclinação da direcção de pesquisa para 4 valores distintos do parâmetro p .

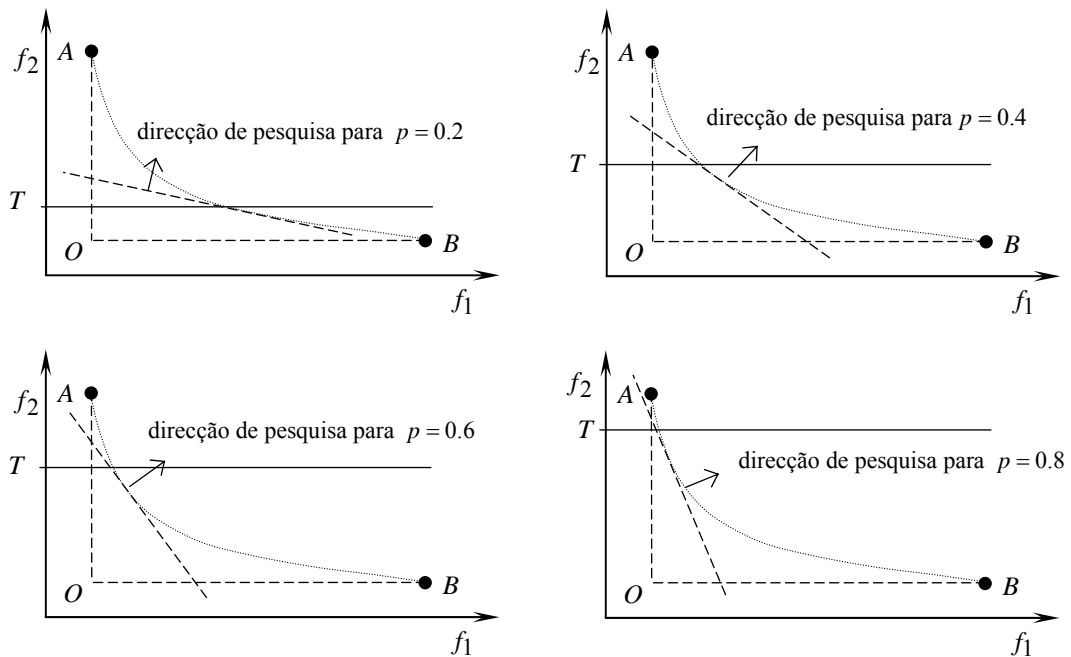


Figura 3 – Variação da direcção de pesquisa para 4 valores diferentes do parâmetro p .

Como não é conhecida a configuração do contorno convexo φ junto ao ponto de intersecção da recta $f_2 = T$ com o próprio contorno convexo φ , apenas se consegue obter uma aproximação da recta tangente. De facto, para determinar essa configuração seria necessário utilizar um procedimento semelhante ao da primeira fase do algoritmo de Handler e Zang [8], situação que esta nova abordagem pretende evitar, pois nesse caso deixaria de ter vantagens relativamente ao algoritmo de Handler e Zang [8].

4.2 Descrição do Algoritmo

Para descrever graficamente o modo como o espaço das soluções é percorrido, bem como a condição de paragem do processo, é apresentado um pequeno exemplo nas Figuras 4, 5 e 6, considerando o valor do parâmetro p igual a 0.4. Da observação da Figura 4, e tendo em conta a forma como se obtêm os pontos A e B , é fácil concluir que à esquerda do segmento de recta \overline{OA} e abaixo do segmento de recta \overline{OB} não existem soluções admissíveis para o problema. Utilizando a direcção de pesquisa calculada pela equação 7, determina-se o primeiro caminho, que será designado por X_1 . Esta solução é admissível para o CMCRA e, por isso, constitui a primeira candidata a solução óptima do problema. É fácil verificar que apenas dentro do triângulo representado a sombreado é possível existir outra solução X_i , para a qual $f_1(X_i) < f_1(X_1)$, ou seja, uma nova candidata a solução óptima para o CMCRA.

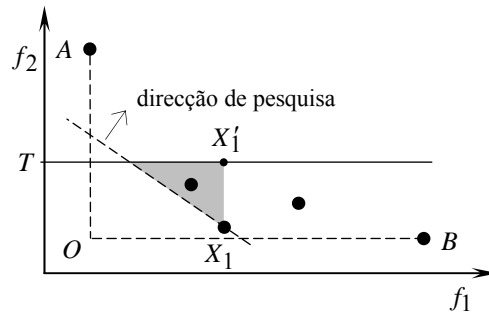


Figura 4 – Nova abordagem baseada nos k CMC – 1ª iteração.

O segundo caminho a ser determinado é representado por X_2 , conforme se mostra na Figura 5. Como também se trata de uma solução admissível para o problema e $f_1(X_2) < f_1(X_1)$ passa a ser esta a candidata a solução óptima. O triângulo a sombreado, é agora mais reduzido relativamente ao da iteração anterior. Para definir a condição de paragem, considere-se a recta r perpendicular ao vector c (direcção da pesquisa), que passa por X_2 . Se o ponto X_2 estivesse fora do triângulo a sombreado, de tal forma que a recta r passasse acima do ponto X_1' (nestas condições X_1 seria a última candidata a solução óptima), o processo terminava (significava que a área do triângulo a sombreado era nula). Como tal não acontece o processo continua.

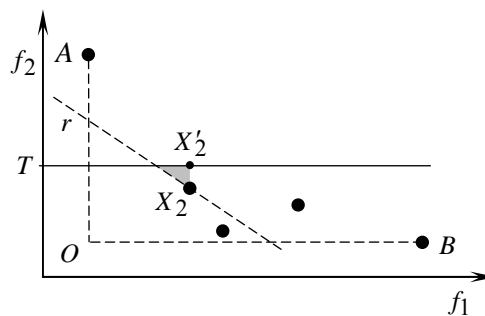


Figura 5 – Nova abordagem baseada nos k CMC – 2ª iteração.

Na Figura 6 é possível observar que com a determinação do terceiro caminho representado por X_3 , o processo termina, pois a recta r (que passa por X_3 e é perpendicular a c) passa acima de X_2' (X_2 é a última solução candidata a óptima). É fácil constatar que não é possível encontrar qualquer solução admissível X_i , para a qual $f_1(X_i) < f_1(X_2)$, por isso, X_2 é a solução óptima para o CMCRA.

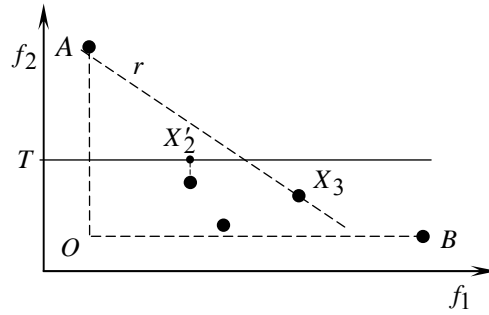


Figura 6 – Nova abordagem baseada nos k CMC – 3ª iteração.

Considere-se a notação $CMC(a,b)$ e $kCMC(a,b)$ para designar respectivamente a determinação do caminho mais curto e dos k caminhos mais curtos entre um nó inicial e um nó terminal numa rede, para os quais os custos associados aos arcos são definidos por $ac_{ij} + bt_{ij}$. Assim, o algoritmo que descreve esta abordagem é o seguinte:

```

A ← CMC(1,0)
B ← CMC(0,1)
CandOpt ← B   {candidata a solução óptima}
μ ←  $\frac{f_1(B) - f_1(A)}{7p^2[f_2(A) - f_2(B)]}$ 
k ← 1
X ← kCMC(1, μ)
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 ← X
    Fim Se
    k ← k+1
    X ← kCMC(1, μ)
Fim Enquanto
SOpt ← CandOpt

```

5. Resultados computacionais

De seguida apresentar-se-ão os resultados computacionais para os 3 algoritmos descritos nas secções anteriores, obtidos com um Pentium III a 1Ghz com 512 RAM. Os testes foram efectuados em redes teóricas geradas aleatoriamente de forma que contenham um ciclo Hamiltoniano também gerado

aleatoriamente. Os valores inteiros dos parâmetros c_{ij} e t_{ij} , atribuídos aos arcos, foram seleccionados aleatoriamente do intervalo $\{1, \dots, 500\}$. O número de nós de cada rede varia entre 10 000 e 40 000 e as relações ($\#$ arcos / $\#$ nós) variam entre 1,5 e 10. Para um número específico de nós e arcos foram geradas 10 redes diferentes e para cada uma destas redes foram testados 4 valores para o parâmetro p (ver secção 4.1). O número de arcos que partem e que chegam a cada nó é calculado de forma aleatória e varia entre 1 e $\#$ nós-1. Nas Tabelas 1, 2, 3 e 4 são apresentados os tempos de cálculo e o número de iterações obtidos por cada um dos algoritmos e para cada um dos 4 valores do parâmetro p . Nestas tabelas são utilizadas as seguintes designações:

Alg1 – abordagem clássica baseada nos k CMC (ver secção 3.1);

Alg2 – algoritmo de Handler e Zang, baseado na relaxação Lagrangeana (ver secção 3.2);

Alg3 – nova abordagem baseada nos k CMC (ver secção 4.2);

$\#$ NOpt Alg1 – número de problemas para os quais o Alg1 não obteve a solução óptima;

$\#$ CMC (Med) – número médio de CMC do nó inicial para o nó final;

$\#$ CMC (Max) – número máximo de CMC do nó inicial para o nó final;

$\#k$ CMC (Med) – número médio de iterações para os k CMC;

$\#k$ CMC (Max) – número máximo de iterações para os k CMC;

O algoritmo implementado para determinar os CMC entre o nó inicial e o nó final e entre o nó inicial e todos os outros foi o de Dial [6] como uma pequena adaptação (para mais detalhes ver [10]). De facto, o algoritmo original apenas pode ser aplicado quando os valores numéricos associados aos arcos são inteiros e, nestes problemas, os valores podem ser reais (a função a minimizar é $ac_{ij} + bt_{ij}$ podendo os valores de a e b serem reais). Relativamente ao algoritmo para determinar os k CMC, implementou-se o proposto por Azevedo *et. al.* [1], com as melhorias posteriormente efectuadas [2], pois segundo os autores é o que obtém melhores resultados em termos de velocidade de cálculo. Neste algoritmo, a dimensão da rede original é aumentada em cada iteração (são copiados e acrescentados à rede original alguns nós e alguns arcos).

Relativamente ao Alg1, para alguns problemas o número de iterações dos k CMC é extremamente elevado e, por isso, a memória necessária para obter a solução óptima é igualmente elevada (por exemplo, numa experiência efectuada para uma determinada rede de teste, 512 MB de memória RAM mais aproximadamente 3 MB de memória virtual não foram suficientes para obter a solução óptima). Optou-se por utilizar apenas a memória RAM disponível, indicando o número de problemas para os quais o algoritmo não obteve a solução óptima. Deste modo, os resultados globais referentes a este algoritmo, apresentados na última linha de cada tabela, devem ser vistos como um limite mínimo para os resultados que se obteriam se a solução óptima fosse sempre alcançada (alguns casos ficarão muito distante da solução óptima).

Em todos os algoritmos é calculado o CMC do nó inicial para todos os outros, que corresponde à primeira iteração do algoritmo dos k CMC. No Alg3 são sempre calculados, para cada problema, exactamente dois CMC do nó inicial para o nó final, para determinar as duas soluções iniciais a partir das quais se define a direcção de pesquisa. Este valor, embora contabilizado nos tempos computacionais, não é apresentado na secção referente ao número de CMC por ser sempre constante e igual a 2.

Tabela 1 – Resultados computacionais para $p = 0.2$.

#Nós	#Arcos	#NOpt Alg1	Tempos computacionais (seg)			Nº de CMC e nº de iterações do k CMC							
			Alg1	Alg2	Alg3	Alg1		Alg2				Alg3	
						#kCMC (Med)	#kCMC (Max)	#CMC (Med)	#CMC (Max)	#kCMC (Med)	#kCMC (Max)	#kCMC (Med)	#kCMC (Max)
10000	15000	0	0.2	0.8	0.5	15.8	73	3.7	5	2.5	6	1.5	2
	25000	0	2.6	1.2	0.7	9 611.1	88 275	4.2	6	4.1	7	2.7	7
	50000	2	5.4	2.0	1.1	28 411.7	105 537	4.9	6	4.2	11	6.7	23
	75000	8	15.3	2.6	1.5	64 056.1	90 561	5.0	6	5.3	8	13.3	32
	100000	7	10.7	3.2	1.8	36 465.6	57 559	4.9	6	6.5	17	44.8	140
20000	30000	0	0.4	1.7	1.1	46.7	209	3.5	5	3.0	4	1.4	2
	50000	0	0.5	2.5	1.4	577.9	1 912	4.5	6	3.2	5	2.5	6
	100000	3	6.2	4.4	2.3	30 413.9	100 030	5.2	6	4.2	14	5.2	13
	150000	6	13.0	6.0	3.1	46 928.4	65 628	5.3	6	5.6	11	11.3	25
	200000	9	15.6	7.4	3.8	45 293.5	59 905	5.4	7	5.2	10	16.7	60
40000	60000	0	0.8	3.6	2.4	12.2	42	3.7	5	2.3	4	1.5	2
	100000	1	3.3	5.3	3.0	9 432.0	88 653	4.5	6	3.2	6	2.6	5
	200000	4	10.5	9.6	4.8	45 490.5	98 009	5.2	6	5.4	16	7.7	16
	300000	9	16.2	13.4	6.6	53 696.7	66 079	5.5	7	6.5	20	10.4	26
	400000	8	14.5	17.1	7.9	32 876.8	46 504	5.6	6	6.4	19	25.9	85
Média global			7.7	5.4	2.8	26 888.6	57 931.7	4.7	5.9	4.5	10.5	10.3	29.6

Tabela 2 – Resultados computacionais para $p = 0.4$.

#Nós	#Arcos	#NOpt Alg1	Tempos computacionais (seg)			Nº de CMC e nº de iterações do k CMC							
			Alg1	Alg2	Alg3	Alg1		Alg2				Alg3	
						#kCMC (Med)	#kCMC (Max)	#CMC (Med)	#CMC (Max)	#kCMC (Med)	#kCMC (Max)	#kCMC (Med)	#kCMC (Max)
10000	15000	0	0.2	0.8	0.5	10.1	73	3.7	5	3.1	11	2.9	13
	25000	0	0.3	1.1	0.7	713.3	3 816	4.0	5	4.6	15	6.2	22
	50000	0	0.4	2.0	1.1	329.8	2 215	4.8	6	2.9	5	5.8	30
	75000	1	3.1	2.9	1.4	11 937.3	76 272	5.5	7	3.6	9	6.5	14
	100000	3	5.1	3.5	1.8	15 770.3	57 559	5.2	6	5.1	13	15.8	65
20000	30000	0	0.4	1.7	1.1	23.6	99	3.5	5	3.4	5	3.5	6
	50000	0	0.5	2.5	1.4	505.6	1 912	4.5	6	5.1	9	4.7	9
	100000	1	4.1	4.4	2.3	17 863.2	91 394	5.1	6	6.4	13	6.2	14
	150000	3	5.9	6.2	3.0	18 739.6	65 628	5.4	6	2.8	4	5.7	14
	200000	3	6.9	7.0	3.7	18 838.9	57 348	4.9	6	7.2	27	11.8	29
40000	60000	0	0.8	3.7	2.4	8.1	42	3.8	5	3.0	9	2.9	9
	100000	0	1.0	5.3	3.0	55.1	300	4.4	6	2.9	5	2.5	5
	200000	1	4.8	9.9	4.8	17 000.4	90 770	5.2	7	4.1	8	5.5	20
	300000	3	7.3	13.4	6.6	20 090.9	63 024	5.2	6	5.5	14	8.5	21
	400000	3	9.2	17.1	8.0	19 017.0	46 504	5.4	7	5.1	15	16.2	91
Média global			3.3	5.4	2.8	9 393.5	37 130.4	4.7	5.9	4.3	10.8	7.0	24.1

Tabela 3 – Resultados computacionais para $p = 0.6$.

#Nós	#Arcos	#NOpt Alg1	Tempos computacionais (seg)			Nº de CMC e nº de iterações do k CMC							
			Alg1	Alg2	Alg3	Alg1		Alg2				Alg3	
						# k CMC (Med)	# k CMC (Max)	#CMC (Med)	#CMC (Max)	# k CMC (Med)	# k CMC (Max)	# k CMC (Med)	# k CMC (Max)
10000	15000	0	0.2	0.8	0.5	9.9	73	3.7	5	3.6	16	5.5	33
	25000	0	0.2	1.1	0.7	25.4	130	4.0	5	3.2	9	4.8	11
	50000	0	0.4	1.9	1.1	170.4	980	4.5	5	4.2	8	7.0	27
	75000	1	2.2	2.8	1.4	7 791.8	76 272	5.3	7	4.2	10	6.9	21
	100000	2	3.4	3.5	1.8	9 943.6	54 873	5.1	7	4.9	12	16.3	47
20000	30000	0	0.4	1.7	1.1	15.8	99	3.5	5	3.2	5	3.8	8
	50000	0	0.5	2.6	1.4	156.6	631	4.6	6	5.5	9	6.6	16
	100000	1	2.4	4.6	2.3	9 318.3	91 394	5.3	6	5.6	19	7.3	16
	150000	2	4.1	6.1	3.1	13 785.1	65 628	5.2	6	9.0	36	11.1	36
	200000	0	2.7	6.9	3.8	5 086.5	46 370	4.7	5	4.4	13	9.0	42
40000	60000	0	0.8	3.7	2.4	2.1	4	3.7	5	2.6	4	2.3	3
	100000	0	1.0	5.2	3.1	23.2	105	4.3	6	3.6	7	3.6	7
	200000	0	1.6	9.3	4.9	186.1	1 293	4.9	7	5.2	11	4.7	9
	300000	2	5.2	13.8	6.7	12 187.9	63 024	5.3	7	4.8	13	8.3	27
	400000	3	7.8	16.8	8.1	14 983.5	46 504	5.1	6	4.4	8	10.2	33
Média global			2.2	5.4	2.8	4 912.4	29 825.3	4.6	5.9	4.6	12.0	7.2	22.4

Tabela 4 – Resultados computacionais para $p = 0.8$.

#Nós	#Arcos	#NOpt Alg1	Tempos computacionais (seg)			Nº de CMC e nº de iterações do k CMC							
			Alg1	Alg2	Alg3	Alg1		Alg2				Alg3	
						# k CMC (Med)	# k CMC (Max)	#CMC (Med)	#CMC (Max)	# k CMC (Med)	# k CMC (Max)	# k CMC (Med)	# k CMC (Max)
10000	15000	0	0.2	0.8	0.5	9.8	73	3.7	5	4.9	24	7.2	42
	25000	0	0.2	1.1	0.7	15.7	130	4.0	5	3.0	5	3.5	8
	50000	0	0.4	1.9	1.1	32.4	261	4.5	5	4.3	10	6.7	23
	75000	1	2.2	2.8	1.5	7 643.8	76 272	5.3	7	5.6	13	9.8	27
	100000	2	3.4	3.3	1.8	9 769.8	54 873	4.9	6	4.0	8	17.7	50
20000	30000	0	0.4	1.7	1.1	13.2	99	3.5	5	3.5	5	3.4	5
	50000	0	0.5	2.5	1.5	29.9	248	4.5	6	4.0	8	5.5	22
	100000	0	0.8	4.4	2.3	7.4	47	5.0	6	3.7	9	5.0	17
	150000	0	1.6	5.9	3.1	2 793.3	27 473	5.0	6	3.4	7	4.9	11
	200000	0	2.6	6.9	3.9	4 656.8	46 370	4.6	5	3.5	8	10.6	52
40000	60000	0	0.8	3.7	2.4	2.0	4	3.7	5	3.3	9	2.4	3
	100000	0	1.0	5.1	3.1	10.9	33	4.2	6	4.3	10	4.7	12
	200000	0	1.6	9.4	5.0	142.9	1 293	4.8	6	5.7	12	5.6	13
	300000	1	3.7	13.6	6.9	6 076.4	57 475	5.1	6	3.8	11	8.4	44
	400000	1	4.5	16.9	8.3	4 919.3	40 883	5.1	6	5.9	17	9.8	15
Média global			1.6	5.3	2.9	2 408.2	20 368.9	4.5	5.7	4.2	10.4	7.0	22.9

6. Conclusões

O algoritmo de Handler e Zang [8] (Alg2), baseado na relaxação Lagrangeana e um dos mais referidos na literatura para tratar o problema do CMCRA, é um processo muito estável, obtendo a solução ótima para um número bastante reduzido de CMC e iterações dos k CMC. No entanto, o número de CMC, embora reduzido, torna o processo algo lento principalmente quando as redes têm dimensões mais elevadas (nas outras duas abordagens o número de CMC é ainda menor).

A abordagem clássica baseada nos k CMC (Alg1) tem a vantagem, relativamente a Alg2, de não calcular nenhum CMC entre dois nós, o que torna o processo extremamente rápido quando o algoritmo obtém a solução ótima num número reduzido de iterações do k CMC (o que normalmente acontece quando o valor do parâmetro p é próximo de 1). No entanto este processo é bastante instável à medida que o valor

de p se afasta de 1, necessitando, em alguns casos, para obter a solução óptima de um número proibitivo de iterações dos k CMC.

O algoritmo por nós sugerido (Alg3), baseado nos k CMC, é bastante mais estável que o Alg1. Relativamente a este calcula, para cada problema, mais dois CMC entre o nó final e o inicial mas tem a vantagem de não necessitar de muitas iterações nos k CMC, independentemente do valor do parâmetro p . Comparativamente ao Alg2, embora o número de iterações dos k CMC não seja, em média, tão reduzido, tem a vantagem de calcular sempre menos CMC entre dois nós, o que origina tempos computacionais bastante mais reduzidos, em alguns casos praticamente metade.

7. Referências

- [1] 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).
- [2] 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).
- [3] Cherkassky, B.V., Goldberg, A.V. e Radzik, T., “Shortest Paths Algorithms: Theory and Experimental Evaluation”, *Mathematical Programming* 73, 129-174 (1996).
- [4] Cohon, J., *Multiobjective Programming and Planning*, Academic Press, 1978.
- [5] Cohon, J. Church, R. e Sheer, D., “Generating Multiobjective Trade-offs: An Algorithm for Bicriterion Problems”, *Water Resources Research* 15, 1001-1010 (1979).
- [6] Dial, R.B., “Algorithm 360: Shortest Path Forest With Topological Ordering”, *Communications of the ACM* 12, 632-633 (1969).
- [7] Dreyfus, S., “An Appraisal of Some Shortest Path Algorithms”, *Operations Research* 17, 345-412, (1969).
- [8] Handler, G.Y. e Zang, I., “A Dual Algorithm for the Constrained Shortest Path Problem”, *Network* 10, 293-310 (1980).
- [9] Johsch, H.C., “The Shortest Route Problem with Constraints”, *J. Math. Analysis Appl.* 14, 191-197 (1966).
- [10] Santos, L. e Coutinho-Rodrigues, J., “Uma Nova Abordagem para o Cálculo do Caminho Mais Curto com uma Restrição Adicional Baseada no Método NISE Bi-objectivo”. Relatório de Investigação, INESC de Coimbra, nº 6 (2004).
- [11] Martins, E.Q.V., “An Algorithm for Ranking Paths that May Contain Cycles”, *European Journal of Operational Research* 18, 123-130 (1984).
- [12] Minoux, M., “Plus Courts Chemins avec Contraintes: Algorithmes et Applications”, *Ann. Télécommunications* 30, 383-394 (1975).
- [13] Shier, D.A., “Iterative Methods for Determining k -Shortest Paths in a Network”, *Network* 6, 205-230 (1976).

- [14] Yen, J.Y., "Finding the k Shortest Loopless Paths in a Network", *Management Science* 17, 712-716 (1971).
- [15] Zhan, F.B. e Noon C.E, "Shortest Path Algorithms on Real Road Networks", *Transportation Science* 32, 65-73 (1998).