



**Susana Nogueira  
da Silva**

**Problemas do caminho mais curto com  
restrições adicionais**



**Susana Nogueira  
da Silva**

**Problemas do caminho mais curto com  
restrições adicionais**

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Matemática e Aplicações, realizada sob a orientação científica do Dr. Agostinho Agra, Professor Auxiliar do Departamento de Matemática da Universidade de Aveiro.

**o júri**

presidente

**Prof. Dr. Domingos Moreira Cardoso**

Professor Catedrático da Universidade de Aveiro

vogais

**Prof<sup>a</sup> Dr<sup>a</sup> Isabel Maria de Jesus Martins**

Professora Auxiliar do Instituto Superior de Agronomia

**Prof. Dr. Agostinho Miguel Mendes Agra**

Professor Auxiliar da Universidade de Aveiro

## **agradecimentos**

Agradeço ao Professor Dr. Agostinho Agra pela orientação, pelo incentivo, pelo encorajamento e apoio incondicional disponibilizado ao longo de todo o período de realização desta dissertação.

Ao Sérgio, pelo apoio e compreensão pelos momentos em que estive ausente.

Aos meus pais e irmãs pelo apoio.

Finalmente uma palavra de apreço dirigida em especial a todos os meus amigos que directa ou indirectamente contribuíram para que este meu projecto se concretizasse.

**palavras-chave**

Problema do caminho mais curto, Problema de rotas de veículos, Relaxação

**resumo**

Neste trabalho estudam-se problemas do caminho mais curto com restrições adicionais. Este tipo de problemas tem variadas aplicações práticas onde é destacado o planeamento de rotas de veículos e o encaminhamento de mensagens em redes de comunicações. O problema de caminho mais curto com restrições adicionais tem tido também grande aplicação como sub-problema de outros problemas. É o caso do problema de caminho mais curto com janelas temporais que surge como sub-problema do problema de determinação de rotas de veículos com janelas temporais. É feita uma descrição das várias variantes do problema de caminho mais curto com restrições adicionais, é apresentada uma revisão da literatura sobre os métodos usados na resolução deste tipo de problemas e são descritas algumas aplicações deste tipo de problemas.

**keywords**

Shortest path problem, vehicle routing problem, relaxation

**abstract**

In this work, shortest path problems with additional constraints are studied. This type of problems has several practical applications, such as vehicle routing planning and the routing of messages in communications networks. The shortest path problem with additional constraints has also application as a sub-problem of other problems. This is the case of the shortest path problem with time windows occurring as sub-problem of the vehicle routing problem with time windows. We present a description of several variants of the shortest path problem with additional constraints, a short literature review on resolution methods for these problems and a description of some applications of this type of problems.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Conceitos Básicos</b>	<b>8</b>
2.1	Grafos . . . . .	8
2.2	Complexidade . . . . .	10
2.3	Programação Linear Inteira . . . . .	12
2.3.1	Problema de Optimização Combinatória . . . . .	15
2.3.2	Métodos de Resolução de um problema de <i>PLI</i> . . . . .	17
2.3.3	Relaxações . . . . .	18
<b>3</b>	<b>Problemas de Caminho mais Curto</b>	<b>24</b>
3.1	O Problema de Caminho mais Curto . . . . .	24
3.2	O Problema de Caminho Mais Curto Multiobjectivo . . . . .	27
3.3	O Problema de Caminho mais Curto com Restrições Adicionais . . . . .	32
3.3.1	O Problema do Caminho mais Curto com Janelas Temporais . . . . .	33
3.3.2	O Problema do Caminho mais Curto com Restrições de Recurso . . . . .	35
3.3.3	Programação Dinâmica . . . . .	38
3.3.4	Revisão da Literatura . . . . .	42
<b>4</b>	<b>Aplicações: Modelação de Problemas Reais</b>	<b>46</b>
4.1	Planeamento de caminhos para Gestão de Frotas . . . . .	46
4.2	Redes de Comunicação . . . . .	53

## CONTEÚDO

---

4.3	Outras aplicações . . . . .	56
4.3.1	Aproximação de Curvas . . . . .	56
4.3.2	Modelação de Aplicações de Engenharia . . . . .	58
4.3.3	Sistema de Gestão de Aviões Militares . . . . .	59
4.4	Redução de outros problemas . . . . .	59
<b>5</b>	<b>Aplicação: sub-problema de outros problemas</b>	<b>64</b>
5.1	Rotas de Veículos em Problemas com Janelas Temporais . . . . .	65
5.2	Relaxações Lagrangeanas . . . . .	68
5.3	Geração de Colunas . . . . .	74
5.3.1	Geração de colunas na resolução da relaxação linear do PCMCRR	75
5.3.2	Aplicação ao Problema de Planeamento de Rotas . . . . .	78
<b>6</b>	<b>Conclusões</b>	<b>84</b>



# Capítulo 1

## Introdução

O Problema de Caminho Mais Curto (PCMC) é um dos problemas clássicos em Optimização e um dos problemas fundamentais da Computação, que vem mantendo o interesse de muitos investigadores. Esta classe de problemas é frequentemente colocada em prática numa grande variedade de aplicações nas quais se deseja realizar algum tipo de transporte entre dois ou mais pontos específicos de uma rede, ao menor custo, o mais rapidamente possível ou de uma forma mais fiável.

Por vezes, na prática, não se deseja apenas determinar o caminho de menor custo ou o caminho mais rápido, mas uma combinação de diferentes objectivos, como por exemplo, determinar um caminho que seja rápido e barato. Este é conhecido como um PCMC multi-objectivo.

Outro tipo de variantes do PCMC ocorrem quando há restrições ao caminho a escolher. Este tipo de problemas será designado nesta tese por Problemas de Caminho Mais Curto com Restrições Adicionais (PCMCRA). Estas restrições podem ser de diversos tipos, por exemplo:

- (i) Restrições de recurso. Neste caso o problema consiste em encontrar um caminho de menor custo e satisfazendo as capacidades máximas disponíveis dos vários

---

recursos, em que assume-se que percorrer um arco implica o consumo de um recurso escasso, pelo que o caminho a ser escolhido deve garantir que o consumo do recurso ao longo dos arcos do caminho não ultrapassa a capacidade máxima desse recurso;

- (ii) Restrições que impõem janelas temporais nos nós. Neste caso o problema consiste em encontrar um caminho de menor custo entre dois nós de um grafo, em que aos arcos estão associados tempos de percurso e associado a cada nó existe uma janela temporal que garante que o nó só pode ser visitado nesse intervalo de tempo.

A importância deste tipo de problemas está naturalmente associada às inúmeras aplicações existentes, onde se destacam os problemas de encaminhamento. Estes problemas consistem em encontrar o melhor percurso numa rede viária, numa frota de aviões, numa rede de linhas de metro e também efectuar o encaminhamento de tráfego em redes de comunicações.

Este tipo de problemas tem tido igualmente grande aplicação como sub-problema de outros problemas. É o caso do problema do caminho mais curto com janelas temporais que surge como sub-problema do problema do Caixeiro Viajante com janelas temporais.

Nesta tese são consideradas várias variantes de problemas do caminho mais curto. São descritas algumas aplicações do PCMCRA a problemas reais e é efectuada a descrição de uma aplicação real onde o PCMCRA surge como sub-problema desse problema mais geral.

### Organização da Tese

O trabalho está organizado em três grupos: conceitos básicos, variantes do problema e aplicações.

No capítulo 2, é feita uma breve referência a conceitos sobre teoria dos grafos, complexidade, programação linear inteira e relaxações. Estes conceitos serão necessários ao longo da tese.

No capítulo 3, descreve-se o problema do caminho mais curto, algumas das suas variantes que incluem restrições adicionais, a abordagem de Programação Dinâmica para um desses problemas e faz-se um resumo das principais referências bibliográficas relativamente a trabalhos efectuados com este tipo de problemas.

No capítulo 4, é feita a descrição de algumas aplicações do PCMCRA a situações reais. Como é impraticável descrever pormenorizadamente todas as aplicações, descrevem-se com algum detalhe algumas aplicações: uma aplicação do PCMC com restrições adicionais ao planeamento de caminhos para gestão de frotas; uma aplicação no contexto das redes de comunicação e uma aplicação indirecta através da reformulação do problema da determinação da sub-árvore óptima de uma árvore com restrições de capacidade como um problema de caminho mais curto com restrições de recurso. Finalmente, é feito um pequeno resumo de outras aplicações práticas.

No capítulo 5, descrevem-se alguns problemas de caminho mais curto com restrições adicionais que surgem como sub-problemas de outros problemas mais gerais. Será dado ênfase à aplicação deste tipo de sub-problemas a problemas de rotas de veículos.

---

# Capítulo 2

## Conceitos Básicos

Neste capítulo é feita uma breve referência a conceitos sobre grafos, complexidade, programação linear e relaxações. Estes conceitos serão necessários ao longo da tese.

### 2.1 Grafos

Nesta secção introduzimos algumas noções elementares da teoria dos grafos.

Um grafo é um par  $G = (N, A)$ , onde  $N = \{v_1, \dots, v_n\}$  é um conjunto finito, cujos elementos são denominados por nós e  $A = \{a_1, \dots, a_m\}$  é também um conjunto finito, cujos elementos são denominados por arcos.

Cada arco  $a_k$  pode ainda ser identificado por um par, ordenado ou não,  $[i, j]$  onde  $i$  e  $j$  são dois nós do grafo. Se  $[i, j]$  for um par ordenado será representado por  $(i, j)$  e diz-se arco orientado ou simplesmente arco, caso contrário, diz-se um arco não orientado ou aresta e representa-se pelo conjunto dos nós  $\{i, j\}$ .

O grafo  $(N, A)$  é um grafo orientado se  $A$  contiver apenas pares ordenados de nós, é um grafo não orientado se  $A$  contiver apenas pares não ordenados e é um grafo misto se contiver pares ordenados e pares não ordenados.

## 2.1 Grafos

---

Graficamente, um grafo pode ser representado como na figura (2.1), onde os nós são representados por círculos, enquanto que os arcos orientados são representados por uma seta e os não orientados por um segmento de recta.

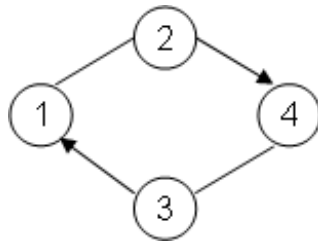


Figura 2.1: Exemplo de um grafo

Caso não seja referido algo em contrário, assume-se que o grafo a considerar,  $G = (N, A)$ , é um grafo orientado.

Sejam  $s$  e  $t$  dois nós de um grafo  $G = (N, A)$ , respectivamente nó inicial e nó final.

Um passeio de  $s$  para  $t$  em  $(N, A)$  é uma sequência, constituída alternadamente por nós e arcos, da forma  $p = v_1, a_1, v_2, \dots, a_{l-1}, v_l$ , onde  $l \geq 2$  e:

- $v_i \in N$ , para  $i \in \{1, \dots, l\}$ ;
- $v_1 = s$  e  $v_l = t$
- $a_i \equiv (v_i, v_{i+1}) \in A$ , para  $i \in \{1, \dots, l-1\}$ .

Um caminho de  $s$  para  $t$  em  $(N, A)$  é um passeio com início em  $s$  e fim em  $t$ , onde:

- $v_i \neq v_j$ , para  $i, j \in \{2, \dots, l-1\}$  tais que  $i \neq j$ ;
- $s \neq v_i$  e  $t \neq v_i$ , para  $i \in \{2, \dots, l-1\}$ .

Isto é, um caminho é um passeio sem repetição de nós, com eventual excepção do nó inicial e final.

Um ciclo é um caminho de um qualquer nó para ele mesmo. No caso de grafos orientados assume-se que um ciclo contém todos os arcos orientados na mesma direcção.

Num grafo, um ciclo hamiltoniano é um ciclo que passa por todos os vértices uma única vez.

Um grafo diz-se acíclico se não contiver ciclos.

De uma maneira geral, os conceitos utilizados para grafos são extensíveis a redes.

Uma rede é um grafo cujos nós e/ou arestas têm associado valores numéricos (custos, capacidades e/ou oferta e procura).

Uma rede pode ser representada por  $(N, A, C)$ , em que  $G = (N, A)$  é um grafo e  $C$  é o conjunto dos valores numéricos, ou mais genericamente vectores, associados aos arcos (comprimentos dos arcos), ou seja, ao arco  $(i, j)$  está associado o valor, ou mais genericamente vector,  $c_{ij}$ , isto é, o comprimento do arco  $(i, j)$  é  $c_{ij}$ .

Considere-se um caminho  $p$  de  $s$  para  $t$  num grafo  $G$ .

O comprimento do caminho  $p$ ,  $c(p)$ , é a soma dos comprimentos dos arcos que pertencem àquele caminho, ou seja,  $c(p) = \sum_{(i,j) \in p} c_{ij}$ .

O conjunto de todos os caminhos de  $s$  para  $t$  identifica-se por  $P$ .

Ao caminho de menor comprimento dá-se o nome de *caminho mais curto*.

## 2.2 Complexidade

Nesta secção são formalizados alguns conceitos de complexidade, com a finalidade de estabelecer uma classificação dos problemas segundo a “dificuldade” da sua resolução.

## 2.2 Complexidade

---

Um algoritmo é um procedimento computacional bem definido que aceita um valor (ou conjunto de valores) como *input* e produz um valor (ou conjunto de valores) como *output*. Pode-se dizer ainda que um algoritmo é uma sequência de passos computacionais que transformam um *input*, num *output* ou que é uma ferramenta para resolver um problema computacional bem definido; problema esse que define a relação *Input/Output* pretendida.

A medida do tempo de resolução de um problema é definida de acordo com a grandeza da concretização. A complexidade é medida em termos de recursos gastos na resolução do problema, e onde o recurso mais importante é o tempo.

Diz-se que um algoritmo tem um gasto de tempo polinomial se o tempo que demora até obter a solução pretendida é da ordem  $\mathcal{O}(n^k)$ , isto é, se existe um  $k$  tal que  $f(n) = \mathcal{O}(g(n))$ , onde  $f(n) \leq k |g(n)|$  e  $n$  é a grandeza da concretização.

Por exemplo, se  $k = 1$ , o tempo de execução do algoritmo diz-se linear e se  $k = 2$  diz-se quadrático. Se a ordem for  $\mathcal{O}(a^{h(n)})$ , onde  $a > 1$  e  $h(n)$  é polinomial, então o tempo gasto na execução do problema é exponencial.

Os problemas que se resolvem em tempo polinomial são chamados tratáveis ou fáceis e pertencem à classe  $P$ . Os problemas para os quais não são conhecidos algoritmos polinomiais são chamados de difíceis.

Existe uma classe mais geral de problemas que inclui os problemas em  $P$ . Trata-se da classe  $NP$  que se define como a classe dos problemas de decisão que podem ser resolvidos em tempo polinomial numa máquina de Turing não-determinista. A designação  $NP$  tem origem em “*solvable by a Nondeterministic Turing machine in polynomial time*” [8].

Existe ainda uma sub-classe dos problemas  $NP$  que reúne problemas importantes para sua vasta aplicação, a classe dos problemas  $NP$ -Completo. São problemas para os quais ainda não se encontrou um algoritmo polinomial, mas que se alguma vez



se encontrar um para um qualquer problema desta classe, então existe um algoritmo polinomial para todos os problemas pertencentes a esta classe.

Conjecturando-se, que este tipo de algoritmos não existe, havendo a distinção entre a classe de problemas  $P$  e  $NP$ , isto é  $P \neq NP$ , então as diferentes classes de problemas estão dispostas de acordo com a figura (2.2).

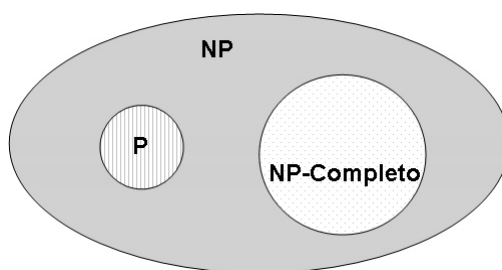


Figura 2.2: Classes de Complexidade

A classe dos problemas  $NP$ -Difícil é constituída pelos problemas de otimização cujo correspondente problema de decisão está na classe  $NP$ -Completo [49].

A resolução de problemas das classes  $NP$  é um dos maiores desafios das Ciências da Computação.

## 2.3 Programação Linear Inteira

Nesta secção é feita uma revisão de alguns conceitos básicos de Programação Linear e de Programação Linear Inteira. São também descritos alguns métodos de resolução de um problema de Programação Linear Inteira.

Considere-se o problema de Programação Linear ( $PL$ ) na sua forma canónica:

$$\max \{cx : Ax \leq b, x \geq 0\}$$

onde  $A$  é uma matriz com  $m$  linhas e  $n$  colunas,  $c$  é um vector linha de dimensão  $n$ ,  $b$  é um vector coluna de dimensão  $m$  e  $x$  é um vector coluna de dimensão  $n$  de variáveis

## 2.3 Programação Linear Inteira

---

desconhecidas.

Geometricamente, as restrições lineares, definem um poliedro, que é chamado de conjunto das soluções admissíveis. Assim, um poliedro  $P$  é um subconjunto de  $\mathbb{R}^n$  descrito por um número finito de desigualdades lineares:  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ .

Sabe-se que existem algoritmos polinomiais (método do elipsóide ou método do ponto interior [44]) que permitem resolver qualquer instância de um problema de  $PL$ . Neste caso, diz-se que estes problemas pertencem à classe  $P$  [8]. Os problemas de Programação Linear podem ser igualmente resolvidos pelo Método do Simplex [5] que embora não seja polinomial, na prática, tem mostrado um comportamento muito eficaz.

Um problema de Programação Linear Inteira ( $PLI$ ) é um problema de Programação Linear ( $PL$ ) em que todas ou algumas das suas variáveis são discretas e têm de assumir valores inteiros.

Num modelo de  $PLI$  podem ocorrer as seguintes situações:

- todas as variáveis de decisão são inteiras, isto é, todas as variáveis estão sujeitas à condição de integralidade; neste caso, está-se perante um problema de Programação Linear Inteira Pura:

$$\max \{cx : Ax \leq b, x \geq 0, x \in \mathbb{Z}\}$$

- só parte das variáveis de decisão são inteiras; neste caso, trata-se de um problema de Programação Linear Inteira Mista:

$$\max \{cx + dy : Ax + Ey \leq b, x \geq 0, y \geq 0, y \in \mathbb{Z}\}$$

onde  $A$  é novamente uma matriz do tipo  $m \times n$ ,  $E$  é uma matriz do tipo  $m \times p$ ,  $d$  é um vector linha de dimensão  $p$  e  $y$  é um vector coluna de dimensão  $p$  de

variáveis inteiras desconhecidas.

- todas as variáveis de decisão são binárias, isto é, todas as variáveis têm como valores admissíveis 0 ou 1; neste caso, está-se perante um problema de Programação Linear Inteira Binária:

$$\max \{cx : Ax \leq b, x \in \{0, 1\}^n\}$$

Poderia parecer que os problemas de *PLI* são relativamente fáceis de resolver, já que existem muito menos soluções a serem consideradas do que num problema de *PL*. Acontece que, mesmo que os problemas de *PLI* cujo conjunto das soluções possíveis seja limitado têm um número finito de soluções possíveis. No entanto, a simples introdução das restrições de integralidade das variáveis num problema de *PL*, transforma-o num problema de características diferentes. Em geral, os problemas de *PLI* são *NP*-difíceis, portanto, não é conhecido qualquer algoritmo que permita obter a solução ótima de toda a instância desse tipo de problemas, num número de passos que possa ser majorado por uma função polinomial do tamanho da instância. E se, tal como é conjecturado,  $P \neq NP$ ; então não existirá tal algoritmo.

Quando um problema de *PLI* tem um número finito de soluções possíveis, pode surgir (erradamente) a ideia da enumeração exaustiva como abordagem simples e eficiente para a determinação da solução ótima: é que um número finito não é necessariamente um número pequeno. De facto, o número de soluções possíveis pode facilmente tornar-se demasiado grande, inviabilizando do ponto de vista computacional a sua enumeração exaustiva. Torna-se portanto necessário usar métodos que examinem apenas uma parte do conjunto de soluções possíveis.

## 2.3 Programação Linear Inteira

---

### 2.3.1 Problema de Optimização Combinatória

Um tipo de problemas nos quais se enquadram os PCMC são os denominados Problemas de Optimização Combinatória (POC).

Considere-se um conjunto finito  $M = \{1, \dots, m\}$ , com pesos  $c_j$  para cada  $j \in M$  e um conjunto  $F$  de subconjuntos admissíveis de  $M$ . O problema de encontrar um subconjunto admissível de peso mínimo é um *POC*:

$$\min_{S \subseteq M} \left\{ \sum_{j \in S} c_j : S \subseteq F \right\}$$

Os *POC* podem ser formulados com problemas de programação linear binária, recorrendo a variáveis binárias associadas a cada objecto de  $M$  que indicam se o objecto pertence ou não a  $S$ . Os vectores cujos componentes são estas variáveis binárias designam-se por vectores de incidência.

Veja-se o exemplo do Problema do Caixeiro Viajante (*PCV*). O *PCV* é talvez o problema mais notório da Investigação Operacional, pois é muito fácil de explicar e muito tentador para resolver.

No problema, o Caixeiro tem que visitar  $n$  cidades exactamente uma e uma só vez e regressar ao ponto de partida. O tempo de duração da viagem entre as cidades  $i$  e  $j$  é dado por  $c_{ij}$ . O problema consiste em encontrar a ordem pela qual o caixeiro deve visitar todas as cidades no menor tempo possível.

Seja  $G = (N, A)$  um grafo onde  $N = \{1, \dots, n\}$  é o conjunto dos nós e  $A$  o conjunto dos arcos e cujos pesos nos arcos  $(i, j)$  são denotados por  $c_{ij}$ .

De seguida, é formulado o problema do Caixeiro Viajante como um POC. Seja  $M = A$  e  $F = \{S \subseteq A : S \text{ é um conjunto de arcos que formam um ciclo hamiltoniano}\}$ ,

então o *PCV* pode ser formulado do seguinte modo:

$$\min_{S \subseteq M} \left\{ \sum_{(i,j) \in S} c_{ij} : S \subseteq F \right\}.$$

O *PCV* pode ser também formulado como um problema de PLI:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{2.1}$$

sujeito a:

$$\sum_{j:(i,j) \in A} x_{ij} = 1, \quad \forall i \in N \tag{2.2}$$

$$\sum_{i:(i,j) \in A} x_{ij} = 1, \quad \forall j \in N \tag{2.3}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \tag{2.4}$$

em que:

$$x_{ij} = \begin{cases} 1, & \text{se o caixeiro vai directamente da cidade } i \text{ para a cidade } j; \\ 0, & \text{no caso contrário.} \end{cases} \quad \forall (i, j) \in A$$

A restrição (2.2) garante que o caixeiro sai da cidade  $i$  apenas uma vez e a restrição (2.3) garante que entra na cidade  $j$  apenas uma vez.

A solução do problema, definido por (2.1) - (2.4) (problema de afectação), pode conter soluções com ciclos e sem qualquer ligação entre eles, ver figura (2.3).

Para eliminar este tipo de soluções, são necessárias mais restrições para garantir a conectividade, impondo que o caixeiro tenha que passar de um conjunto de cidades para outro. Estas restrições são denominadas por restrições de corte:

$$\sum_{(i,j) \in A: i \in S, j \notin S} x_{ij} \geq 1, \quad \forall S \subset N, S \neq \emptyset.$$

## 2.3 Programação Linear Inteira

---

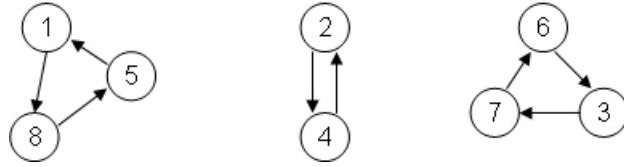


Figura 2.3: Existência de sub-ciclos

Com base na figura (2.3), considere-se um exemplo de uma restrição de corte.

Seja  $S = \{1, 5, 8\}$ . Assim,

$$x_{12} + x_{13} + x_{14} + x_{16} + x_{17} + x_{52} + x_{53} + x_{54} + x_{56} + x_{57} + x_{82} + x_{83} + x_{84} + x_{86} + x_{87} \geq 1$$

pelo que esta restrição de corte não é satisfeita pela solução apresentada, pois todas as variáveis do lado esquerdo da desigualdade são nulas.

Uma alternativa é substituir estas restrições por restrições de “eliminação de sub-ciclos”:

$$\sum_{(i,j) \in A: i \in S, j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset N, 2 \leq |S| \leq n - 1.$$

Este tipo de restrições também são utilizadas para modelar problemas de caminho mais curto, como será visto mais adiante nesta tese.

### 2.3.2 Métodos de Resolução de um problema de *PLI*

Quando os problemas de Optimização Combinatória são *NP*-difíceis, existem métodos gerais de resolução não polinomiais. Provavelmente, o mais conhecido é o *Branch and Bound* que consiste em subdividir (“*Branch*”) o espaço das soluções admissíveis obtendo deste modo um conjunto de sub-problemas “mais simples”, cada um correspondendo a um conjunto de soluções admissíveis diferente. A cada problema faz-se corresponder um nó na árvore de enumeração. De modo a evitar proceder a uma enumeração completa (exaustiva) de todos os possíveis sub-problemas (usualmente em número exponencial)

procura-se cancelar essa pesquisa (“*Bound*”) em cada nó (sub-problema) através de uma das seguintes regras:

- (i) provando que o sub-problema não tem soluções admissíveis;
- (ii) obtendo a solução óptima desse sub-problema (sem ramificar);
- (iii) provando que todas soluções desses sub-problemas têm um valor da função objectivo pior ou igual ao valor de uma solução admissível já encontrada.

Dependendo do tipo de problemas, pode ser possível, como na generalidade dos problemas tratados nesta tese, recorrer a algoritmos de programação dinâmica.

Como os algoritmos exactos conhecidos para problemas *NP*-difíceis são não polinomiais, na prática, os tempos de execução podem ser incontroláveis. Por isso, recorre-se com frequência às chamadas heurísticas.

Uma Heurística é uma técnica para obter boas soluções, com um custo computacional aceitável, muitas vezes baseada em procedimentos simples e intuitivos. Não oferece garantias de optimalidade, nem de admissibilidade, e pode nem sequer dar informação da qualidade duma dada solução admissível. Baseiam-se no compromisso entre a qualidade da solução e o tempo necessário para a obter.

### 2.3.3 Relaxações

Para avaliar as soluções admissíveis encontradas é usual recorrer a relaxações do problema. Assim, as relaxações têm como objectivo obter bons limites inferiores (superiores) para problemas de minimização (maximização) de modo a diminuir a árvore de pesquisa do *Branch and Bound* (o número de nós depende da qualidade dos limites encontrados) e a avaliar a qualidade das soluções obtidas pelas heurísticas.

## 2.3 Programação Linear Inteira

---

Por exemplo, num problema de minimização, o valor da função objectivo de uma solução admissível fornece um limite superior  $\bar{z}$ , para o valor óptimo do problema  $z^*$ . Por seu lado, o valor óptimo de uma relaxação fornece um limite inferior  $\underline{z}$  para  $z^*$ . Assim, com base numa solução admissível e na resolução de um problema que seja uma relaxação do problema original é possível enquadrar o valor óptimo  $\underline{z} \leq z^* \leq \bar{z}$ , ver figura (2.4).

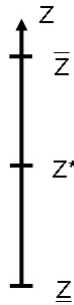


Figura 2.4: Limites superiores e inferiores para o valor óptimo.

Note-se que o valor das relaxações é fundamental no ponto (iii) do método de *Branch and Bound* pois é majorando (problema de maximização) ou minorando (problema de minimização) que se pode provar que o valor óptimo de um sub-problema é pior ou igual ao valor de uma solução admissível entretanto encontrada (por exemplo usando uma heurística).

Dada a importância das relaxações nesta tese, é descrito com cuidado o conceito de relaxação.

Seja  $Q$  um Problema de Programação Inteira. Obtém-se uma relaxação de  $Q$ ,  $\bar{Q}$ , aumentando o conjunto de soluções admissíveis e/ou substituindo a função objectivo por uma outra função objectivo que tenha um valor igual ou superior (inferior) ao da função original, para os problemas de maximização (minimização), em toda a solução admissível. Verifica-se que caso uma relaxação não tenha soluções admissíveis, então o problema original também não tem [49].



Existem vários tipos de relaxações para obtenção de limites para o valor óptimo de um problema, entre elas, as mais comuns e descritas neste trabalho são a Relaxação Linear, a Relaxação Lagrangeana e a Relaxação Combinatória.

### 1. Relaxação Linear

A relaxação linear é uma das principais relaxações de problemas de  $PLI$ . Basicamente consiste em relaxar (ignorar) as restrições que obrigam algumas variáveis a assumir valores inteiros.

**Definição:** Para o problema de programação inteira  $z^{PLI} = \max \{cx : x \in P \cap Z^n\}$  com formulação  $P = \{x \in \mathbb{R}_+^n : Ax \leq b\}$ , a relaxação linear é dada por  $z^{PL} = \max \{cx : x \in P\}$ .

No caso de um problema de  $PLI$  binária,  $z^{PLI} = \max \{cx : x \in P \cap \{0, 1\}^n\}$ , a relaxação linear é dada por  $z^{PL} = \max \{cx : x \in P, x \leq \mathbf{1}\}$ , onde  $\mathbf{1}$  é o vector com uns (1) em todos os componentes.

A relaxação linear pode ser resolvida eficientemente, por exemplo, por métodos não polinomiais, método do Simplex [5], ou por métodos polinomiais, como o método do elipsóide e métodos de pontos interiores [44].

Por vezes o valor óptimo da relaxação linear coincide com o do problema inteiro, isto é,  $z^{PL} = z^{PLI}$ . Neste caso, em vez se resolver o problema inteiro pode ser resolvido o problema linear.

### 2. Relaxação Combinatória

Sempre que a relaxação de um POC é um outro problema de Optimização Combinatória, diz-se que é uma relaxação combinatória. Em muitos casos, como no exemplo seguinte, a relaxação é um problema fácil que pode ser resolvido rapidamente.

#### Problema do Caixeiro Viajante

Na formulação do problema do caixeiro viajante, com o grafo  $G = (N, A)$  e os pesos nos arcos  $(i, j)$  denotados por  $c_{ij}$ , os caminhos encontrados são precisamente as afectações que não contêm sub-ciclos (2.5). Assim,

$$z^{PCV} = \min_{T \subseteq A} \left\{ \sum_{(i,j) \in T} c_{ij} : T \text{ forma um ciclo} \right\} \geq z^{AF}$$
$$z^{AF} = \min_{T \subseteq A} \left\{ \sum_{(i,j) \in T} c_{ij} : T \text{ forma uma afectação} \right\}$$

onde  $z^{PCV}$  é o valor óptimo do Problema do Caixeiro Viajante e  $z^{AF}$  é o valor óptimo do Problema de Afectação.

### 3. Relaxação Lagrangeana

A Relaxação Lagrangeana é uma técnica bem conhecida e usada frequentemente na obtenção de majorantes/minorantes para problemas de Optimização Combinatória [41].

Considere-se o seguinte problema de programação linear (inteira):

$$z^* = \max \{cx : Ax \leq b, x \in X \subseteq Z^n\}$$

Suponha-se que o conjunto de restrições  $Ax \leq b$  complicam o problema. Uma relaxação pode ser obtida pela eliminação do conjunto de restrições  $Ax \leq b$  do problema, sem as quais o problema ficaria simples de resolver. Ao invés de eliminar simplesmente essas restrições, a relaxação Lagrangeana incorpora-as na função objectivo, penalizando as soluções que violam as restrições relaxadas:

$$\max \{cx + \mu^T(b - Ax) : x \in X\},$$

onde  $\mu \geq 0$ .

**Proposição:** Seja  $z(\mu) = \max \{cx + \mu(b - Ax) : x \in X\}$ . Então  $z(\mu) \geq z^*$ , para todo  $\mu \geq 0$ .

**Prova:** Seja  $x^*$  uma solução ótima para o problema de PLI. Como  $x^*$  é admissível no problema de *PLI*,  $x^* \in X$ . Logo  $Ax^* \leq b$ . Como  $\mu \geq 0$ , vem  $cx^* \leq cx^* + \mu(b - Ax^*) \leq z(\mu)$ . De onde se concluiu que  $z(\mu) \geq z^*$ . ■

De acordo com a proposição anterior, a solução ótima do problema relaxado, para qualquer  $\mu \geq 0$ , induz um limite superior  $z(\mu) \geq z^*$ , onde  $z^*$  é o valor da solução ótima. Uma vez que  $z(\mu)$  é um limite superior de  $z^*$ , naturalmente deseja-se minimizá-lo, dando origem ao problema Dual Lagrangeano:

$$\min_{\mu \geq 0} z(\mu) = \min_{\mu \geq 0} \left\{ \max \{cx + \mu(b - Ax), \quad x \in X\} \right\}$$

Um método bastante utilizado para encontrar uma solução aproximada para o problema Dual Lagrangeano é o Método do Subgradiente, que não é explorado nesta tese.

## 2.3 Programação Linear Inteira

---

# Capítulo 3

## Problemas de Caminho mais Curto

O problema de caminho mais curto é um dos problemas fundamentais da Optimização. Tem sido estudado com profundidade e são conhecidos alguns algoritmos eficientes no caso em que os pesos são positivos, como por exemplo o algoritmo de Dijkstra [3]. Esta classe de problemas ocorre frequentemente numa grande variedade de situações práticas nas quais se deseja realizar algum tipo de transporte entre dois ou mais pontos específicos de um grafo, com o menor custo ou menor tempo possível.

### 3.1 O Problema de Caminho mais Curto

Os problemas de caminho mais curto sem restrições adicionais e com um só objetivo são fundamentais e frequentes, principalmente no estudo de problemas em redes, por exemplo de redes de transportes ou redes de comunicações. Este problema surge quando se pretende determinar o caminho mais curto, mais barato ou mais fiável, entre um ou vários pares de nós de uma rede. Estes problemas surgiram a partir de adaptações a uma grande variedade de problemas práticos, não só como modelos únicos mas também como sub-problemas de problemas mais complexos. Por exemplo, surgiram nas indústrias de telecomunicações e de transportes sempre que se pretendia enviar uma mensagem, ou um veículo, entre dois locais geograficamente distantes, o mais rápido ou o mais barato possível.

### 3.1 O Problema de Caminho mais Curto

---

Sejam  $s$  e  $t$  dois nós de um grafo  $G = (N, A)$ , em que a cada arco está associado apenas um valor,  $c_{ij}$ , que corresponde ao comprimento do arco  $(i, j)$ . Seja  $c(p)$  o comprimento de um caminho  $p$  de  $s$  para  $t$  no grafo  $G$  e seja  $P$  o conjunto de todos os caminhos de  $s$  para  $t$ . Com a resolução de um problema de caminho mais curto entre os nós  $s$  e  $t$ , no grafo  $G$ , pretende-se determinar o caminho de menor comprimento existente em  $P$ ; isto é, determinar  $p \in P$ , tal que  $c(p) \leq c(q)$ , para todo o  $q \in P$ .

Este problema pode ser formulado como um POC da seguinte forma:

$$\min\{c(p) : p \in P\}$$

e pode ainda ser formulado como um problema de PLI, no caso em que não existem ciclos de comprimento negativos, da seguinte forma:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (3.1)$$

sujeito a:

$$\sum_{j:(s,j) \in A} x_{sj} = 1, \quad (3.2)$$

$$\sum_{j:(j,t) \in A} x_{jt} = 1, \quad (3.3)$$

$$\sum_{i:(i,j) \in A} x_{ij} - \sum_{k:(j,k) \in A} x_{jk} = 0, \quad \forall j \in N \setminus \{s, t\} \quad (3.4)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \quad (3.5)$$

em que,

$$x_{ij} = \begin{cases} 1, & \text{se o arco } (i, j) \text{ pertence ao caminho,} \\ 0, & \text{se o arco } (i, j) \text{ não pertence ao caminho.} \end{cases} \quad \forall (i, j) \in A$$

e onde a equação (3.2) garante que sai apenas um arco da origem  $s$ , a equação (3.3) garante que chega um só arco ao destino  $t$ , as equações (3.4) representam as restrições

de conservação de fluxo em cada nó, isto é, se chega um arco ao nó  $j$  então de  $j$  também tem que sair um arco e (3.5) garante que as variáveis de decisão  $x_{ij}$  tomam apenas o valor 0 ou 1, nas condições anteriormente definidas.

Pode-se ainda observar que: (ver [37])

- Se  $c_{ij} > 0, \forall (i, j) \in A$ , o comprimento de um caminho é maior do que o de qualquer dos seus sub-caminhos. Neste caso, o problema é polinomial pertencendo à classe P;
- Se  $c_{ij} > 0, \forall (i, j) \in A$ , qualquer sub-caminho de um caminho mais curto, é ele próprio um caminho mais curto (princípio da optimalidade);

Note-se que esta formulação (3.2)-(3.5) não impede a ocorrência de soluções que incluam ciclos e que, portanto, não são caminhos, conforme se pode observar pelo exemplo apresentado na figura (3.1) que satisfaz o problema (3.2)-(3.5).

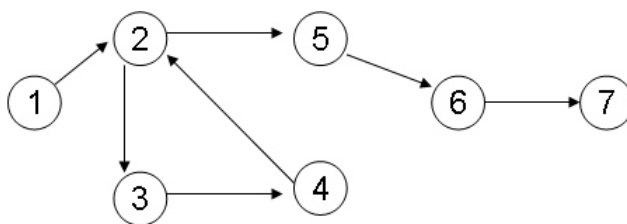


Figura 3.1: Exemplo de um passeio com ciclos

Contudo, se não existirem ciclos de comprimento negativo então as soluções (extremas) que se obtêm resolvendo a relaxação linear de (3.2)-(3.5) são caminhos.

No caso geral, isto é, de poderem existir ciclos de peso negativo, é necessário acrescentar à formulação anterior as restrições de eliminação de ciclos, de modo a garantir

## 3.2 O Problema de Caminho Mais Curto Multiobjectivo

---

a inexistência de ciclos:

$$\sum_{(i,j) \in A: i \in S, j \in S} x_{ij} \leq |S| - 1, \quad \forall S \neq \emptyset, \quad S \subseteq N \quad (3.6)$$

Para  $|S| = k$ , um ciclo envolvendo todos os nós em  $S$ , contém, pelo menos,  $k$  arcos. Assim, as restrições anteriores garantem a existência de, no máximo,  $k - 1$  arcos para o subconjunto  $S$  de nós. Considerando todo o subconjunto de nós  $S$ , com  $S \neq N$  e  $S \neq \emptyset$  garante-se que não ocorrem ciclos na solução.

No exemplo anterior considerando  $S = \{2, 3, 4\}$  facilmente se verifica que a correspondente solução não satisfaz (3.6).

Neste caso mais geral, o PCMC é *NP*-difícil.

## 3.2 O Problema de Caminho Mais Curto Multiobjectivo

Muitos problemas reais podem ser modelados como um problema de caminho mais curto multiobjectivo, em que os dados do modelo são organizados recorrendo a uma estrutura em rede. Como exemplo, o caso de circulação de veículos. Pretende-se determinar o melhor caminho entre dois nós de uma rede de estradas (por exemplo, entre um ponto onde se carregam componentes que devem ser transportadas para uma fábrica localizada num outro ponto da rede), tendo em conta um compromisso entre dois objectivos: distância percorrida e tempo gasto. O tempo gasto num percurso pode ser influenciado por atrasos na entrega da mercadoria, uma vez que o tempo de circulação em cada artéria (arco da rede) pode variar muito, devido a problemas de congestão de tráfego, acidentes, obras de manutenção, filas de espera para pagamentos de portagens, etc.



Na prática, muitas vezes não se deseja apenas determinar o caminho de menor custo ou mais rápido, mas uma combinação de diferentes critérios, por exemplo, um caminho que seja rápido e barato. Os problemas multiobjectivo referem-se aos casos em que as alternativas são definidas implicitamente por um conjunto de restrições e os critérios são considerados através de funções a otimizar. Como não é possível otimizar sobre todos os critérios simultaneamente, é usual proceder à ponderação de cada critério obtendo deste modo uma única função objectivo que se deseja minimizar.

Como exemplo, podem-se identificar os seguintes problemas que envolvem múltiplos critérios:

- Escolher o melhor local para a construção de uma ponte, em que os critérios poderiam ser o custo, o impacto sobre o rio (ambiental e utilização do rio), o volume de tráfego, o impacto sobre as margens, a estética, o custo da travessia, etc..
- Encontrar o caminho mais económico, para efectuar a entrega/recolha de produtos aos clientes duma determinada firma, em que os critérios poderiam ser o tempo, a distância, o atraso, o tráfego, etc..

Por existir conflito entre os critérios, a entidade que tem a responsabilidade de decidir, tem que ponderar os compromissos a efectuar com vista a encontrar a solução que lhe pareça mais satisfatória. Neste caso, está-se perante um problema multicritério.

O problema multicritério está relacionado com os métodos e procedimentos, pelos quais os vários critérios podem ser formalmente associados no processo de análise. De uma forma geral, estes problemas dividem-se em problemas multiatributo e multiobjectivo. Os primeiros caracterizam-se pela existência de um número finito de alternativas explicitamente conhecidas, a avaliar em presença de múltiplos critérios. Os problemas multiobjectivo referem-se aos casos em que as alternativas são definidas implicitamente por um conjunto de restrições e os critérios são operacionalizados através de funções a

## 3.2 O Problema de Caminho Mais Curto Multiobjectivo

---

optimizar.

Nos problemas indicados anteriormente, o primeiro corresponde a um problema multiatributo, pois as alternativas a considerar são explicitamente conhecidas à partida e o segundo problema corresponde a um problema multiobjectivo, uma vez que as alternativas possíveis são implicitamente definidas por um conjunto de restrições matemáticas.

Relativamente a trabalhos efectuados neste tipo de problemas, Carlos Barrico [1], fez uma abordagem ao Problema de Caminho Mais Curto Multiobjectivo, em particular à sua aplicação ao Problema de Encaminhamento em Redes Integradas de Comunicações.

Guerriero e Musmanno [34], trataram o problema do caminho mais curto multicritério apresentando uma classe de métodos de fixação de etiquetas para gerar o conjunto de todos os caminhos não dominados desde a origem a todos os outros nós.

Crina Grosan e Ajith Abraham [4], trataram o problema do caminho mais curto com restrições de capacidade, onde o fluxo a enviar da origem ao destino está sujeito a restrições de custos. Trabalharam com duas funções objectivo: minimização do custo do caminho da origem para o destino e minimização do número de arestas comuns cuja capacidade é inferior ao exigido. As restrições do problema são consideradas como objectivos.

## O Problema Multiobjectivo

É comum existirem diversos critérios a serem utilizados na avaliação de uma solução para um problema combinatório, o que torna o problema multiobjectivo ou multicritério. Em particular, nos problemas de optimização quando o espaço de decisões é composto por variáveis discretas, fica caracterizado o seguinte problema de optimização

combinatória multiobjectivo:

$$\begin{aligned}\max z_1 &= f_1(x) = c_1x \\ \max z_2 &= f_2(x) = c_2x \\ &\vdots \\ \max z_h &= f_h(x) = c_hx\end{aligned}$$

sujeito a:

$$x \in X = \{Ax = b, x \geq 0, x \in \mathbb{Z}^n\}$$

onde  $h$  representa o número de funções objectivo (critérios),  $n$  é o número de variáveis do problema (de decisão),  $m$  é o número de restrições do problema,  $X$  é a região admissível no espaço das decisões (ou das variáveis),  $x$  é o vector das variáveis do problema,  $c_j$  corresponde ao vector dos coeficientes da função objectivo  $f_j$ , com  $j = 1, \dots, h$ ,  $A$  é a matriz dos coeficientes com  $m$  linhas e  $n$  colunas e  $b$  é o vector dos termos independentes (recursos disponíveis).

A região admissível no espaço das funções objectivo ou espaço de decisões (o conjunto de todas as imagens dos pontos em  $X$ ), pode ser definida da seguinte forma:

$$Z = \{z \in \mathbb{Z}^h : z = (f_1(x), f_2(x), \dots, f_h(x)), x \in X\}.$$

Na resolução de problemas com apenas um objectivo, procura-se encontrar a solução óptima, ou seja, a solução admissível que optimize a função objectivo, cujo valor é único, mesmo que existam soluções óptimas alternativas. Em problemas com múltiplos objectivos, esse conceito não é aplicável, uma vez que uma solução admissível que optimize um dos objectivos, não optimiza, em geral, os restantes objectivos, quando estes estão em conflito.

Na resolução de problemas multiobjectivo, pretende-se encontrar uma “melhor” solução de compromisso que possa constituir uma solução final do problema de decisão.

## 3.2 O Problema de Caminho Mais Curto Multiobjective

---

Assim, a noção de solução óptima é substituída pela noção de solução não-dominada ou solução óptima de Pareto.

Uma solução é não dominada, ou óptima de Pareto, se começando desse ponto, o valor de qualquer função objectivo não pode ser melhorado sem piorar pelo menos um dos valores das restantes função objectivo.

### Formulação do PCMC Multiobjective

Considere-se um grafo  $G = (N, A)$  e considere-se  $C = (c^1, c^2, \dots, c^h)$  com  $h \geq 1$ , que corresponde aos valores numéricos não negativos associados a cada arco. Ou seja, ao arco  $(i, j)$  estão associados os valores  $c_{ij} = (c_{ij}^1, \dots, c_{ij}^h)$ . Sejam  $s$  e  $t$  dois nós da rede, respectivamente os nós inicial e final. Os valores numéricos associados ao caminho  $p$  de  $s$  para  $t$  na rede  $G$  formam um  $h$ -uplo de somas, em que cada uma delas corresponde à soma dos valores associados aos arcos que pertencem àquele caminho. Ou seja,

$$c(p) = (c^1(p), c^2(p), \dots, c^h(p))$$

em que

$$c^w(p) = \sum_{(i,j) \in p} c_{ij}^w, \quad w \in \{1, 2, \dots, h\}.$$

O problema do caminho mais curto com vários objectivos ( $h$ ) entre os nós  $s$  e  $t$  do grafo  $G$ , pode ser formulado da seguinte forma:

$$\min \quad Z = (Z_1, \dots, Z_h) \tag{3.7}$$

sujeito a:

$$(3.2) - (3.5)$$

onde,

$$Z_1 = \sum_{(i,j) \in A} c_{ij}^1 x_{ij}$$

$$\dots$$

$$Z_h = \sum_{(i,j) \in A} c_{ij}^h x_{ij}$$

e

$$x_{ij} = \begin{cases} 1, & \text{se o arco } (i,j) \text{ pertence ao caminho;} \\ 0, & \text{caso contrário.} \end{cases} \quad \forall (i,j) \in A$$

### 3.3 O Problema de Caminho mais Curto com Restrições Adicionais

Na prática, quando se pretende determinar o CMC entre dois nós, é usual existirem restrições adicionais relativas ao caminho a determinar, nomeadamente, existirem janelas temporais relativas às visitas aos nós, isto é, o nó só pode ser visitado num determinado intervalo de tempo ou existirem restrições relativas ao consumo de um recurso, neste caso assume-se que percorrer um arco implica o consumo de um recurso escasso, pelo que o caminho a ser escolhido deve garantir que o consumo do recurso não ultrapassa a capacidade máxima desse recurso. Os PCMC onde existam recursos escassos associados aos nós, facilmente podem ser convertidos em PCMC onde exista um consumo de recurso associado à utilização nos arcos.

Por vezes, também se pretende determinar o caminho mais curto entre os nós de um grafo que passe obrigatoriamente por determinados nós. Este problema é conhecido com *Vertex Constrained Shortest Path*, [31]. No caso de se pretender que ao determinar o caminho mais curto entre os nós origem e destino se passe por todos os nós da rede, o problema transforma-se no problema do caixeiro viajante.

Existe uma grande variedade de problemas de CMC com restrições adicionais e a terminologia associada a esse tipo de problemas varia muito consoante os autores.

### 3.3 O Problema de Caminho mais Curto com Restrições Adicionais

---

Nesta tese, são considerados dois tipos de problemas do CMC com restrições adicionais: os problemas de CMC com janelas temporais em que se impõem janelas temporais nos nós, obrigando a que se o nó é visitado então essa visita deve decorrer dentro da janela temporal respectiva; e os problemas de CMC com restrições de recurso, restrições essas que não são mais do que restrições do tipo saco-mochila que limitam o consumo de recursos ao longo do caminho.

#### 3.3.1 O Problema do Caminho mais Curto com Janelas Temporais

O Problema do Caminho Mais Curto com Janelas Temporais (PCMCJT), também conhecido por, *Shortest Path Problem with Time Window* consiste em encontrar o caminho com menor custo entre dois nós de um grafo, cujos respectivos nós só podem ser visitados num determinado intervalo de tempo. Este problema surge frequentemente como uma relaxação do problema do Caixeiro Viajante com janelas temporais obtido por relaxação da restrição que obriga a que todos os nós têm que ser visitados e apenas uma vez [22].

Seja  $G = (N, A)$  um grafo com nó origem  $s$  e nó destino  $t$ . Um intervalo  $[a_i, b_i]$ , com  $i \in N$ , é associado a cada nó do grafo, onde  $a_i$  representa a hora a que o cliente  $i$  pode começar a ser servido e  $b_i$  representa a hora a que o cliente  $i$  deixa de poder ser servido, isto é, o cliente  $i$  só pode ser servido no intervalo de tempo  $[a_i, b_i]$ . Uma entrega pode chegar ao cliente antes do início da sua janela temporal e esperar até que seja possível efectuar o serviço (hora de início da janela temporal desse cliente), mas não pode chegar depois do fim da janela temporal.

A cada arco  $(i, j) \in A$  está associado um custo,  $c_{ij}$ , que pode ser positivo ou negativo e um tempo  $t_{ij}$  positivo. Assume-se que o tempo para servir  $i$  está incluído em  $t_{ij}$ ,

para todos os nós  $i \in N$ . Note-se que os arcos  $(i, j) \in A$  têm que respeitar a condição  $a_i + t_{ij} \leq b_j$ . Portanto  $t_{ij}$  representará, por exemplo, o tempo de deslocação do nó  $i$  para o nó  $j$  mais o tempo de servir o cliente no nó  $i$ .

Para formular o problema como um PLI, usam-se variáveis  $s_i$  que representam a hora a que o cliente  $i$  começa a ser servido. Matematicamente, tem-se:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (3.8)$$

sujeito a:

$$(3.2) \quad - \quad (3.4)$$

$$x_{ij} \geq 0, \quad \forall (i, j) \in A \quad (3.9)$$

$$x_{ij}(s_i + t_{ij} - s_j) \leq 0, \quad \forall (i, j) \in A, \quad (3.10)$$

$$a_i \leq s_i \leq b_i, \quad \forall i \in N \quad (3.11)$$

A função objectivo (3.8) minimiza o total de custos. As restrições (3.2)-(3.4), (3.9) são restrições de caminho mais curto entre  $s$  e  $t$ . A restrição (3.10) estabelece a relação entre a hora do início da visita a um nó com a hora de início da visita ao nó seguinte. Só é relevante quando  $x_{ij} = 1$ , isto é, o nó  $j$  é visitado imediatamente a seguir ao nó  $i$ . Nesse caso, a restrição fica  $s_i + t_{ij} \leq s_j$ , isto é, o nó  $j$  só pode ser servido  $t_{ij}$  unidades de tempo depois de ter começado a servir o nó  $i$ .

A restrição (3.11) garante que todas as janelas temporais são respeitadas.

Note-se que as restrições (3.10) não são lineares mas podem ser linearizadas substituindo (3.10) por  $s_i + t_{ij} - M(1 - x_{ij}) \leq s_j$ , onde  $M$  é um valor elevado. Se  $x_{ij} = 0$  então  $s_i + t_{ij} - M \leq s_j$ . Se  $x_{ij} = 1$  então  $s_i + t_{ij} \leq s_j$ , de onde resulta (3.10).

Por vezes, quando existe mais do que uma janela temporal associada a cada nó, certos autores definem o problema como um problema do tipo “*Shortest Path Problem with Resource Window*”. As janelas temporais podem ser consideradas apenas em

### 3.3 O Problema de Caminho mais Curto com Restrições Adicionais

---

termos de penalização na função objectivo e não como restrições do problema. Neste caso é usual atribuir uma penalização quando a janela temporal num nó não é satisfeita.

#### 3.3.2 O Problema do Caminho mais Curto com Restrições de Recurso

Uma outra variante do problema de caminho mais curto é quando há recursos limitados. Desta forma, as limitações de recursos são colocadas como restrições. Este é chamado de Problema de Caminho Mais Curto com Restrições de Recurso (PCMCR), também conhecido por *Resource Constrained Shortest Path Problem*, ver [51], [39] e [45].

De seguida são apresentados alguns tipos de PCMCR.

Geralmente, associado a cada arco de um grafo existe um peso  $w_{ij}$  que representa o consumo de um determinado recurso, que pode ser tempo, dinheiro, combustível, etc. A esse recurso é usualmente associada uma restrição que limita o seu consumo a uma capacidade máxima,  $w$ . Um caminho  $p$  entre o nó origem  $s$  e o nó destino  $t$  diz-se admissível se verifica a restrição de recurso

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq w \quad (3.12)$$

Este problema do CMC com apenas uma restrição do tipo saco-mochila é designado por certos autores como *Weight Constrained Shortest Path Problem*, [30] e [50], ou simplesmente *Constrained Shortest Path Problem*, [39]. Sendo o problema saco-mochila NP-difícil, e como o mesmo pode ser reduzido a um PCMCR, então o PCMCR também é NP-difícil, mesmo considerando o caso de grafos acíclicos com todos os pesos e custos positivos. No entanto, são conhecidos algoritmos pseudo-polinomiais para resolver o problema [8].

Existem também outras formas de tratar este problema que são satisfatórias para



algumas situações reais. Por exemplo usando algoritmos de programação dinâmica.

O PCMCRR está também relacionado com o problema de caminho mais curto multi-objectivo onde são procuradas todas as soluções óptimas de pareto. Para problemas do tipo PCMCRR com uma restrição de recurso, existe uma solução óptima que corresponde a uma solução óptima de pareto dos problemas de caminho mais curto bi-objectivo (têm apenas duas funções objectivo), minimizando os custos e os pesos. Assim, os métodos que geram as soluções óptimas de pareto para os problemas bi-objectivo também geram soluções óptimas para o problema PCMCRR com uma única restrição de recurso. Embora para resolver este tipo de problemas esteja longe de ser um método muito eficiente, alguns métodos desenvolvidos inicialmente para o problema multiobjectivo tornaram-se úteis neste caso [48].

No caso de existir apenas uma restrição em que os pesos são iguais a 1 para todos os arcos, passa a existir um limite no número de arcos que o caminho pode ter e está-se perante um Problema de Caminho mais Curto com Restrições de Cardinalidade (PCMCR), onde a restrição (3.12) pode ser escrita como:

$$\sum_{(i,j) \in A} x_{ij} \leq w \quad (3.13)$$

Esta restrição limita o número de arcos que o caminho seleccionado pode ter.

O PCMCR pode ser resolvido em tempo polinomial. Note-se que os algoritmos de programação dinâmica usuais para o caso geral, são polinomiais para este caso particular.

Considere-se agora o caso mais geral de PCMCR.

Seja  $G = (N, A)$ , um grafo onde cada arco  $(i, j) \in A$  possui um custo, que pode ser positivo ou negativo. Seja  $R = \{r_1, \dots, r_R\}$  o conjunto de  $R$  recursos distintos, cada um deles quantificado por  $w_k$  e seja  $w_{ij}$  a quantidade do recurso  $r$  consumido por percorrer

### 3.3 O Problema de Caminho mais Curto com Restrições Adicionais

---

o arco  $(i, j)$ .

Seja  $P$  o caminho orientado entre os nós origem,  $s$  e destino,  $t$ . O consumo do recurso  $k$  ao longo do caminho  $P$  é dado por:

$$\sum_{(i,j) \in P} w_{ij}^k, \quad k = 1, \dots, R$$

O caminho mais curto com restrições de recurso consiste em encontrar um caminho  $P$  minimizando o custo do caminho e satisfazendo as capacidades máximas disponíveis dos vários recursos.

No caso de não existirem ciclos de comprimento negativo, a formulação em PLI para o problema do CMCR, é a seguinte:

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

sujeito a:

$$(3.2) \quad - \quad (3.4)$$

$$\sum_{(i,j) \in A} w_{ij}^k x_{ij} \leq w_k, \quad k = 1, \dots, R, \quad (3.14)$$

onde as equações (3.2)-(3.4) representam as restrições do caminho mais curto entre  $s$  e  $t$  e as desigualdades (3.14) representam restrições do tipo “saco-mochila” que garantem que o consumo de cada recurso  $k$  não excede o total  $w_k$ .

O PCMCRR pode ainda ser relacionado com o PCMCJT. Se  $R$  é o conjunto dos recursos admissíveis para o caminho a determinar, pode-se associar um tempo de viagem  $t_{ij}^k$  que corresponde ao consumo de  $w_{ij}^k$  unidades de recurso em cada ligação  $(i, j) \in A$ , para  $k \in R$ . A cada nó fazem-se corresponder  $|R|$  restrições de intervalo de tempo  $[a_i^k, b_i^k]$ , uma para cada recurso  $k \in R$ . Para cada  $k \in R$ , e para cada  $i \in N$ ,  $a_i^k$  corresponde ao valor do caminho mais curto de  $s$  até  $i$ , onde os custos são os consumos do recurso  $k$  ( $w_{ij}^k$ ) e  $b_i^k$  corresponde ao caminho mais longo da origem  $s$  até  $i$ . Para  $i = t$ ,

$b_i^k$  assume o valor  $w^k$ , ou seja a capacidade máxima do recurso  $k$ .

Assim, pode-se escrever a seguinte formulação:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (3.15)$$

sujeito a:

$$(3.2) \quad - \quad (3.4)$$

$$x_{ij}(s_i^k + t_{ij}^k - s_j^k) \leq 0, \quad \forall (i,j) \in A, \quad \forall k \in R, \quad (3.16)$$

$$a_i^k \leq s_i^k \leq b_i^k, \quad \forall (i,j) \in A, \quad \forall k \in R, \quad (3.17)$$

$$x_{ij} \geq 0, \quad \forall (i,j) \in A. \quad (3.18)$$

onde as desigualdades (3.16)-(3.17) garantem as restrições temporais correspondentes às restrições de recurso.

Quando os coeficientes da função objectivo nos PCMCRR e/ou PCMCJT podem assumir valores negativos, as formulações apresentadas não garantem a não existência de ciclos, nesse caso, se se pretender que não ocorram ciclos é necessário incluir as restrições de eliminação de sub-ciclos. Usualmente este tipo de problemas em que se impõe a não ocorrência de ciclos é conhecido por *Elementary Shortest Path Problem*.

### 3.3.3 Programação Dinâmica

Uma parte significativa dos trabalhos sobre PCMCRA têm-se baseado em técnicas de programação dinâmica.

A programação dinâmica é uma técnica bastante poderosa para resolver determinados tipos de problemas computacionais. Esta técnica consiste em subdividir o problema em causa em sub-problemas menores, mais fáceis de serem resolvidos. O paradigma de programação dinâmica segue os seguintes passos: remover um elemento do problema; resolver o problema menor; e usar a solução do problema menor para adicionar

### 3.3 O Problema de Caminho mais Curto com Restrições Adicionais

---

o elemento removido de maneira adequada, produzindo uma solução para o problema maior.

Uma das características da programação dinâmica é que permite obter a solução óptima do problema, isto é, é um algoritmo exacto.

De seguida exemplifica-se um algoritmo para o PCMC com uma única restrição de recurso.

O PCMCRR definido num grafo  $G = (N, A)$  onde  $N = \{1, \dots, n\}$ , pode ser formulado com recurso à programação dinâmica, uma vez que um caminho óptimo do nó 1 para o nó  $n$  é construído a partir de sub-caminhos óptimos.

Seja  $c_j(r)$  o custo do menor caminho com peso menor ou igual a  $r$  do nó origem para o nó  $j$ .

Fazendo  $c_1(r) = 0$  para  $0 \leq r \leq w$  e  $c_j(0) = \infty$ ,  $\forall j = 2, \dots, n$  obtém-se a seguinte equação recursiva:

$$c_j(r) = \min \left\{ c_j(r-1), \min_{(i,j) \in A, w_{ij} \leq r} \{c_i(r-w_{ij}) + c_{ij}\} \right\}$$

onde  $c_{ij}$  corresponde ao custo para ir do vértice  $i$  ao vértice  $j$  e  $w_{ij}$  corresponde ao consumo de recursos para ir vértice  $i$  ao vértice  $j$ . Este algoritmo tem complexidade  $O(n w)$ .

Esta técnica também pode ser generalizada para problemas com mais restrições. No entanto, o método da programação dinâmica para o PCMCRR gera um elevado número de estados, pelo que para contornar o problema, muitos métodos desenvolvidos com base na programação dinâmica procuram apenas gerar estados não dominados. Um estado associado a um nó  $i$ ,  $i \in N$  é caracterizado por um par  $(C_i, W_i)$  onde  $C_i$  representa o custo desse caminho e  $W_i$  representa o peso de um caminho do nó origem para o nó  $i$ . Dados dois estados associados ao nó  $i$ :  $(C_i^1, W_i^1)$  e  $(C_i^2, W_i^2)$  diz-se que o estado  $(C_i^1, W_i^1)$  domina o estado  $(C_i^2, W_i^2)$  se  $C_i^1 \leq C_i^2$  e  $W_i^1 \leq W_i^2$ .

### 3. Problemas de Caminho mais Curto

---

De seguida, é considerado um exemplo de modo a ilustrar o algoritmo de programação dinâmica.

Seja  $G = (N, A)$  com os pares  $(c_{ij}, w_{ij})$  associados aos arcos, em que  $c_{ij}$  representa o custo para ir do nó  $i$  para o nó  $j$  e  $w_{ij}$  representa a quantidade de recurso consumida na deslocação do nó  $i$  para o nó  $j$ , (ver figura (3.2)). Considere-se também a quantidade de 9 unidades disponíveis do recurso ( $w = 9$ ).

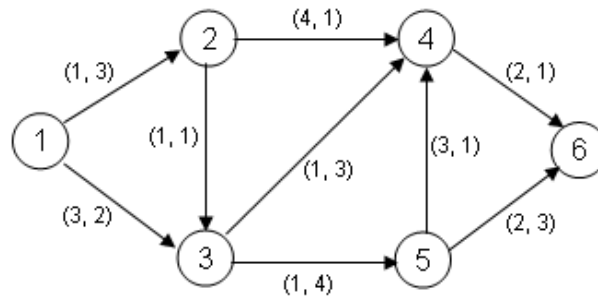


Figura 3.2: Grafo do exemplo

Aplicando o algoritmo anterior ao grafo da figura (3.2) obtém-se a tabela (3.1) que representa os custos dos caminhos de menor custo do nó 1 ao nó  $j$ .

$c_j \backslash r$	0	1	2	3	4	5	6	7	8	9
$c_1$	0	0	0	0	0	0	0	0	0	0
$c_2$	$\infty$	$\infty$	$\infty$	1	1	1	1	1	1	1
$c_3$	$\infty$	$\infty$	3	3	2	2	2	2	2	2
$c_4$	$\infty$	$\infty$	$\infty$	$\infty$	5	4	4	3	3	3
$c_5$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	4	4	3	3
$c_6$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	6	6	6	5	5

Tabela 3.1: Tabela de Custos

### 3.3 O Problema de Caminho mais Curto com Restrições Adicionais

---

Por observação da tabela (3.1), conclui-se que alguns estados podem ser eliminados, por existir um caminho mais curto com um gasto inferior.

Na tabela (3.2) podem ser verificados os estados não dominados correspondentes ao exemplo da figura (3.2).

1	2	3	4	5	6
(0,0)	(1,3)	(3,2)	(5,4)	(4,6)	(6,5)
		(2,4)	(4,5)	(3,8)	(5,8)
			(3,7)		

Tabela 3.2: Tabela de estados não dominados

De modo a gerar apenas os estados não dominados é usual recorrer a técnicas de fixação de etiquetas em que para cada nó só são geradas as etiquetas associadas aos estados não dominados. Note-se que o número de estados não dominados cresce rapidamente. Para contornar este problema certos autores propõem uma pesquisa bi-direccional (do nó origem para o nó destino e do nó destino para o nó origem) de modo a controlar este aumento do número de etiquetas. A meio da pesquisa será necessário combinar as listas de etiquetas geradas pelas duas pesquisas [27]. Outra forma de redução do número de etiquetas consiste em usar limites superiores e inferiores que permitam excluir estados a partir dos quais não é possível obter a solução óptima. Outros autores procuram proceder a uma redução do problema usando pré-processamento, ver [20] e [30].

#### 3.3.4 Revisão da Literatura

Nesta secção, é apresentada uma breve revisão da literatura sobre métodos de resolução para o problema do caminho mais curto com restrições adicionais.

Desde que o problema do caminho mais curto com restrições adicionais tem sido usado em diversos tipos de problemas, estes têm sido estudados mais exhaustivamente e têm surgido muitos trabalhos que utilizam vários métodos de resolução ou mesmo combinam alguns deles.

Como a literatura existente é muito vasta, de seguida é efectuado um resumo de alguns trabalhos efectuados não se pretendendo que seja uma apresentação exhaustiva de todos os trabalhos realizados na área.

Desde muito cedo que se encontram artigos sobre a aplicação do método de programação dinâmica a problemas de caminho mais curto.

Salienta-se o trabalho de Joksch [43] que aplicou o método a problemas do CMC com um único recurso e considerando apenas custos positivos.

Hassin [42] propôs dois algoritmos pseudo-polinomiais exactos em grafos acíclicos.

Aneja et al. [38] apresentaram um algoritmo baseado em programação dinâmica (fixação de etiquetas) para o problema de caminho mais curto com várias restrições.

Desrochers [2], desenvolveu num trabalho pioneiro técnicas de Programação Dinâmica para o PCMCJT.

Entretanto, foram sendo desenvolvidos uma grande variedade de algoritmos baseados em programação dinâmica usando métodos de fixação de etiquetas, nomeadamente Desrochers e Soumis em [11] para o PCMCJT.

Ioachim et al. [35] consideraram o PCMC com janelas temporais e com custos lineares nos tempos de início da visita a cada nó  $i$ . Apresentaram um algoritmo de programação dinâmica para o caso em que o grafo é acíclico.

### 3.3 O Problema de Caminho mais Curto com Restrições Adicionais

---

Desrosiers et al. [39], recorrendo também a métodos de programação dinâmica, modelaram problemas de rotas de veículos com restrições de tempo e de capacidade.

Jaumard et al. [10], descreveram um algoritmo baseado em programação dinâmica para o PCMC com janelas temporais nos arcos.

Feillet et al. [32] propuseram um algoritmo de programação dinâmica exacto, adaptado do algoritmo de Desrochers [2], e usaram-no para resolver o PCMCJT através de geração de colunas.

Irnich e Desaulniers [13], trabalharam o mesmo método para o caso geral do PCMCRA tratando vários tipos de restrições.

Boland et al. [20] propuseram um algoritmo de programação dinâmica baseado em fixação de etiquetas para resolver o problema de caminho mais curto com restrições de recurso e custos arbitrários em grafos gerais usando restrições nos nós de modo a proibir a repetição dos nós no caminho.

Recentemente, Faramroze, Nemhauser e Savelsbergh [14], apresentaram um algoritmo baseado em vários melhoramentos dos algoritmos de Programação Dinâmica para resolver o PCMCRR decorrente de problemas de geração de colunas para formulações que envolvem redes de grandes dimensões e um grande número de restrições.

Relativamente aos métodos de resolução usando relaxações, são destacados os seguintes trabalhos.

Handler e Zang [15] usaram métodos baseados em relaxações Lagrangeanas, para otimizar a solução do problema do caminho mais curto com uma única restrição de recurso e assumindo pesos não negativos nos arcos.

Beasley e Christofides [31] recorreram à relaxação lagrangeana de modo a obter limites inferiores e usaram o procedimento Branch and Bound para otimizar o problema do caminho mais curto com restrições de recurso.

Mehlhorn e Ziegelmann [28], propuseram um algoritmo combinatório para resolver a relaxação linear de problemas de caminho mais curto com múltiplas restrições de recurso. Efectuaram comparações computacionais com outros métodos de resolver a



relaxação linear e apresentaram um algoritmo exacto para o problema.

Mesmo com métodos sofisticados para procura de soluções, resolver um PCMCRA pode gerar tempos de execução muito demorados. É onde entram as heurísticas para o problema.

Um estudo sobre heurísticas para o PCMCRA foi realizado por Irnich e Desaulniers onde fizeram a distinção entre as três principais áreas de aplicação de heurísticas para: Pré-processamento, programação dinâmica e busca directa, ver [13].

Mais referências sobre métodos de resolução para PCMCRA podem ser encontradas em [13].

### 3.3 O Problema de Caminho mais Curto com Restrições Adicionais

---

# Capítulo 4

## Aplicações: Modelação de Problemas Reais

O PCMC sem restrições adicionais tem inúmeras aplicações reais. Alguns exemplos dessas aplicações podem ser consultados em [37]. Neste capítulo são abordadas apenas aplicações modeladas como PCMCRA.

### 4.1 Planeamento de caminhos para Gestão de Frotas

No planeamento e controlo de frotas de veículos, ver por exemplo [21], [25], normalmente é necessário encontrar a melhor rota (o melhor caminho) tendo em conta a missão da frota, a tipologia da carga ou o tipo de utilizadores. A missão da frota pode estar relacionada com o transporte de cargas, transporte humano, serviços de emergência (ambulâncias e bombeiros), etc. No entanto, também pode ser considerado que a carga pode ser valiosa, pesada, volumosa, perigosa ou tóxica, enquanto que os passageiros podem ser clientes de táxi, turistas, crianças, idosos ou deficientes.

Conhecendo a origem e o destino da rota, é necessário avaliar um série de características, para encontrar o melhor caminho entre a origem e o destino, e normalmente estas são expressas por valores numéricos. Assim, para a determinação da rota mais

## 4.1 Planeamento de caminhos para Gestão de Frotas

---

favorável deve-se ter em conta, não só as características do tráfico mas também as características das vias.

Como será visto, este problema pode ser formulado como um problema do caminho mais curto restrições de recursos, onde os recursos estão relacionados com os atributos das estradas (ligações).

### Estudo dos atributos das ligações

No estudo de problemas reais, como por exemplo no estudo de caminhos mais curtos em redes de estradas, um caminho pode ser considerado admissível se apenas forem satisfeitas algumas características. Estas características podem ser distâncias, tempo, custos, etc. Mas, um caminho que é óptimo em relação a determinada característica, pode não ser praticável em relação às outras características que também são importantes na resolução do problema em geral. Assim, há a necessidade de associar vários atributos a cada ligação e classificar os vários tipos de atributos. Os autores [25], sugerem a seguinte distribuição:

1. Atributos quantitativos totalizáveis;
2. Atributos quantitativos não totalizáveis;
3. Atributos qualitativos.

Atributos tais como a distância percorrida, tempo de viagem, custos de combustível ou portagens, podem ser somados ao longo do percurso e contabilizados num todo. São os *atributos quantitativos totalizáveis*.

Outros atributos tais como a inclinação e largura da estrada, números de semáforos, número de rotundas ou número de faixas de rodagem, já não podem ser considerados como uma soma total ao longo do percurso, pois por exemplo a largura da estrada pode ser de 8 metros numa determinada estrada e 12 metros numa auto estrada. Ora

estes valores terão que ser considerados como uma média a calcular para cada restrição deste tipo. São os *atributos quantitativos não totalizáveis*.

Os atributos restantes, tais como o tipo de estrada (nacional, IP, auto-estrada, etc.), o estado de conservação da estrada, o grau de probabilidade de encontrar filas de trânsito, condicionamentos devidos a obras e acidentes, também não podem ser considerados como a soma total ao longo do percurso. São os atributos expressos por índices - *atributos qualitativos*.

Como foi referido na secção (3.2), quando são associados muitos atributos às ligações do percurso, os Problemas de Caminho mais Curto, são geralmente tratados como Caminhos mais Curtos com múltiplos objectivos. Em alternativa, se for apenas associado um atributo, é considerado um Problema de Caminho mais Curto com um só objectivo e os outros atributos são interpretados como recursos num conjunto de restrições adicionais ao problema.

Assim, neste contexto, se apenas forem considerados atributos quantitativos totalizáveis, está-se perante um PCMCRR. Se forem também considerados atributos quantitativos não totalizáveis e qualitativos, será mais complexo e está-se perante um problema mais geral de caminho mais curto com restrições adicionais que se podem reduzir a restrições de recurso. Neste caso mais geral, os coeficientes das variáveis nas restrições de recurso podem ter valores arbitrários (não necessariamente positivos).

### **Caminho mais Curto com Restrições Gerais de Recurso**

O Problema do caminho mais curto com restrições gerais de recurso, conhecido por, *The General Resource Constrained Shortest Path Problem*, consiste em encontrar o caminho mais curto entre dois nós de uma rede de estradas que satisfaça um conjunto de características (restrições) gerais não necessariamente numéricas e possíveis de totali-

## 4.1 Planeamento de caminhos para Gestão de Frotas

---

zar (restrições do tipo 2 e 3 já enumeradas).

De seguida é explicado como descrever os vários tipos de restrições através de restrições lineares.

Seja  $G = (N, A)$ , um grafo onde  $N$  é o conjunto dos nós e  $A$  é o conjunto de arcos,  $c_{ij}$  o custo associado a cada ligação  $(i, j) \in A$  e seja  $x_{ij}$  a variável associada a cada ligação  $(i, j) \in A$ . Se a ligação  $(i, j) \in P$  então  $x_{ij} = 1$ , caso contrário,  $x_{ij} = 0$ .

### Restrições relativas a atributos quantitativos não totalizáveis

Seguindo [25], é explicado como formular as restrições relativas a atributos quantitativos não totalizáveis. Todas elas se podem formular de uma ou de duas maneiras, por eliminação de arcos (que não obedecem a alguma restrição) ou através de restrições do tipo saco-mochila com coeficientes arbitrários (podem ser negativos). É possível também definir várias restrições em relação a um mesmo atributo, por exemplo, a largura da estrada. Pode-se excluir troços em que a largura da estrada é inferior a  $x$  metros e ainda se pode impor que mais de 90% do percurso seja efectuado em estradas com largura superior a  $y$  metros.

Em [25] são considerados as seguintes condições para que um caminho seja considerado admissível:

- (i) Um caminho admissível não pode conter ligações cujo valor do parâmetro seja menor (ou maior) que um valor fixo  $U$ ;
- (ii) A média dos valores do parâmetro associado a um caminho admissível tem que ser maior (ou menor) que  $U$ ;
- (iii) Um caminho admissível tem pelo menos  $X$  quilómetros de estrada (ou pelo menos um segmento de estrada) cujo parâmetro é menor (ou maior) que  $U$ ;

- (iv) Um caminho é admissível se pelo menos  $p\%$  da sua extensão é constituída por ligações cujo valor do parâmetro é maior (ou menor) que  $U$ , por exemplo, um caminho é admissível se 90% do percurso tiver pelo menos 2 faixas.

O primeiro tipo de condição (restrição) é simples de garantir, basta eliminar do grafo as ligações que não são possíveis de realizar para determinado atributo e calcular o caminho mais curto para o grafo restante.

A segunda condição pode ser formulada como:

$$\frac{\sum_{(i,j) \in A} u_{ij} l_{ij} x_{ij}}{\sum_{(i,j) \in A} l_{ij} x_{ij}} \leq U$$

onde  $u_{ij}$  é o valor do parâmetro a satisfazer na ligação  $(i, j)$ ,  $l_{ij}$  é o comprimento de  $(i, j)$  e  $U$  é o valor máximo que o valor médio do parâmetro a considerar pode assumir.

Esta restrição ainda pode ser escrita de uma forma linear como:

$$\sum_{(i,j) \in A} (u_{ij} - U) l_{ij} x_{ij} \leq 0.$$

Deste modo, os coeficientes das variáveis  $x_{ij}$  podem ser positivos, negativos ou nulos.

No terceiro e quarto casos, a formulação linear é mais difícil. Uma maneira de o fazer é substituir o atributo original por um atributo binário  $b_{ij}$ , que é 1 se o arco  $(i, j)$  satisfaz a condição do parâmetro (tem que ser maior que  $U$ ) e 0 no caso contrário.

Assim, usando o novo atributo, a terceira condição pode ser formulado como:

$$\sum_{(i,j) \in A} b_{ij} t_{ij} x_{ij} \geq T,$$

onde  $t_{ij}$  é o tempo de viagem para percorrer o arco  $(i, j)$  e  $T$  é o valor mínimo do total do tempo da viagem para todas as ligações do caminho onde  $b_{ij}$  é igual a 1.

## 4.1 Planeamento de caminhos para Gestão de Frotas

---

Usando o atributo  $b_{ij}$  na quarta condição, obtém-se a seguinte formulação:

$$\sum_{(i,j) \in A} b_{ij} l_{ij} x_{ij} \geq \frac{p}{100} \sum_{(i,j) \in A} l_{ij} x_{ij},$$

onde  $p$  é a percentagem mínima do comprimento do caminho com as características consideradas para a solução.

Esta condição ainda pode ser reescrita como:

$$\sum_{(i,j) \in A} \left( b_{ij} - \frac{p}{100} \right) l_{ij} x_{ij} \geq 0.$$

Novamente, os coeficientes de  $x_{ij}$  podem ser positivos ou negativos.

### Restrições Relativas a Atributos Qualitativos

Seguindo os autores [25], as restrições relacionadas com atributos do tipo 3 (qualitativas)  $d_{ij}$  são muito similares às anteriores. Por exemplo, pode-se impor que 50% do percurso seja efectuado em auto-estrada.

Neste tipo de restrições, os autores propõem as seguintes condições para um caminho ser admissível:

- (i) Um caminho admissível pode conter ligações cujo valor do parâmetro é igual a  $D$  (ou maior que  $D$ );
- (ii) Um caminho admissível tem pelo menos  $X$  quilómetros de estrada (ou pelo menos um segmento de estrada com tempo de viagem superior a  $T$ ) cujo parâmetro é menor (ou maior) que  $D$ ;
- (iii) Um caminho é admissível se pelo menos  $p\%$  da sua extensão é constituída por ligações cujo valor do parâmetro é igual (maior ou menor) que  $D$ , por exemplo, um caminho é admissível se 80% do percurso for realizado em auto-estrada.

Para a primeira condição, procede-se de igual modo como no caso anterior, eliminando os caminhos que não são admissíveis.



Nas segundas e terceiras condições, pode-se adoptar o procedimento dos atributos anteriores. Neste caso, substituindo o atributo original da ligação  $(i, j)$ ,  $d_{ij}$ , por um atributo binário,  $b_{ij}$ , cujo valor é 1 se o arco  $(i, j)$  satisfaz a condição ( $d_{ij} = D$  ou  $d_{ij} < D$  ou  $d_{ij} > D$ ) e 0 no caso contrário.

Usando o novo atributo, a segunda condição pode ser formulada como:

$$\sum_{(i,j) \in A} b_{ij} t_{ij} x_{ij} \geq T$$

e a terceira condição por:

$$\sum_{(i,j) \in A} \left( b_{ij} - \frac{p}{100} \right) l_{ij} x_{ij} \geq 0$$

.

Deste modo, pode-se verificar que todas as condições definidas podem ser modeladas como restrições do tipo (3.12). Como os coeficientes das variáveis são arbitrários é necessário adicionar as restrições de eliminação de sub-ciclos:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset N : |S| \geq 2.$$

Note-se que a generalidade dos algoritmos para o caminho mais curto com restrições de recurso assumem que os coeficientes das restrições de recurso são não negativos.

## 4.2 Redes de Comunicação

Em redes de telecomunicações, os problemas de CMCRRA ocorrem, por exemplo, associados a problemas de encaminhamento de tráfego sujeitos a determinados requisitos de serviço que garantam uma qualidade de serviço pretendida.

Esta secção é baseada essencialmente em Amaro Sousa, [47].

Durante o tempo de vida de uma rede de telecomunicações, a quantidade de tráfego que a rede precisa de suportar varia quer pelas características dos serviços oferecidos, quer pela evolução do número e localização de clientes, pelo que de tempos a tempos, a rede precisa de ser redimensionada por forma a continuar a suportar todo o tráfego mantendo os custos o mais baixo possível e os mesmos requisitos de serviço.

A gestão de uma rede de telecomunicações baseia-se essencialmente em duas fases: uma fase de investimento em que o operador dimensiona os recursos da rede para suportar uma determinada quantidade de tráfego estimada num horizonte temporal previsível, e uma fase de operação da rede em que os recursos se mantêm inalterados e o operador gere-os de forma a cumprir com os requisitos dos clientes da melhor forma possível.

Os problemas de optimização relativos à primeira fase são chamados de problemas de dimensionamento em que se pretende minimizar o custo da rede resultante que cumpra com os requisitos dos serviços proporcionados. Os problemas de optimização relativos à segunda fase são problemas de engenharia de tráfego em que se pretende optimizar o desempenho da rede face aos requisitos dos serviços proporcionados.

### Problemas de dimensionamento de Redes

Seja  $G = (N, A)$  o grafo que define uma rede de telecomunicações, onde  $N$  é o conjunto dos nós da rede e  $A$  o conjunto das arestas  $\{i, j\}$  em que é possível estabelecer ligações de rede. Em cada aresta  $\{i, j\}$  é possível estabelecer uma ligação com uma capacidade dada por um múltiplo inteiro não negativo de módulos de capacidade individual  $\alpha$  a

um custo por módulo dado por  $c_{\{i,j\}}$ . Seja  $K$  o conjunto de fluxos a suportar pela rede. Para cada  $k \in K$  é definido:

- $s_k$  para o nó de origem,
- $t_k$  para o nó destino,
- $b_k$  para a largura de banda em cada sentido.

Cada aresta  $\{i, j\} \in A$  representa um par de nós entre os quais é possível instalar um conjunto de ligações.

Considere-se agora dois casos de problemas de encaminhamento:

#### Caso A: Encaminhamento sem restrições

O problema de dimensionamento de redes consiste na determinação da topologia física de menor custo sem nenhuma restrição aos percursos de encaminhamento dos pares de tráfego.

O problema pode ser formulado do seguinte modo:

$$\min \sum_{\{i,j\} \in A} c_{\{ij\}} y_{\{ij\}}$$

sujeito a:

$$\sum_{j:\{i,j\} \in A} x_{s_k j}^k = 1, \forall k \in K \quad (4.1)$$

$$\sum_{j:\{i,j\} \in A} x_{ij}^k - \sum_{j:\{i,j\} \in A} x_{ji}^k = 0, \forall i \in N \setminus \{s_k, t_k\}, \forall k \in K \quad (4.2)$$

$$\sum_{j:\{i,j\} \in A} x_{j t_k}^k = 1, \forall k \in K \quad (4.3)$$

$$\sum_{k \in K} b_k x_{ij}^k + \sum_{k \in K} b_k x_{ji}^k \leq \alpha y_{\{ij\}}, \forall (i, j) \in A, \quad (4.4)$$

$$x_{ij}^k \in \{0, 1\}, \forall (i, j) \in A, \forall k \in K \quad (4.5)$$

$$y_{\{ij\}} \in \mathbb{Z}^+ \quad (4.6)$$

## 4.2 Redes de Comunicação

---

em que na função objectivo  $c_{\{ij\}}$  indica o custo de cada ligação entre os nós  $i$  e  $j$  e as variáveis inteiras  $y_{\{ij\}}$  indicam o número de ligações instaladas entre os nós  $i$  e  $j$  na solução final. As variáveis binárias  $x_{ij}^k$  indicam se o percurso de encaminhamento do fluxo  $k$  atravessa o conjunto de ligações entre os nós  $i$  e  $j$  no sentido de  $i$  para  $j$ .

Para cada fluxo  $k \in K$ , o conjunto das restrições (4.1) asseguram que existe um caminho do nó origem  $s_k$  para o nó destino  $t_k$ . O percurso inclui um arco de saída do nó origem através da equação (4.1), inclui um arco de entrada no nó destino através da equação (4.3) e a equação (4.2) garante que todos os outros nós da rede, se existir um arco de entrada, tem também que existir um arco de saída. As restrições (4.4) garantem ainda que a rede tem capacidade suficiente para suportar o tráfego esperado. Em cada aresta  $\{i, j\}$  a capacidade das ligações a instalar é dada por  $\alpha y_{\{ij\}}$ .

### Caso B: Restrições de salto dos percursos de encaminhamento

Para garantir a qualidade do serviço quer relativamente à fiabilidade da rede quer relativamente aos atrasos máximos no envio da informação é usual limitar o número de arcos utilizados no encaminhamento da rede.

Limitar o número de saltos de encaminhamento numa rede influencia a fiabilidade da rede, pois a probabilidade de um fluxo ser afectado com uma falha é menor quanto menor for o número de arcos percorridos, e influencia também o atraso máximo e a variabilidade do atraso em redes com comutação de pacotes (nos nós) quando os atrasos de filas de espera são os mais determinantes.

Seja  $w$  o número máximo de arestas (saltos) de todos os percursos de encaminhamento, esta variante pode ser definida pelo seguinte modelo:

$$\min \sum_{\{i,j\} \in A} c_{\{ij\}} y_{\{ij\}} \quad (4.7)$$

sujeito a:

$$(4.1) \quad - \quad (4.4)$$

$$\sum_{(i,j) \in A} x_{ij}^k \leq w, \quad \forall k \in K \quad (4.8)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in A, \forall k \in K \quad (4.9)$$

$$y_{\{ij\}} \in \mathbb{Z}^+ \quad (4.10)$$

onde a nova restrição (4.8) impõe que o número de arestas usadas em cada percurso de encaminhamento seja no máximo  $w$ .

Note-se que o problema (4.1)-(4.3), (4.8), (4.10) é um PCMCRC.

### 4.3 Outras aplicações

Existem outras aplicações que podem ser modeladas como problemas de caminho mais curto com restrições adicionais, ver por exemplo [29].

Nesta secção será feito um pequeno resumo de algumas aplicações que podem ser modeladas como um PCMCRA.

#### 4.3.1 Aproximação de Curvas

Dahl e Realfsen [6], estudaram o Problema de Aproximação de Curvas Lineares e mostraram como modelar o problema como um problema de caminho mais curto com restrições de cardinalidade. Este problema tem aplicação notável em áreas como a cartografia, computação gráfica e processamento de imagem.

O problema consiste em aproximar uma curva linear por troços por uma nova curva passando por um menor número de pontos intermédios de forma a minimizar o erro da aproximação. Como exemplo, pode-se ver a figura (4.1) em que está representada

### 4.3 Outras aplicações

---

uma função que passa inicialmente por todos os 7 pontos (linha contínua) e uma sua aproximação (linha tracejada) que passa apenas por 4 dos pontos iniciais.

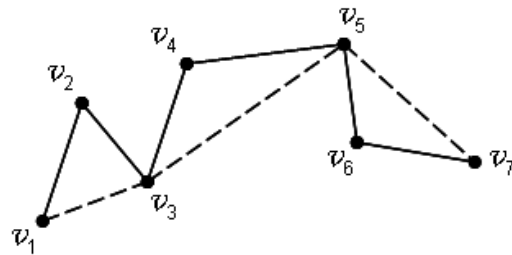


Figura 4.1: Aproximação de curva passando por 7 pontos

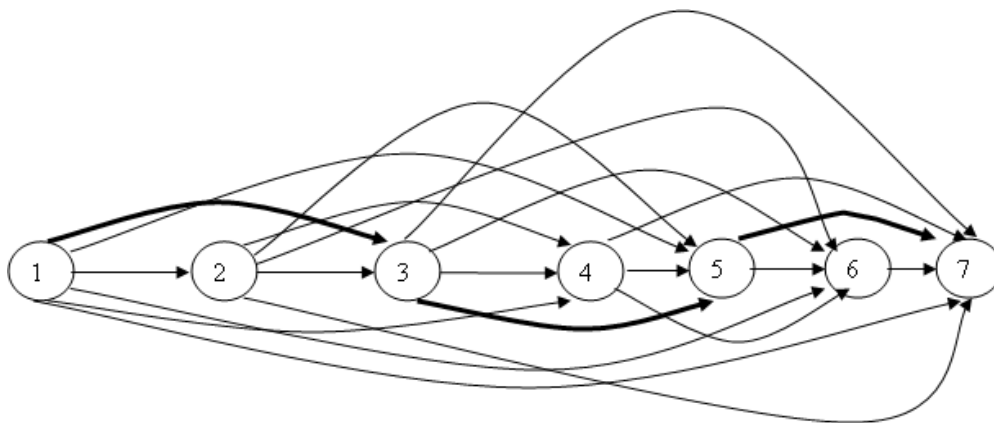


Figura 4.2: Caminho mais curto correspondente à figura (4.1)

Os autores formulam o problema como um PCMC com restrições de cardinalidade, num grafo dirigido onde os nós do grafo correspondem aos nós da curva, a restrição de cardinalidade limita o número de pontos intermédios (nós visitados pelo caminho) aos pontos de quebra da curva linear por troços e existe um arco entre cada par de pontos orientados da esquerda para a direita.

O caminho a tracejado da figura (4.1) corresponde ao caminho 1-3-5-7 (caminho com 3 arcos) no grafo da figura (4.2). O custo associado aos arcos representa uma medida

da diferença entre o valor da função e o valor da aproximação entre os respectivos nós.

### 4.3.2 Modelação de Aplicações de Engenharia

Eliman e Kohler [9], mostraram como modelar duas aplicações de engenharia como um problema de caminho mais curto com restrições de recurso (múltiplos recursos): determinação de sequências óptimas no processo de tratamento de águas residuais e custo mínimo no desenho e construção de estruturas.

#### Determinação de sequências óptimas no tratamento de águas residuais

Este modelo foi usado para determinar sequências óptimas no processo de tratamento de águas residuais num complexo industrial e numa ETAR no *Kuwait*. O projecto do tratamento de águas foi requerido para reduzir o custo dos tratamentos das águas de forma a garantir níveis de concentração de poluentes aceitáveis.

Dado um afluente específico da ETAR, pretende-se determinar a sequência dos processos de tratamento que reduzam a concentração de poluentes para níveis predefinidos, a um menor custo.

As restrições de sequenciamento dos processos de tratamento são representadas implicitamente por um grafo acíclico orientado.

Uma sequência de processos de tratamento é admissível ou realizável se existe um caminho, no grafo, da origem para o destino.

São consideradas restrições adicionais aos caminhos que garantam níveis aceitáveis de concentração de poluentes no final do processo.

#### Desenho e Construção de Estruturas

Este projecto, foi usado para ajudar Engenheiros a conhecer a resistência térmica de construção de estruturas. O problema consiste em desenhar compostos térmicos efici-

#### 4.4 Redução de outros problemas

---

entes para estruturas de paredes e tectos a um menor custo. Pretende-se determinar o material a incluir em cada camada e a sequência de materiais.

#### 4.3.3 Sistema de Gestão de Aviões Militares

Zabarankin et al. [36] apresentaram uma aplicação para um sistema de gestão de aviões militares. O problema descrito em [36] surge no contexto de voos militares em território inimigo onde são conhecidos um grande número de radares do inimigo. O problema consiste em encontrar um caminho que minimize o risco de ser detectado pelos radares ao longo de um percurso e sujeito a algumas restrições, por exemplo, tempo do voo, consumo de combustível ou tamanho do percurso a realizar.

### 4.4 Redução de outros problemas

O problema do Caminho mais curto com Restrições Adicionais (PCMCCRA), surge igualmente como generalização de outros problemas de optimização combinatória. Nesta secção é exemplificado um caso desses, reduzindo o problema da determinação da sub-árvore óptima de uma árvore com restrições de capacidade (PSARC) a um PCMCRR. Esta redução foi originalmente construída por Sílvia em [46].

O PSARC tem várias aplicações, nomeadamente no desenho e expansão de Redes de Acesso Local de Telecomunicações [12], [24] e [23]. As Redes de Acesso Local de Telecomunicações têm, normalmente, uma estrutura em árvore, onde os clientes individuais que são representados pelos nós, estão ligados a uma central de distribuição, a raiz da árvore.

De seguida é descrito o PSARC e é mostrada uma redução do PSARC ao PCMCRR.



---

#### 4. Aplicações: Modelação de Problemas Reais

---

Considere-se uma árvore não orientada  $T = (N, A)$ , onde  $N = \{0, \dots, n\}$  representa o conjunto dos nós da árvore,  $A$  representa o conjuntos das arestas e a cada nó  $v \in N$  está associado uma procura  $d_i$  e um lucro  $c_i$ , indivisíveis. O nó 0 é a raiz da árvore. O PSARC consiste em encontrar uma sub-árvore  $T' = (N', A')$  de  $T$  com raiz no nó 0 tal que a soma dos lucros sobre  $T'$ ,  $\sum_{i \in N'} c_i$  é maximizada e a soma das procuras nos nós sobre  $T'$  não excede a capacidade do saco-mochila  $w$ , isto é,  $\sum_{i \in V'} d_i \leq w$ .

Se seguida é apresentada uma formulação do PSARC como um problema de PLI.

Seja  $p_i$  o antecessor do nó  $i$  no único caminho da raiz para o nó  $i$  (com  $p_0 = 0$ ) e seja  $x_j = \begin{cases} 1, & \text{se } j \in N'; \\ 0, & \text{caso contrário.} \end{cases}$

Assim, o PSARC pode ser formulado do seguinte modo:

$$\max \sum_{j=0}^n c_j x_j$$

sujeito a:

$$x_{p_j} \geq x_j, \quad j = 1, \dots, n \tag{4.11}$$

$$\sum_{j=0}^n d_j x_j \leq w, \tag{4.12}$$

$$x_j \in \{0, 1\}, \quad j = 0, \dots, n \tag{4.13}$$

em que, a restrição (4.11) significa que se um nó é seleccionado então o seu antecessor também tem que ser seleccionado e a restrição (4.12) significa que soma das procuras sobre  $T'$  não excede uma dada capacidade do saco-mochila  $w$ . Note-se que se forem removidas as restrições  $x_{p_j} \geq x_j$ , o PSARC é reduzido ao Problema do Saco-Mochila 0-1.

De seguida, é apresentada uma das reformulações propostas em [46] em que o PSARC é reformulado como um problema do caminho mais curto com restrição de

#### 4.4 Redução de outros problemas

---

recurso onde o objectivo é mostrar que o PSARC é um caso particular do problema do caminho mais curto com restrição de recurso.

Considere-se como exemplo a árvore da figura (4.3) com  $w = 11$ . Os nós são etiquetados por ordem de pesquisa em profundidade da esquerda para a direita (*DFO* - *Depth First Order*), os números dentro dos parêntesis indicam os lucros e as procuras, por esta ordem. Por simplicidade assume-se neste texto que  $d_0 = c_0 = 0$

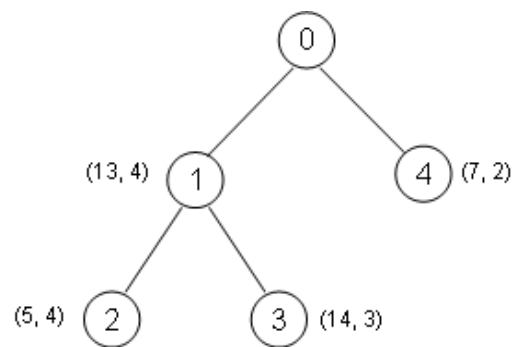


Figura 4.3: Exemplo

Construa-se então o seguinte grafo  $G$  com base nesta árvore, ver figura (4.4).

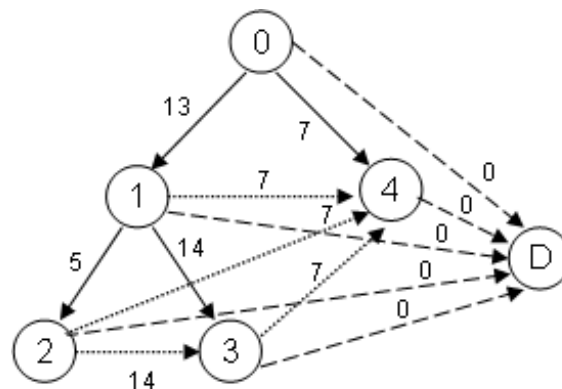


Figura 4.4: Representação do grafo associado à árvore da figura (4.3)

Pretende-se que uma sub-árvore de  $T$  corresponda a um único caminho no novo grafo  $G$ . Neste caminho os nós são percorridos seguindo a ordenação *DFO*. Em  $G$

incluiu-se um novo nó, nó  $D$ , fictício, como nó destino. O novo grafo  $G$  é um grafo orientado cujos nós são os nós do grafo original e mais o nó fictício  $D$  e onde existem três tipos de arcos. Os arcos correspondentes às arestas de  $T$  (linha contínua) com a orientação dada pela ordenação  $DFO$ , os arcos que unem cada nó  $i \in N$  ao nó destino  $D$  (linha tracejada) que se destinam a assegurar que cada nó  $i \in N$  pode ser o último nó, segundo a ordenação  $DFO$ , a ser seleccionado para  $T'$ . Finalmente existe um terceiro tipo de arcos (linha ponteada) que permitem seleccionar um nó de um ramo imediatamente após um outro nó de outro ramo. Note-se que o nó  $j$  pode ser seleccionado imediatamente após o nó  $i$ , se o antecessor de  $j$ , no caminho da raiz para  $j$ , já tiver sido seleccionado, o que acontece, seguindo a ordenação  $DFO$ , se o antecessor de  $j$  estiver no caminho da raiz para o nó  $i$ .

Portanto, uma sub-árvore em  $T$  corresponde a um caminho de 0 para  $D$  em  $G$  e vice-versa.

Para cada nó  $i \in N$ , sejam:

- $N^-(i)$ : o conjunto dos antecessores de  $i$ ;
- $N^+(i)$ : o conjunto dos sucessores de  $i$ .

Seja  $P(j)$  o único caminho do vértice  $i$  para o vértice  $j$ . E considere-se que o arco  $(i, j)$  existe se  $P(p_j) \subseteq P(i)$ .

A cada arco associa-se uma variável binária que indica se o arco está ou não na solução óptima:

$$x_{(i,j)} = \begin{cases} 1, & \text{se o arco de } i \text{ para } j \text{ está na solução;} \\ 0, & \text{caso contrário.} \end{cases}, \quad j = 1, \dots, n, \quad i \in N^-(j)$$

e a cada arco  $(i, D)$  também se associa uma variável binária que indica se o arco está ou não na solução óptima:

$$z_i = \begin{cases} 1, & \text{se o arco de } i \text{ para } D \text{ está na solução;} \\ 0, & \text{caso contrário.} \end{cases}, \quad i = 1, \dots, n.$$

#### 4.4 Redução de outros problemas

---

Assim, o PSARC pode ser formulado do seguinte modo:

$$\max \sum_{i \in \{1, \dots, n\}} c_i \sum_{j \in N^+(i)} x_{(i,j)} \quad (4.14)$$

sujeito a:

$$\sum_{j \in N^+(0)} x_{(0,j)} + z_0 = 1 \quad (4.15)$$

$$\sum_{i \in N^-(j)} x_{(i,j)} = \sum_{t \in N^+(j)} x_{(j,t)} + z_j, \quad \forall 1 \leq j < n \quad (4.16)$$

$$\sum_{i \in N^-(n)} x_{(i,n)} = z_n \quad (4.17)$$

$$\sum_{i \in N^-(n)} z_i = 1 \quad (4.18)$$

$$\sum_{j \in \{1, \dots, n\}} d_j \sum_{i \in N^-(j)} x_{(i,j)} \leq w \quad (4.19)$$

$$x_{i,j} \in \{0, 1\}, \quad \forall (i,j) \in A. \quad (4.20)$$

em que as restrições (4.15) garantem que do primeiro nó, o nó 0, sai apenas um arco. As restrições (4.16) garantem a conservação do fluxo em cada nó  $j$ , com  $1 \leq j < n$ . As restrições (4.17) garantem a conservação do fluxo no nó  $n$ . As restrições (4.18) garantem que ao último nó, o nó  $D$ , chega apenas um arco e finalmente as restrições (4.19) garantem que a capacidade do saco-mochila não é excedida.

O problema (4.14)-(4.20) é um PCMC com uma restrição de recurso.

Com base nesta redução é possível aplicar as técnicas de resolução do PCMCRA ao PSARC.

## Capítulo 5

# Aplicação: sub-problema de outros problemas

Para além das aplicações directas do PCMCRA a problemas reais, os PCMCRA ocorrem também como sub-problemas de inúmeros problemas reais. É o caso dos problemas de rotas de veículos, também conhecido por *Vehicle Routing Problem* (VRP).

Existem muitas variantes do problema de rotas de veículos. De seguida identificam-se alguns desses problemas:

- Problema de rotas de veículos com janelas temporais em que todos os clientes têm que ser servidos num determinado intervalo de tempo (*VRP with time windows*), [22];
- Problema de rotas de veículos com restrições de capacidade em que todos os veículos têm um limite de capacidade, (*Capacitated VRP*) [40];
- Problema de rotas de veículos com vários armazéns em que o vendedor tem que abastecer em vários armazéns para visitar os clientes, (*Multiple Depot VRP*) [7];
- Problema de rotas de veículos com entrega e recolha em que os clientes podem enviar e trazer materiais para o armazém (*VRP with Pick-Up and Delivering*), [17];

## 5.1 Rotas de Veículos em Problemas com Janelas Temporais

---

Muitos problemas resultam da combinação de dois ou mais dos problemas.

Neste capítulo é exemplificada apenas a aplicação dos problemas de CMCRA ao problema da determinação de rotas de veículos com janelas temporais.

### 5.1 Rotas de Veículos em Problemas com Janelas Temporais

A determinação de rotas para veículos em problemas com janelas temporais, VRPTW (*Vehicle Routing Problem with Time Windows*), é uma generalização do problema de determinação de rotas para veículos, *Vehicle Routing Problem* e consiste na determinação de um conjunto de rotas para um conjunto de veículos que têm que visitar um conjunto de clientes. Cada rota é definida como um percurso que tem início num ponto inicial (armazém), visita, por uma determinada ordem, um sub-conjunto de clientes, cada um dentro de um intervalo de tempo específico (janela temporal), retornando finalmente ao mesmo ponto inicial. Nenhum veículo satisfaz um conjunto de procura superiores à sua capacidade.

O objectivo é minimizar o total das distâncias a percorrer pelos veículos. Adicionalmente pode-se considerar também limites no número de veículos ou custos de utilização dos veículos. Para um estudo mais aprofundado consultar [19] e [22].

#### Formulação do Problema

O problema é definido por uma frota de veículos,  $V$ , um conjunto de clientes,  $C$ , e um grafo  $G = (N, A)$ . A frota é considerada homogénea (todos os veículos têm características semelhantes). O grafo possui  $|C| + 2$  nós, onde os clientes são denotados por  $1, 2, \dots, n$  e o armazém é denotado pelo nó  $0$  (ponto de partida) e  $n + 1$  (ponto de chegada). O conjunto dos nós,  $\{0, 1, \dots, n + 1\}$  é denotado por  $N$  e o conjunto dos arcos,  $A$ , representam as ligações directas entre a origem, os clientes e o ponto de chegada (retorno ao armazém). Não há nenhum arco incidente no nó  $0$  nem nenhum arco com

origem (divergente) no nó  $n + 1$ .

A cada arco  $(i, j) \in A$ , com  $i \neq j$ , está associado um custo  $c_{ij}$ , por exemplo, a distância, e um tempo  $t_{ij}$ , que pode incluir o tempo de visita ao cliente  $i$ .

Cada veículo tem uma determinada capacidade  $q$ , que supomos igual para todos os veículos e não pode ser excedida, e a cada cliente  $i$  corresponde um determinado pedido  $d_i$ .

Para cada cliente, o início da entrega deve estar dentro de uma janela temporal,  $[a_i; b_i]$ . Um veículo pode chegar a um cliente antes do início da sua janela temporal e esperar até que seja possível efectuar as entregas (hora de início da janela temporal desse cliente), mas não pode chegar depois do fim da janela temporal.

Assume-se que todos os dados  $q, a_i, b_i, d_i, i \in C, c_{ij}, (i, j) \in A$  têm valores não negativos e  $t_{ij}, (i, j) \in A$  são positivos. Finalmente, assume-se também que os custos  $c_{ij}$  e os tempos  $t_{ij}$  satisfazem a desigualdade triangular.

Como variáveis de decisão, ter-se-á:

1. Para cada arco  $(i, j) \in A$ , com  $i \neq j, i \neq n+1, j \neq 0$ , e cada veículo  $k$ , as variáveis

$x_{ij}^k$  definidas como

$$x_{ij}^k = \begin{cases} 1, & \text{se o veículo } k \text{ se desloca directamente do nó } i \text{ para o nó } j \\ 0, & \text{no caso contrário.} \end{cases}$$

2. As variáveis  $s_i^k$ , definidas para cada nó  $i$  e para cada veículo  $k$ , que nos indicam o instante de tempo em que um veículo  $k$  inicia o serviço no cliente  $i$ . Assume-se que  $s_0^k = 0$  e que  $s_{n+1}^k$  indica o tempo de chegada do veículo  $k$  ao armazém.

Para um determinado número de veículos, o objectivo é definir as rotas, minimizando os custos e onde:

## 5.1 Rotas de Veículos em Problemas com Janelas Temporais

---

- cada cliente é servido apenas uma vez;
- todas as rotas têm origem no nó 0 e terminam no nó  $n + 1$ ;
- as restrições de capacidade dos veículos são satisfeitas;
- a janela temporal de cada cliente é satisfeita.

Matematicamente, pode-se formular este problema da seguinte forma:

$$\min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^k, \quad (5.1)$$

sujeito a:

$$\sum_{k \in V} \sum_{j \in N} x_{ij}^k = 1, \quad \forall i \in C, \quad (5.2)$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ij}^k \leq q, \quad \forall k \in V, \quad (5.3)$$

$$\sum_{j \in N} x_{0j}^k = 1, \quad \forall k \in V, \quad (5.4)$$

$$\sum_{i \in N} x_{ih}^k - \sum_{j \in N} x_{hj}^k = 0, \quad \forall h \in C, \quad \forall k \in V \quad (5.5)$$

$$\sum_{i \in N} x_{i,n+1}^k = 1, \quad \forall k \in V, \quad (5.6)$$

$$x_{ij}^k (s_i^k + t_{ij} - s_j^k) \leq 0, \quad \forall i, j \in N, \quad \forall k \in V, \quad (5.7)$$

$$a_i \leq s_i^k \leq b_i, \quad \forall i \in N, \quad \forall k \in V, \quad (5.8)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in N, \quad \forall k \in V. \quad (5.9)$$

A função objectivo (5.1) minimiza o total de custos da viagem. As restrições (5.2) garantem que cada cliente é visitado apenas uma vez assegurando que de cada cliente sai um só veículo que segue para um outro cliente ou para o armazém. As restrições (5.3) são relativas à capacidade do veículo, garantindo que nenhum veículo satisfaz um conjunto de procuras superiores à sua capacidade. As restrições (5.4), (5.5) e (5.6), são restrições de conservação de fluxo que garantem que cada veículo parte do nó 0, só sai de um nó  $h$  se lá tiver entrado previamente e que termina o percurso no nó  $n + 1$ .



As restrições (5.7) estabelecem a relação entre a hora do início da visita a um cliente com a hora de início da visita ao cliente seguinte. Note-se que as restrições (5.7) não são lineares mas podem ser convertidas em restrições lineares (ver secção 3.3.1). As restrições (5.8) garantem que todas as janelas temporais são respeitadas.

## 5.2 Relaxações Lagrangeanas

De seguida considere-se algumas abordagens clássicas para o VRPTW, onde ocorre como sub-problema, o problema da determinação do Caminho mais Curto com restrições adicionais.

Comece-se por reformular o modelo anterior, introduzindo novas variáveis binárias  $y_i^k$  que indicam se o veículo  $k$  visita ou não o cliente  $i$ . Portanto  $y_i^k = \sum_{j \in N} x_{ij}^k$  e obtem-se então a seguinte formulação:

$$(FORM1) \quad \min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^k$$

sujeito a:

$$\sum_{k \in V} y_i^k = 1, \quad \forall i \in C, \quad (5.10)$$

$$\sum_{i \in C} d_i y_i^k \leq q, \quad \forall k \in V, \quad (5.11)$$

$$y_i^k = \sum_{j \in N} x_{ij}^k, \quad \forall k \in V, \quad \forall i \in C, \quad (5.12)$$

$$\sum_{j \in N} x_{0j}^k = 1, \quad \forall k \in V, \quad (5.13)$$

$$\sum_{i \in N} x_{ih}^k - \sum_{j \in N} x_{hj}^k = 0, \quad \forall h \in C, \quad \forall k \in V \quad (5.14)$$

$$\sum_{i \in N} x_{i,n+1}^k = 1, \quad \forall k \in V, \quad (5.15)$$

## 5.2 Relaxações Lagrangeanas

---

$$x_{ij}^k (s_i^k + t_{ij} - s_j^k) \leq 0, \quad \forall i, j \in N, \quad \forall k \in V, \quad (5.16)$$

$$a_i \leq s_i^k \leq b_i, \quad \forall i \in C, \quad \forall k \in V, \quad (5.17)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in N, \quad \forall k \in V, \quad (5.18)$$

$$y_i^k \in \{0, 1\}, \quad \forall k \in V, \quad \forall i \in C. \quad (5.19)$$

De seguida, são analisadas diferentes relaxações Lagrangeanas para o VRPTW baseadas na relaxação das restrições de ligação (5.12), que ligam as variáveis  $y_i^k$  às variáveis  $x_i^k$ .

### Relaxação 1

Associando um multiplicador  $\lambda^k \in \mathbb{R}$  a cada restrição  $y_i^k = \sum_{j \in N} x_{ij}^k$ ,  $\forall k \in V$ , obtém-se o seguinte problema relaxado:

$$L^1(\lambda) = \min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^k + \sum_{k \in V} \lambda^k \left( \sum_{j \in N} x_{ij}^k - y_i^k \right)$$

sujeito a:

$$\sum_{k \in V} y_i^k = 1, \quad \forall i \in C, \quad (5.20)$$

$$\sum_{i \in C} d_i y_i^k \leq q, \quad \forall k \in V, \quad (5.21)$$

$$y_i^k \in \{0, 1\}, \quad \forall i \in C, \quad \forall k \in V, \quad (5.22)$$

$$\sum_{j \in N} x_{0j}^k = 1, \quad \forall k \in V, \quad (5.23)$$

$$\sum_{i \in N} x_{ih}^k - \sum_{j \in N} x_{hj}^k = 0, \quad \forall h \in C, \quad \forall k \in V \quad (5.24)$$

$$\sum_{i \in N} x_{i,n+1}^k = 1, \quad \forall k \in V, \quad (5.25)$$

$$x_{ij}^k (s_i^k + t_{ij} - s_j^k) \leq 0, \quad \forall i, j \in N, \quad \forall k \in V, \quad (5.26)$$

$$a_i \leq s_i^k \leq b_i, \quad \forall i \in C, \quad \forall k \in V, \quad (5.27)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in N, \quad \forall k \in V. \quad (5.28)$$

Assim, a função objectivo de  $L^1(\lambda)$  pode ser escrita da seguinte forma:

$$\begin{aligned} \min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^k + \sum_{k \in V} \sum_{j \in N} \lambda^k x_{ij}^k - \sum_{k \in V} \lambda^k y_i^k \\ \min \sum_{k \in V} \sum_{j \in N} \left( \sum_{i \in N} c_{ij} + \lambda^k \right) x_{ij}^k - \sum_{k \in V} \lambda^k y_i^k \end{aligned}$$

Então:

$$L^1(\lambda) = \min \left\{ \sum_{k \in V} \sum_{j \in N} \left( \sum_{i \in N} c_{ij} + \lambda^k \right) x_{ij}^k - \sum_{k \in V} \lambda^k y_i^k : (5.20) - (5.28) \right\}$$

onde as restrições (5.20), (5.21) e (5.22) só envolvem variáveis  $y_i^k$  e as restantes: (5.23) - (5.28) só envolvem as variáveis  $x_{ij}^k$  e  $s_i^k$ . Deste modo, o problema relaxado  $L^1(\lambda)$  envolve dois sub-problemas distintos:

**Sub-problema  $L_1^1$ :**

$$L_1^1(\lambda) = \min \left\{ \sum_{k \in V} \sum_{j \in N} \left( \sum_{i \in N} c_{ij} + \lambda^k \right) x_{ij}^k : (5.23) - (5.28) \right\}$$

**Sub-problema  $L_2^1$ :**

$$L_2^1(\lambda) = \min \left\{ - \sum_{k \in V} \lambda^k y_i^k : (5.20), (5.21), (5.22) \right\}$$

O problema  $L_1^1$ , só envolve variáveis em  $x_{ij}^k$  e  $s_i^k$  e pode ser separado em  $|V|$  problemas, um por cada veículo, sendo cada deles um PCMCJT.

O problema  $L_2^1$ , apenas envolve variáveis  $y_i^k$  e traduz um problema de afectação generalizada [33].

## 5.2 Relaxações Lagrangeanas

---

### Relaxação 2

De seguida é considerada uma nova formulação obtida de (*FORM1*) por substituição das restrições (5.11) por (5.3).

$$(FORM2) \quad \min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^k$$

sujeito a:

$$\sum_{k \in V} y_i^k = 1, \quad \forall i \in C, \quad (5.29)$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ij}^k \leq q, \quad \forall k \in V, \quad (5.30)$$

$$y_i^k = \sum_{j \in N} x_{ij}^k, \quad \forall k \in V, \quad \forall i \in C, \quad (5.31)$$

$$\sum_{j \in N} x_{0j}^k = 1, \quad \forall k \in V, \quad (5.32)$$

$$\sum_{i \in N} x_{ih}^k - \sum_{j \in N} x_{hj}^k = 0, \quad \forall h \in C, \quad \forall k \in V \quad (5.33)$$

$$\sum_{i \in N} x_{i,n+1}^k = 1, \quad \forall k \in V, \quad (5.34)$$

$$x_{ij}^k (s_i^k + t_{ij} - s_j^k) \leq 0, \quad \forall i, j \in N, \quad \forall k \in V, \quad (5.35)$$

$$a_i \leq s_i^k \leq b_i, \quad \forall i \in C, \quad \forall k \in V, \quad (5.36)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in N, \quad \forall k \in V, \quad (5.37)$$

$$y_i^k \in \{0, 1\}, \quad \forall k \in V, \quad \forall i \in C. \quad (5.38)$$

Relaxando as restrições (5.31) obtém-se a seguinte relaxação Lagrangeana:

$$L^2(\lambda) = \min \left\{ \sum_{k \in V} \sum_{j \in N} \left( \sum_{i \in N} c_{ij} + \lambda^k \right) x_{ij}^k - \sum_{k \in V} \lambda^k y_i^k : (5.29), (5.30), (5.32) - (5.38) \right\}.$$

Novamente,  $L^2(\lambda)$  pode ser separado em dois sub-problemas:

**Sub-problema  $L_1^2$**

$$L_1^2(\lambda) = \min \left\{ \sum_{k \in V} \sum_{j \in N} \left( \sum_{i \in N} c_{ij} + \lambda^k \right) x_{ij}^k : (5.30), (5.32) - (5.37) \right\}$$

**Sub-problema  $L_2^2$**

$$L_2^2(\lambda) = \min \left\{ - \sum_{k \in V} \lambda^k y_i^k : (5.29), (5.38) \right\}$$

O problema  $L_1^2$ , só envolve variáveis em  $x_{ij}^k$  e  $s_i^k$  sendo um PCMC com janelas temporais e restrições de recurso.

O problema  $L_2^2$ , apenas envolve variáveis  $y_{ij}^k$  e traduz um problema de semi-afectação.

**Relaxação 3**

Considerando agora a formulação que se obtém por inclusão simultânea das restrições (5.11) e (5.12):

$$(FORM3) \quad \min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^k$$

sujeito a:

$$\sum_{k \in V} y_i^k = 1, \quad \forall i \in C, \tag{5.39}$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ij}^k \leq q, \quad \forall k \in V, \tag{5.40}$$

$$\sum_{i \in C} d_i y_{ij}^k \leq q, \quad \forall k \in V, \tag{5.41}$$

$$y_i^k = \sum_{j \in N} x_{ij}^k, \quad \forall k \in V, \quad \forall i \in C, \tag{5.42}$$

## 5.2 Relaxações Lagrangeanas

---

$$\sum_{j \in N} x_{0j}^k = 1, \quad \forall k \in V, \quad (5.43)$$

$$\sum_{i \in N} x_{ih}^k - \sum_{j \in N} x_{hj}^k = 0, \quad \forall h \in C, \quad \forall k \in V \quad (5.44)$$

$$\sum_{i \in N} x_{i,n+1}^k = 1, \quad \forall k \in V, \quad (5.45)$$

$$x_{ij}^k (s_i^k + t_{ij} - s_j^k) \leq 0, \quad \forall i, j \in N, \quad \forall k \in V, \quad (5.46)$$

$$a_i \leq s_i^k \leq b_i, \quad \forall i \in N, \quad \forall k \in V, \quad (5.47)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in N, \quad \forall k \in V, \quad (5.48)$$

$$y_i^k \in \{0, 1\}, \quad \forall k \in V, \quad \forall i \in C. \quad (5.49)$$

Relaxando novamente a restrição (5.42) obtém-se a seguinte relaxação Lagrangeana:

$$L^3(\lambda) = \min \left\{ \sum_{k \in V} \sum_{j \in N} \left( \sum_{i \in N} c_{ij} + \lambda^k \right) x_{ij}^k - \sum_{k \in V} \lambda^k y_i^k : (5.39) - (5.49) \right\}.$$

Novamente,  $L^3(\lambda)$  pode ser separado em dois sub-problemas:

**Sub-problema  $L_1^3$**

$$L_1^3(\lambda) = \min \left\{ \sum_{k \in V} \sum_{j \in N} \left( \sum_{i \in N} c_{ij} + \lambda^k \right) x_{ij}^k : (5.40), (5.42), (5.43) - (5.48) \right\}$$

**Sub-problema  $L_2^3$**

$$L_2^3(\lambda) = \min \left\{ - \sum_{k \in V} \lambda^k y_i^k : (5.39), (5.41), (5.42), (5.49) \right\}$$

O problema  $L_1^3$ , só envolve variáveis em  $x_{ij}^k$  e  $s_i^k$  sendo um PCMC com janelas temporais e restrições de recurso.

O problema  $L_2^3$ , apenas envolve variáveis  $y_i^k$  e traduz um problema de afectação.

### 5.3 Geração de Colunas

O método de Geração de Colunas tem sido aplicado em vários problemas, como por exemplo o PCMCJT para resolução da respectiva relaxação linear. Usualmente recorre-se ao método de Geração de Colunas quando o problema em causa envolve um número muito elevado de variáveis. Este tipo de problemas estão usualmente associado ao método de decomposição de Dantzig-Wolfe (DWD).

Os princípios básicos da geração de colunas e da decomposição de Dantzig-Wolfe, foram publicados há mais de quatro décadas. Embora os princípios sejam há muito conhecidos, só mais recentemente este método mostrou ser muito eficiente na resolução de problemas reais. Para isso contribuiu em grande medida a associação do método de geração de colunas ao método Branch and Bound e o grande desenvolvimento da capacidade computacional. Desta associação nasceram os métodos Branch and Price.

O PCMCRA ocorre como sub-problema de outros problemas mais complexos quando se recorre ao método de Geração de Colunas para se resolver as relaxações lineares desses problemas.

Por exemplo, Cláudio Alves e José Carvalho em [16] analisaram um novo problema de planeamento de rotas, o *Prize Collecting Vehicle Routing Problem with service restrictions*, sugerido por um caso de recolha de desperdícios de madeira. É um problema onde a frota é homogénea, o depósito único e em que a visita a alguns clientes não é obrigatória, mas é condicionada pelas necessidades totais de desperdícios. Propõem uma formulação para o problema que deriva de um modelo de fluxo de três índices para o problema de planeamento de rotas de veículos. Para a optimização do plano de rotas, exploraram métodos de decomposição. Analisam, em particular, a aplicação do método de decomposição de Dantzig-Wolfe à formulação proposta e, para a obtenção de soluções inteiras, o método branch and price.

### 5.3 Geração de Colunas

---

Mais exemplos podem ser encontrados no artigo recente de Faramroze, Nemhauser e Savelsbergh em [14], onde é desenvolvido um algoritmo baseado na programação dinâmica para resolver o PCMCRA que ocorre no recurso ao método de geração de colunas para problemas de grande dimensão.

Na secção 5.3.2, é mostrado como o PCMCRA ocorre como sub-problema na resolução de uma relaxação do VRPTW usando o método de geração de colunas.

Para exemplificar os conceitos básicos, de seguida é exemplificada a aplicação do método de geração de colunas ao PCMC com uma restrição de recurso (ver secção 3.3.2), baseando-nos em [26].

#### 5.3.1 Geração de colunas na resolução da relaxação linear do PCMCRR

Considere-se um grafo acíclico  $G = (N, A)$ . Seja  $c_{ij}$  o custo associado a cada arco  $(i, j) \in A$  e seja  $w_{ij}$  o peso associado ao arco  $(i, j) \in A$ . O objectivo é encontrar o caminho mais curto desde o nó origem  $s$  até ao nó destino  $t$ , de tal modo que o tempo total da viagem do caminho encontrado não exceda  $w$  unidades de tempo.

Pode-se então escrever a seguinte formulação:

$$z^* = \min \sum_{(i,j) \in A} c_{ij} x_{ij}, \quad (5.50)$$



sujeito a:

$$\sum_{j:(s,j) \in A} x_{sj} = 1, \quad (5.51)$$

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(i,j) \in A} x_{ji} = 0, \quad \forall i \in N \setminus \{s, t\} \quad (5.52)$$

$$\sum_{i:(i,t) \in A} x_{it} = 1, \quad (5.53)$$

$$\sum_{(i,j) \in A} w_{ij} x_{ij} \leq w, \quad (5.54)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (5.55)$$

Seja  $X$  o conjunto das soluções que satisfazem (5.51)-(5.53) e (5.55). Note-se que as soluções extremas do invólucro convexo de  $X$  correspondem a caminhos.

Seja  $P$  o conjunto de todos os caminhos de  $s$  para  $t$ . A cada caminho  $p \in P$  associam-se as variáveis  $x_{ij}^p$  que indicam se o arco  $(i, j)$  pertence ao caminho  $P$ . Assim, pode-se expressar cada aresta como uma combinação linear convexa dos caminhos em  $P$ :

$$x_{ij} = \sum_{p \in P} x_{ij}^p \lambda_p, \quad (i, j) \in A, \quad (5.56)$$

$$\sum_{p \in P} \lambda_p = 1, \quad (5.57)$$

$$\lambda_p \geq 0, \quad p \in P. \quad (5.58)$$

Agora, se forem substituídas as variáveis  $x_{ij}$  na formulação anterior, obtém-se o chamado Problema Mestre (*Master Problem*) formalizado por:

$$z^* = \min \sum_{p \in P} \left( \sum_{(i,j) \in A} c_{ij} x_{ij}^p \right) \lambda_p$$

### 5.3 Geração de Colunas

---

sujeito a:

$$\sum_{p \in P} \left( \sum_{(s,j) \in A} w_{ij} x_{ij}^p \right) \lambda_p \leq w, \quad (5.59)$$

$$\sum_{p \in P} \lambda_p = 1, \quad (5.60)$$

$$\lambda_p \geq 0, \quad p \in P, \quad (5.61)$$

$$\sum_{p \in P} x_{ij}^p \lambda_p = x_{ij}, \quad (i, j) \in A, \quad (5.62)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A. \quad (5.63)$$

onde o coeficiente de  $\lambda_p$  na função objectivo é o custo do caminho  $p$  e o coeficiente  $\lambda_p$  em (5.59) corresponde à duração do caminho  $p$ . Com as restrições (5.62) e (5.63) são preservadas as ligações das variáveis e é recuperada a solução nas variáveis  $x_{ij}$  para o problema original (5.51)-(5.55) com base numa solução do problema mestre.

Ao considerar a relaxação linear deste problema, isto é, ao se relaxar (5.63), já não há a necessidade de ligação das variáveis  $x_{ij}$  e  $\lambda_p$  e podem-se deixar as restrições (5.62) como estão. Sejam  $\pi_1$  e  $\pi_0$  as variáveis duais associadas às restrições (5.59) e (5.60) respectivamente. Para grafos de grandes dimensões, a cardinalidade de  $P$  seria impeditiva e não se conseguiria explicitar o estado de todas as variáveis no problema mestre. Assim, a ideia da “geração de colunas” é trabalhar com um subconjunto aceitável de variáveis (neste caso, caminhos), formando o problema mestre restrito (*restricted master problem*). As variáveis são adicionadas quando são necessárias, tal como no método do Simplex. Em cada iteração é verificado se existe uma variável não básica com custo reduzido negativo. Caso exista, uma dessas variáveis torna-se variável básica por substituição com uma variável básica. Caso contrário, a solução corrente é óptima.

No método de geração de colunas cada iteração consiste em:

- a) otimizar o problema restrito de modo a determinar o valor óptimo  $\bar{z}$  da função objectivo e determinar os multiplicadores duais  $\pi_0$  e  $\pi_1$ , e

b) encontrar, se existir, a variável  $\lambda_p$  com menor custo reduzido negativo:

$$\bar{c}_p = \sum_{(i,j) \in A} c_{ij} x_{pij} - \pi_1 \left( \sum_{(i,j) \in A} w_{ij} x_{pij} \right) - \pi_0 < 0. \quad (5.64)$$

A determinação da variável com menor custo reduzido reduz-se a otimizar um sub-problema, no nosso caso: o problema do caminho mais curto com uma modificação da estrutura nos custos:

$$\bar{c}^* = \min \sum_{(i,j) \in A} (c_{ij} - \pi_1 w_{ij}) x_{ij} - \pi_0. \quad (5.65)$$

sujeito a: (5.51)-(5.53), (5.55).

Assim, se  $\bar{c}^* \geq 0$ , todos os custos reduzidos são não negativos e obtém-se por isso a solução óptima da relaxação linear do problema mestre. Caso contrário, a variável encontrada é adicionada ao problema restrito e repete-se o processo.

Note-se que na resolução deste problema do CMC não é necessário considerar (5.55) pois todas as soluções extremas de (5.51)-(5.53) são inteiras.

### 5.3.2 Aplicação ao Problema de Planeamento de Rotas

Considere-se a formulação do problema de determinação de rotas veículos (5.1)-(5.9), apresentada na secção (5.1).

O método aqui apresentado foi introduzido por [18] para o VRPTW considerando apenas um veículo.

A abordagem da geração de colunas explora o facto de apenas a restrição (5.2) relacionar os veículos entre si.

Seja  $P^k$  o conjunto de caminhos admissíveis do veículo  $k$ ,  $k \in V$ , isto é, caminhos que correspondem a soluções que satisfazem (5.3)-(5.9). Assim,  $p \in P^k$  corresponde a

### 5.3 Geração de Colunas

---

um caminho que pode ser descrito usando os valores binários  $x_{ijp}^k$ , onde  $x_{ijp}^k = 1$  se o veículo  $k$  vai directamente do nó  $i$  para o nó  $j$  e  $x_{ijp}^k = 0$  no caso contrário.

Considera-se o problema mestre inteiro definido por (5.1)-(5.2) e (5.9), ou seja, contém a função objectivo, a afectação de um veículo a um cliente e as variáveis de decisão. As restantes restrições são apenas consideradas no sub-problema para a determinação dos caminhos, o qual, como se verá mais adiante, tem a função objectivo modificada e pode ser decomposto em  $|V|$  sub-problemas independentes, um por cada veículo.

Trabalhe-se agora com o problema mestre linear (5.1)-(5.2) que se obtém do problema mestre inteiro por remoção das restrições (5.9).

Qualquer solução  $x_{ij}^k$  do problema mestre (linear) (5.1)-(5.2) pode ser escrita como uma combinação convexa não negativa de um número finito de caminhos em  $P^k$  (soluções de (5.3)-(5.9), isto é,

$$x_{ij}^k = \sum_{p \in P^k} x_{ijp}^k y_p^k, \quad \forall k \in V, \quad \forall (i, j) \in A, \quad (5.66)$$

$$\sum_{p \in P^k} y_p^k = 1, \quad \forall k \in V, \quad (5.67)$$

$$y_p^k \geq 0, \quad \forall k \in V, \quad \forall p \in P^k. \quad (5.68)$$

Substituindo  $x_{ij}^k$  em (5.1)-(5.2) vem:

$$\min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} \sum_{p \in P^k} x_{ijp}^k y_p^k, \quad (5.69)$$

sujeito a:

$$\sum_{k \in V} \sum_{j \in N} \sum_{p \in P^k} x_{ijp}^k y_p^k = 1, \quad \forall i \in C, \quad (5.70)$$

$$\sum_{p \in P^k} y_p^k = 1, \quad \forall k \in V, \quad (5.71)$$

$$y_p^k \geq 0, \quad \forall k \in V, \quad \forall p \in P^k. \quad (5.72)$$

---

## 5. Aplicação: sub-problema de outros problemas

---

Para simplificar a notação pode-se definir o custo do caminho,  $c_p^k$ ,  $p \in P^k$  por:

$$c_p^k = \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijp}^k, \quad \forall k \in V, \quad \forall p \in P^k. \quad (5.73)$$

O coeficiente de  $y_p^k$  em (5.70) representa o número de vezes que o cliente  $i$  é visitado pelo veículo  $k$ , definido por  $a_{ip}^k$ :

$$a_{ip}^k = \sum_{j \in N} x_{ijp}^k, \quad \forall k \in V, \quad \forall i \in N, \quad \forall p \in P^k. \quad (5.74)$$

Assim, pode-se substituir estes coeficientes no problema (5.1)-(5.2) e obtém-se a formulação do problema mestre (linear):

$$\min \sum_{k \in V} \sum_{p \in P^k} c_p^k y_p^k \quad (5.75)$$

sujeito a:

$$\sum_{k \in V} \sum_{p \in P^k} a_{ip}^k y_p^k = 1, \quad \forall i \in C, \quad (5.76)$$

$$\sum_{p \in P^k} y_p^k = 1, \quad \forall k \in V, \quad (5.77)$$

$$y_p^k \geq 0, \quad \forall k \in V, \quad \forall p \in P^k. \quad (5.78)$$

Note-se que substituindo (5.78) por

$$y_p^k \in \{0, 1\}, \quad \forall k \in V, \quad \forall p \in P^k$$

então obtém-se uma formulação alternativa para o VRPTW. Esta formulação é a generalização da reformulação, como Problema de Partição, do problema VRPTW (ver [18]) para o caso em que existe mais do que um veículo. Neste caso, as variáveis  $y_p^k$  podem ser interpretadas como variáveis binárias que indicam se o caminho  $p \in P^k$  é escolhido ou não. As restrições (5.77) garantem que para cada veículo é escolhido um e um só caminho. Este caminho pode conter apenas o arco  $(0, n + 1)$ , o que significa que o veículo não é usado. As restrições (5.76) garantem que cada cliente é servido por um único veículo e portanto um único caminho.

### 5.3 Geração de Colunas

---

Como se está a assumir que existe apenas um armazém e a frota de veículos é homogénea com as mesmas condições iniciais para todos os veículos, então o conjunto dos caminhos,  $P^k$  não varia de veículo para veículo, isto é,  $P^k = P, \forall k \in V$  e portanto  $x_{ijp}^k = x_{ijp}, \forall k \in V, a_{ijp}^k = a_{ijp}, \forall k \in V, c_p^k = c_p, \forall k \in V$ .

Substituindo  $P^k$  por  $P, a_{ip}^k$  por  $a_{ip}$  e  $c_p^k$  por  $c_p$  em (5.75)-(5.78) tem-se:

$$\min \sum_{k \in V} \sum_{p \in P} c_p^k y_p^k \quad (5.79)$$

sujeito a:

$$\sum_{k \in V} \sum_{p \in P} a_{ip} y_p^k = 1, \quad \forall i \in C, \quad (5.80)$$

$$\sum_{p \in P} y_p^k = 1, \quad \forall k \in V, \quad (5.81)$$

$$y_p^k \geq 0, \quad \forall k \in V, \quad \forall p \in P. \quad (5.82)$$

As restrições (5.82) podem ser agregadas resultando a restrição:

$$\sum_{p \in P} \sum_{k \in V} y_p^k = |V|. \quad (5.83)$$

Como se está a assumir que o número de veículos é suficientemente elevado, pode-se eliminar a restrição (5.83).

Se for definido  $y_p$  por  $y_p = \sum_{k \in V} y_p^k$ , então o índice  $k$  pode ser retirado da formulação (5.79)-(5.82). Obtém-se assim a relaxação linear clássica do problema da partição:

$$\min \sum_{p \in P} c_p y_p \quad (5.84)$$

sujeito a:

$$\sum_{p \in P} a_{ip} y_p = 1, \quad \forall i \in C, \quad (5.85)$$

$$y_p \geq 0, \quad \forall p \in P. \quad (5.86)$$

---

## 5. Aplicação: sub-problema de outros problemas

---

O problema mestre restrito que se obtém do problema mestre por exclusão de um subconjunto de variáveis (colunas/caminhos) pode então ser formulado matematicamente por:

$$\min \sum_{p \in P'} c_p y_p \quad (5.87)$$

sujeito a:

$$\sum_{p \in P'} a_{ip} y_p = 1, \quad \forall i \in C, \quad (5.88)$$

$$y_p \geq 0, \quad \forall p \in P'. \quad (5.89)$$

em que o conjunto  $P'$  contém todos os caminhos já gerados,  $a_{ip}$  denota o número de vezes que o cliente  $i$  é servido no caminho  $p$  e  $c_p$  corresponde ao custo do caminho e  $y_p$  representa o número de vezes que o caminho  $p \in P'$  é usado.

Note-se que  $P'$  inclui usualmente um número de caminhos muito inferior ao de  $P$ . Inicialmente podem-se considerar em  $P'$  as rotas que consistem em cada veículo visitar apenas um cliente, isto é,  $0 \rightarrow i \rightarrow n + 1$ .

Ao resolver o problema mestre restrito é obtida uma solução  $y = (y_1, y_2, \dots, y_{|P'|})$ . Se  $y$  for inteira, então é uma solução admissível, mas não necessariamente uma solução ótima para o PCMCJT. Para além disso, da resolução do problema mestre restrito é obtida uma solução dual,  $\pi_1, \dots, \pi_n$ .

Quando a solução ótima do problema mestre restrito é encontrada, no método de geração de colunas é necessário procurar uma nova variável (caminho em  $P \setminus P'$ ) com custos reduzidos negativos. Esse caminho é encontrado através da resolução de um sub-problema.

Note-se que se for considerado  $P^k = P, \forall k \in V$ , o sub-problema original fica decomposto em  $|V|$  sub-problemas idênticos, um por cada veículo. Assim, a nova

### 5.3 Geração de Colunas

---

variável (caminho) é obtida resolvendo o seguinte sub-problema (PCMCJT):

$$\min \sum_{i \in N} \sum_{j \in N} \widehat{c}_{ij} x_{ij} \quad (5.90)$$

sujeito a:

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ij} \leq q, \quad (5.91)$$

$$\sum_{j \in N} x_{0j} = 1, \quad (5.92)$$

$$\sum_{i \in N} x_{ih} - \sum_{j \in N} x_{hj} = 0, \forall h \in C, \quad (5.93)$$

$$\sum_{i \in N} x_{i,n+1} = 1, \quad (5.94)$$

$$x_{ij}(s_i + t_{ij} - s_j) \leq 0, \quad \forall (i, j) \in A, \quad (5.95)$$

$$a_i \leq s_i \leq b_i, \quad \forall i \in C, \quad (5.96)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in N. \quad (5.97)$$

em que a restrição (5.91) é relativa à capacidade do veículo, as restrições (5.95) e (5.96) são relativas às restrições de tempo. As restrições (5.92), (5.93) e (5.94), são restrições de fluxo que garantem que cada ligação parte do nó 0, só sai de um nó  $h$  se lá tiver entrado previamente e que termina o percurso no nó  $n + 1$ .

Na resolução deste sub-problema do VRPTW,  $\widehat{c}_{ij}$  é o custo “modificado” ao percorrer o arco  $(i, j)$ , onde  $\widehat{c}_{ij} = c_{ij} - \pi_i$ . Enquanto que  $c_{ij} \in \mathbb{Z}^+$ , neste caso  $\widehat{c}_{ij}$  pode ser um número real qualquer, portanto deixa de haver garantia que não ocorrem ciclos na resolução do problema (5.90)-(5.97).

É de notar que ao contrário do sub-problema obtido para a resolução da relaxação linear do PCMCRA, aqui tem-se que manter as restrições de integralidade (5.97) nas variáveis  $x_{ij}$ .



# Capítulo 6

## Conclusões

Neste trabalho foi abordado o problema de caminho mais curto com restrições adicionais, onde foram destacados dois tipos de problemas: problemas com restrições de recurso e restrições de janelas temporais.

Foram abordadas várias aplicações reais modeladas como problemas de caminho mais curto com restrições adicionais. Efectuou-se mais detalhadamente o estudo de uma aplicação de planeamento de frotas de veículos e de uma aplicação de encaminhamento de tráfego em redes de comunicação. Por último, efectuou-se uma abordagem ao problema da determinação da sub-árvore óptima de uma árvore com restrições de capacidade, reduzindo-o a um problema de caminho mais curto com restrições adicionais.

Foi ainda estudado o problema de rotas de veículos com janelas temporais, onde após efectuarmos a modelação do problema como um problema de caminho mais curto com restrições adicionais, efectuou-se três tipos de relaxações lagrangeanas, de onde se obtiveram três sub-problemas distintos do tipo PCMCRA. Foi feito ainda um estudo do método de geração de colunas aplicado ao problema de rotas de veículos de onde se obteve, nomeadamente como sub-problema, o PCMCRA.

---

# Bibliografia

- [1] C. Barrico, *Uma abordagem ao problema de caminho mais curto multiobjectivo - aplicação ao problema de encaminhamento em redes integradas de comunicações*, Tese de Mestrado, Faculdade de Ciências e Tecnologia, Universidade de Coimbra, 1998.
- [2] M. Desrochers, *An algorithm for the shortest path problem with resource constraints*, Tech. Report Les Cahiers du GERAD G-88-27, Universidade de Montreal, Canada (1988).
- [3] E. W. Dijkstra, *A discipline of programming*, Prentice Hall, USA, 1997.
- [4] C. Grosan e A. Abraham, *Solving shortest capacitated path problem using a bi-objective heuristic approach*, AMS '07: Proceedings of the First Asia International Conference on Modelling & Simulation (Washington, DC, USA), IEEE Computer Society, 2007, pp. 427–433.
- [5] S. G. Nash e A. Sofer, *Linear and nonlinear programming*, McGraw-Hill, 1996.
- [6] G. Dahl e B. Realfsen, *Curve approximation and constrained shortest path problems*, Networks (2000), no. 36, 1–8.
- [7] J. Carlsson e D. Gez e A. Subramaniam, *Solving min-max multi-depot vehicle routing problem*, (2007), 1–17.
- [8] M. Garey e D. Johnson, *Computers and intractability: a guide to the theory of np-completeness*, Freeman, New York, 1979.

## BIBLIOGRAFIA

---

- [9] A. Elimam e D. Kohler, *Two engineering applications of a constrained shortest-path model*, European Journal of Operational Research (1997), no. 103, 426–438.
- [10] B. Jaumard e F. Semet e T. Vovor, *A two-phase resource constrained shortest path algorithm for acyclic graphs*, Les Cahiers du GERAD G-96-48, École des Hautes Études Commerciales, Montréal, Canada (1996).
- [11] M. Desrochers e F. Soumis, *A generalized permanent labeling algorithm for the shortest path problem with time window*, Infor **26** (1988), 191–212.
- [12] D. X. Shaw e G. Cho, *Reformulation, column generation and lagrangian relaxation for local access network design problems*, (1994).
- [13] S. Irnich e G. Desaulniers, *Shortest path problems with resource constraints*, Column Generation (Desrosiers J. e Solomon M. M. Desaulniers, G., ed.), Springer Science, New York, 2005, pp. 33–65.
- [14] Faramroze. G. E. e G. L. Nemhauser e M. P. Savelsbergh, *Shortest path based column generation on large networks with many resource constraints*, Junho 2008.
- [15] G. Y. Handler e I. Zang, *A dual algorithm for the constrained shortest path problem*, Networks (1980), no. 10, 293–310.
- [16] C. Alves e J. Carvalho, *Planeamento de rotas num sistema de recolha de desperdícios de madeira*, Investigação Operacional (2004), 21–43.
- [17] G. Desaulniers e J. Desrosiers e A. Erdmann e M. M. Solomon e F. Soumis, *Vrp with pickup and delivery*, (2001), 225–242.
- [18] M. Desrochers e J. Desrosiers e M. Solomon, *A new optimization algorithm for the vehicle routing problem with time window*, Operations Research **40** (1992), no. 2, 342–354.
- [19] G. Desaulniers e J. Desrosiers e M. Solomon (Eds.), *Column generation*, Springer Science, New York, 2005.

- [20] N. Boland e J. Dethridge e I. Dumitrescu, *Accelerated label setting algorithms for the elementary resource constrained shortest path problem*, Operations Research Letters **34** (2006), no. 1, 58–68.
- [21] A. Moura e J. F. Oliveira, *Uma heurística composta para a determinação de rotas para veículos em problemas com janelas temporais e entregas e recolhas*, Investição Operacional (2004), no. 24, 45–62.
- [22] B. Kallehauge e J. Larsen e O. Madsen e M. Solomon, *Vehicle routing problem with time windows*, Column Generation (G. Desaulniers e J. Desrosiers e M. Solomon, ed.), Springer Science, New York, 2005, pp. 67–98.
- [23] D. J. Van Der Merwe e J. M. Hattingh, *Tree knapsack approaches for local access network design*, European Journal of Operational Research (2006), no. 174, 1968–1878.
- [24] T. L. Magnati e L. A. Wolsey, *Optimal trees*, Handbooks in Operations Research and Management Science **7** (1995), 503–615.
- [25] P. Avella e M. Boccia e A. Sforza, *Resource constrained shortest path problems in path planning for fleet management*, Journal of Mathematical Modelling and Algorithms **3** (2004), 1–17.
- [26] J. Desrosiers e M. E. Lübbecke, *A primer in column generation*, Column Generation (G. Desaulniers e J. Desrosiers e M. Solomon, ed.), Springer Science, New York, 2005, pp. 1–32.
- [27] G. Righini e M. Salani, *Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints*, Discrete Optimization **3** (2006), 255–273.
- [28] K. Mehlhorn e M. Ziegelmann, *Resource constrained shortest paths*, 7th Ann European Symp on Algorithms (ESA 2000), LNCS 1879, Springer-Verlag, Heidelberg, Berlin (2000), 326–337.

## BIBLIOGRAFIA

---

- [29] ———, *Cnop - a package for constrained network optimization*, ALENEX 2001 (2001), 17–31.
- [30] I. Dumitrescu e N. Boland, *Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem*, Networks **42** (2003), no. 3, 135–153.
- [31] J. E. Beasley e N. Cristofides, *An algorithm for the resource constrained shortest path problem*, Networks **19** (1989), 379–394.
- [32] D. Feillet e P. Dejax e M. Gendreau e C. Gueguen, *An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems*, Networks **44** (2004), no. 3, 216–229.
- [33] S. Martelo e P. Toth, *Knapsack problems: algorithms and computer implementations*, John Wiley and Sons, 1990.
- [34] F. Guerriero e R. Musmanno, *Label correcting methods to solve multicriteria shortest path problems*, Journal of Optimization of Theory and Applications **111** (2001), no. 3, 589–613.
- [35] I. Ioachim e S. Gélinas e F. Soumis e J. Desrosiers, *A dynamic programming algorithm for the shortest path problem with time windows and linear node costs*, Networks (1998), no. 31, 193–204.
- [36] M. Zabrankin e S. Uryasev e P. Pardalos, *Optimal risk path algorithms*, Cooperative Control and Optimization (R. Murphey e P. Pardalos, ed.), vol. 66, Springer US, 2002, pp. 273–298.
- [37] R. Ahuja e T. Magnanti e J. Orlin, *Network flows - theory, algorithms, and applications*, Prentice Hall, 1993.
- [38] P. Aneja e V. Aggarwal e K. Nair, *Shortest chain subject to side conditions*, Networks **13** (1983), 295–302.

- [39] J. Desrosiers e Y. Dumas e M. Solomon e F. Soumis, *Time constrained routing and scheduling*, Handbook of operations research and management science, vol. 8, Network Routing, Amsterdam, 1995, pp. 70–103.
- [40] G. Laporte e Y. Nobert e M. Desrochers, *Optimal routing under capacity and distance restrictions*, Operations Research **33** (1984), no. 5, 1050–1073.
- [41] M. L. Fisher, *The lagrangian relaxation method of solving integer programming problems*, Management Science **27** (1981), 1–28.
- [42] R. Hassin, *Approximation schemes for the restricted shortest path problems*, Mathematics of Operations Research (1992), no. 17, 36–42.
- [43] H. C. Joksch, *The shortest route problem with constraints*, Journal of Mathematical Analysis and Applications (1966), no. 14, 191–197.
- [44] N. Karmarkar, *A new polynomial time algorithm for linear programming*, Combinatorica **4** (1984), 375–395.
- [45] R. B. Shima, *Caminhos mínimos sob restrições: uma revisão e aplicações*, Universidade Federal De Santa Catarina, Florianópolis, 2006.
- [46] S. Silva, *Problema do saco-mochila com estrutura em árvore*, Tese de Mestrado, Universidade de Aveiro, 2008.
- [47] A. Sousa, *Modelos de problemas de optimização em redes de telecomunicações*, Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, Outubro 2006.
- [48] A. Warburton, *Approximation of pareto optima in multiple-objective, shortest-path problems*, Operations Research **35** (1987), no. 1, 70–79.
- [49] Laurence A. Wolsy, *Integer programming*, John Wiley and Sons, 1998.

## BIBLIOGRAFIA

---

- [50] G. Xue, *Primal-dual algorithms for computing weight-constrained shortest paths and weight-constrained minimum spanning trees*, IEEE Computer Society (2000), 271–277.
- [51] M. Ziegelmann, *Constrained shortest path and related problems*, Tese de Doutorado, Universitat de Saarlandes, 2001.