

**Cálculo de soluções eficientes do
problema de *carpooling* biobjetivo**

Teresa de Jesus Romão Figueira de Sousa

Cálculo de soluções eficientes do problema de *carpooling* biobjetivo

Teresa de Jesus Romão Figueira de Sousa

Dissertação para a obtenção do Grau de **Mestre em Matemática**
Área de Especialização em **Estatística, Otimização e Matemática Financeira**

Júri

Presidente: Adérito Araújo

Orientador: Marta Pascoal

Vogal: João Gouveia

3 de Junho de 2015

Resumo

Este trabalho foca problemas de otimização linear inteira com duas funções objetivo. Após introdução de alguns conceitos iniciais descrevem-se vários métodos geradores para o cálculo do conjunto de soluções eficientes de tais problemas, isto é, métodos que calculam as soluções sem recorrer a condições adicionais. Descrevem-se o método NISE, o método das restrições e o método de Tchebycheff, que são aplicados a duas formulações de um problema de transportes em boleias partilhadas, vulgo *carpooling*. Por fim, são apresentados testes computacionais, realizados para concretizações do problema de *carpooling* geradas aleatoriamente. Os métodos discutidos são comparados com base nos resultados obtidos.

Palavras Chave: Otimização linear inteira, Problemas bicritério, *Carpooling*, Métodos geradores de soluções eficientes.

Abstract

This work focuses linear integer optimization problems with two objective functions. After introducing some initial concepts several generating methods for the determination of the set of efficient solutions of such problems, that is, methods that compute the solutions without taking additional conditions into account, are described. The described methods are the NISE method, the constraints method and the Tchebycheff method, which are applied to two formulations of a ride sharing transportation problem, commonly known as carpooling. Finally, computational tests, on randomly generated carpooling problem instances, are presented. The discussed methods are compared based on the obtained results.

Keywords: Linear integer optimization, Bicriteria problems, Carpooling, Efficient solution generating methods.

Agradecimentos

Dedico esta dissertação à minha irmã e aos meus pais. Insisto em não os separar, porque, apesar de o pai já nos ter deixado há alguns anos, há relações que a passagem da morte não destrói.

E agradeço a todos aqueles a quem também o devo:

À minha queridíssima amiga Madalena, pelo incentivo que me deu desde o início.

Ao Filipe, por todo o apoio, paciência, disponibilidade e atenção. E por me mostrar que a programação não é uma área obscura.

Aos professores com quem tive o privilégio de ter aulas ao longo da parte letiva deste segundo ciclo de estudos em Matemática. Pelas reflexões que me proporcionaram e por me terem levado a descobrir a Matemática - também muito interessante - que existe noutras áreas para além daquela que eu já conhecia: o ensino.

Dedico um agradecimento especial à Senhora Doutora Marta Pascoal: pela amabilidade em aceitar ser minha orientadora, pela sua disponibilidade, pelas noções mais elementares de Programação Linear (leccionadas ainda no primeiro ano de mestrado) e, sobretudo, pelas reflexões que o seu pensamento me proporciona e pelo interesse que me cria na área da Programação Linear Multiobjetivo.

Resta ainda um agradecimento aos amigos que fiz em Coimbra.

Conteúdo

1	Introdução	1
2	O problema de <i>carpooling</i>	5
2.1	Formulação como um programa linear inteiro	6
2.2	Formulações como programas lineares bicritério	8
2.3	Conceitos de Programação Linear Biobjetivo	9
3	Resolução de problemas de <i>carpooling</i> bicritério	13
3.1	Método NISE	13
3.2	Método das restrições	16
3.3	Método de Tchebycheff	18
4	Aplicação dos métodos	25
4.1	Método NISE	25
4.2	Método das restrições	26
4.3	Método de Tchebycheff	28
4.4	Método de Tchebycheff com aplicação binária	30
4.5	Método de Tchebycheff com aplicação sequencial	31
5	Resultados computacionais	35
6	Conclusão	41
A	Matrizes de insatisfação	45

Capítulo 1

Introdução

A Programação Linear constitui um dos ramos mais desenvolvidos da matemática aplicada. É uma área científica autónoma relativamente recente (meados do século XX) e engloba um conjunto de técnicas que permitem resolver problemas de otimização, cujas restrições e função objetivo são descritas através de relações lineares.

Em problemas de otimização que modelam o mundo real pretende-se, geralmente, minimizar custos, maximizar lucros, entre outros. Muitas vezes é necessário otimizar vários objetivos simultaneamente, o que se traduz em problemas multiobjetivo, também chamados problemas multicritério. Numa primeira análise, podemos questionar a relevância de formular um problema de otimização considerando mais do que um objetivo. Suponhamos, por exemplo, que se pretende minimizar o custo total de um material a adquirir, assim como a quantidade a encomendar do mesmo material. Estes são dois objetivos que estão fortemente relacionados, pelo que, neste caso, se torna desnecessário considerar ambos explicitamente. No entanto, tendo em atenção a multidimensionalidade da realidade e que a existência de situações com critérios múltiplos e conflituosos entre si é uma constante, os problemas monoobjetivo de otimização com um único objetivo tornam-se, em muitos casos, redutores. Outro exemplo consiste em encontrar o trajeto mais económico para efetuar entregas de uma determinada empresa, o que implica ter em consideração fatores tais como tempo, distância, consumo de combustível e tráfego. Ao contrário do caso anterior, estes fatores poderão ser conflituosos entre si. Como tal, uma solução “boa” relativamente a um deles poderá não o ser para os restantes. No que se segue serão abordados problemas que utilizam dois objetivos, os quais devido à sua natureza, colocam em causa o paradigma da otimalidade e, conseqüentemente, não se podem reduzir a problemas com apenas um objetivo. Este tipo de problemas dizem-se problemas biobjetivo, ou problemas bicritério.

Enquanto no caso monocritério a função objetivo traduz as preferências da entidade que decide (conhecida como o agente de decisão), no caso bicritério devem

reunir-se as preferências do agente de decisão relativas aos vários critérios considerados, e analisar *a posteriori* as diversas opções que conduzem à tomada de decisões, de modo a obter uma solução de compromisso que seja satisfatória.

Os métodos conhecidos para lidar com problemas de Programação Linear Inteira/Inteira-Mista Multiobjetivo podem classificar-se em métodos iterativos e em métodos geradores [5, 6]. Os métodos iterativos compreendem duas fases iniciais: uma de cálculo, em que são calculadas soluções eficientes; e uma de diálogo, em que, face a estas soluções, o agente de decisão é chamado a expressar as suas preferências, de modo a conduzir o processo iterativo e, assim, reduzir o âmbito da pesquisa e o respetivo esforço computacional.

Relativamente aos métodos geradores, estes foram concebidos para determinar todas as soluções eficientes do problema ou um subconjunto destas soluções, dado um critério de relaxamento previamente definido. Os métodos geradores exigem normalmente um esforço computacional elevado, não só por calcularem um número elevado de soluções eficientes, mas, sobretudo, ao realizarem por si só o cálculo da próxima solução eficiente sem interação do agente de decisão. Por esta razão, observa-se que a maior parte dos métodos geradores se destina a problemas apenas com variáveis binárias, ou a problemas biobjetivo.

A presente dissertação centra-se nos métodos geradores para problemas de Programação Linear Inteira Biobjetivo. Em particular, os métodos geradores serão aplicados a um problema de *carpooling* com duas funções objetivo. Este problema pode ser representado através de um programa linear inteiro, e portanto os métodos geradores permitirão calcular todas as soluções eficientes.

O trabalho organiza-se do seguinte modo. No Capítulo 2 será apresentado o conceito de *carpooling* e a formulação do problema de *carpooling* como um programa linear inteiro. Por forma a ter um modelo que reflita o espírito do *carpooling*, apresentaremos duas formulações alternativas do problema que incluem, cada uma, uma função objetivo adicional. Serão também introduzidos alguns conceitos de Programação Linear Biobjetivo. Em seguida, no Capítulo 3, descrever-se-ão alguns métodos para resolução de problemas de otimização linear com dois objetivos: o método NISE, o método das restrições, e o método de Tchebycheff. No Capítulo 4, exemplificar-se-ão os métodos descritos no capítulo anterior para resolução do problema de *carpooling* com dois objetivos. Consideram-se duas abordagens: um problema de *carpooling* com insatisfação e número de veículos mínimos (PCIVM); e um problema de *carpooling*

com insatisfação e emissões de gases poluentes mínimas de cada veículo (PCIEM).

Por fim, no Capítulo 5 os métodos apresentados serão testados, por meio de experiência computacional, considerando o problema de *carpooling* em que se pretendem minimizar a insatisfação e a emissão de gases poluentes. Os resultados recolhidos sobre o desempenho dos diferentes métodos serão discutidos, de modo a validar os resultados teóricos e a comparar as abordagens estudadas.

Capítulo 2

O problema de *carpooling*

A prática de *carpooling*, também designado por *ride-sharing*, ou *lift-sharing* refere-se à partilha de um veículo privado, pertencente a um dos intervenientes que se deslocam sob a forma de “boleia”. Reúnem-se no mesmo automóvel utilizadores que efetuam ou pretendam efetuar pelo menos uma parte dum mesmo itinerário nas suas viagens. As despesas de deslocação são partilhadas entre todos os utilizadores do mesmo veículo.

As vantagens do *carpooling* são significativas, pois o facto de existirem mais pessoas a utilizar um veículo reduz, nomeadamente, os custos (combustível, portagens, etc.) da viagem de cada pessoa. Este conceito é também uma forma de transporte mais sustentável do ponto de vista ambiental, reduzindo as emissões de gases poluentes, como o dióxido de carbono, o congestionamento do tráfego motorizado nas estradas, em particular em hora de ponta, e os problemas de estacionamento, por exemplo nos centros das cidades. Do ponto de vista social pode, também, destacar-se a redução do *stress* da condução e uma melhoria da segurança rodoviária, através da redução da probabilidade de acidentes. O reforço do espírito de equipa, através do estímulo à socialização entre os colaboradores, e a melhoria da utilização do estacionamento privado nas instalações, através da redução das necessidades, representam algumas das vantagens inerentes às empresas e instituições.

Contudo, sob o ponto de vista do utilizador, existem também algumas desvantagens tais como perda da independência e rigidez do sistema, em particular no que diz respeito às horas das viagens, indução de *stress* devido ao compromisso de horários ou de presença, e perda de intimidade na partilha do veículo privado com outras pessoas. Estas desvantagens, como é natural, causam alguma insatisfação.

Nesta forma de viajar, as reservas e consultas de percursos partilhados disponíveis efetuam-se mediante inscrição num grupo de *carpooler*, através da *internet*, com vertentes públicas e privadas. Depois de encontrar um transporte adequado, entram em contato para organizar os detalhes (custos, pontos de encontro, etc.) e, desta

forma, a viagem ficar definida. O serviço de *carpooling* é comumente implementado para deslocções curtas, mas é cada vez mais utilizado para viagens longas, com a formalidade e a regularidade das modalidades a variar entre os regimes e as viagens. Este serviço tem sido implementado em muitas empresas, instituições e serviços. A nível europeu os clubes de *carpoolers* começam a ter uma expressão significativa em várias cidades. Para mais informações sobre *carpooling* ver, por exemplo, [1, 3, 11].

2.1. Formulação como um programa linear inteiro

Consideremos uma rede (V, E) , onde cada nó de V corresponde a um veículo (e ao seu condutor) e cada arco $(i, j) \in E$ corresponde à possibilidade da pessoa i viajar no veículo j . Suponhamos, sem perda de generalidade, que a rede é completa, isto é, $E = V \times V$. Cada viagem está associada a um indivíduo que, ou conduz o próprio veículo, ou se desloca como passageiro com algum outro condutor. Cada arco $(i, j) \in E$ está associado a um peso $w_{ij} \in \mathbb{R}$, que representa o grau de insatisfação da pessoa i viajar de boleia com o condutor j (ou seja, no veículo j). Cada veículo j também tem uma capacidade máxima de pessoas que pode levar, incluindo o próprio condutor, denotada por $c_j \in \mathbb{N}$.

De acordo com Hartman [11], o problema de *carpooling* pode ser descrito por um programa linear binário, com as seguintes variáveis de decisão:

- y_i , para $i \in V$, com o valor 1 se e só se o condutor i utiliza o seu próprio veículo e 0 caso contrário.
- x_{ij} , para $i, j \in V$, com valor 1 se e só se o indivíduo i viaja no veículo j e 0 caso contrário.

O problema pode então ser formulado por:

$$\min z_1(x, y) = \sum_{(i,j) \in E} w_{ij} x_{ij} \quad (2.1)$$

$$\text{s. a } \sum_{(i,j) \in E, j \neq i} x_{ij} + y_i = 1, \quad i \in V \quad (2.2)$$

$$x_{ij} - y_j \leq 0, \quad (i, j) \in E \wedge i \neq j \quad (2.3)$$

$$x_{ii} - y_i = 0, \quad i \in V \quad (2.4)$$

$$\sum_{(i,j) \in E} x_{ij} - c_j y_j \leq 0, \quad j \in V \quad (2.5)$$

$$x_{ij} \in \{0, 1\}, \quad y_j \in \{0, 1\}, \quad (i, j) \in E, \quad j \in V \quad (2.6)$$

2.1 Formulação como um programa linear inteiro

Relativamente aos pesos, assume-se que quem leva o veículo não se importa de o fazer, enquanto que os restantes passageiros têm uma opinião formada sobre quem leva o veículo. Assim, w_{ij} representa o grau de insatisfação da pessoa $i \in V$ viajar à boleia com a pessoa $j \in V$. Nestas condições, admite-se ainda que $w_{ii} = 0$, para $i \in V$. Na formulação apresentada, a função a minimizar, ou seja, a função objetivo z_1 , representa a insatisfação total. Podemos relacionar esta insatisfação com as desvantagens, anteriormente referidas, para o utilizador.

As restrições (2.2) indicam que todos os intervenientes do *carpooling* participam da viagem, duma forma ou doutra. Mais concretamente, o indivíduo $i \in V$ pode participar deslocando-se no próprio veículo ($x_{ii} = 1$), ou no de outro interveniente ($x_{ij} = 1$, para algum $j \in V - \{i\}$).

As restrições (2.3) e (2.4) indicam que se alguma pessoa viajar num veículo, então este participará da viagem. Adicionalmente, um veículo participa da viagem se e só se o seu condutor seguir nesse mesmo veículo.

As restrições (2.5) indicam que o número de pessoas que viaja num veículo é limitado pela capacidade deste.

Por fim, as restrições (2.6) indicam que as variáveis de decisão representam escolhas binárias.

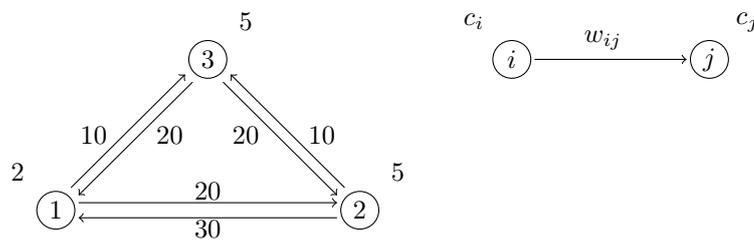


Figura 2.1: Rede (V, E)

Como exemplo de um tal problema de *carpooling* consideremos a rede representada na Figura 2.1, uma rede completa constituída por três veículos. Como referido anteriormente, considerar-se-á $w_{ii} = 0$, para todo o $i \in V$. O modelo linear associado

a este problema é

$$\begin{aligned} \min \quad & 20x_{12} + 10x_{13} + 30x_{21} + 10x_{23} + 20x_{31} + 20x_{32} \\ \text{s. a} \quad & x_{i1} + x_{i2} + x_{i3} + y_i = 1, \quad i = 1, 2, 3 \end{aligned} \quad (2.7)$$

$$x_{21} - y_1 \leq 0 \quad (2.8)$$

$$x_{31} - y_1 \leq 0 \quad (2.9)$$

$$x_{12} - y_2 \leq 0 \quad (2.10)$$

$$x_{32} - y_2 \leq 0 \quad (2.11)$$

$$x_{13} - y_3 \leq 0 \quad (2.12)$$

$$x_{23} - y_3 \leq 0 \quad (2.13)$$

$$x_{ii} - y_i = 0, \quad i = 1, 2, 3 \quad (2.14)$$

$$x_{1i} + x_{2i} + x_{3i} - 2y_i \leq 0, \quad i = 1, 2, 3 \quad (2.15)$$

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33}, y_1, y_2, y_3 \in \{0, 1\} \quad (2.16)$$

Como as variáveis de decisão deste programa linear são binárias, no que se segue, as soluções obtidas apresentam apenas aquelas cujo valor é 1.

Com base na rede que representa o problema, observa-se que a solução com a menor insatisfação consiste em cada indivíduo viajar no próprio veículo. Esta situação corresponde à solução $\{x_{11}, x_{22}, x_{33}, y_1, y_2, y_3\}$ e tem valor objetivo 0. Este resultado é pouco interessante na prática, dado que não implica qualquer tipo de boleia. Esta situação dá-se, pois este modelo é redutor no que toca a representar o problema de *carpooling*. Veremos adiante tentativas mais realistas de o fazer.

2.2. Formulações como programas lineares bicritério

Como vimos anteriormente, a solução ótima do problema na Figura 2.1, a de menor insatisfação, não implica “boleias”. Por forma a ter um modelo que reflita verdadeiramente o espírito do *carpooling*, apresentaremos duas formulações alternativas do problema, que incluem, cada uma, uma função objetivo adicional. Essas formulações apresentam duas funções objetivo conflituosas e que estão de acordo com as vantagens e desvantagens anteriormente mencionadas.

O primeiro problema é denominado problema de *carpooling* com insatisfação e número de veículos mínimos (PCIVM). Neste caso pretende-se minimizar duas

funções objetivo, $z_1(x, y)$, como definida em (2.1), e

$$z_2(x, y) = \sum_{j \in V} y_j.$$

A função objetivo z_2 representa o número de veículos participantes da viagem.

O segundo problema é denominado problema de *carpooling* com insatisfação e emissões mínimas (PCIEM). Aqui admite-se que, além dos dados anteriores, relativamente à insatisfação, também são conhecidos valores associados à emissão de gases poluentes de cada veículo, e_i , $i \in V$. Pretende-se então minimizar $z_1(x, y)$, tal como definida em (2.1), e

$$z_3(x, y) = \sum_{j \in V} e_j y_j,$$

que representa a quantidade de gases poluentes emitidos.

As restrições de PCIVM e de PCIEM continuam a ser (2.2)–(2.6). É ainda de notar que PCIVM é um caso particular de PCIEM, quando se admite que todos os veículos são igualmente poluentes, mais concretamente, que $e_i = k$ para todo o $i \in V$, com k constante.

Retomando o exemplo ilustrado na Figura 2.1, se se pretender minimizar apenas o número de veículos utilizados, isto é, z_2 , obtêm-se três soluções alternativas, todas elas envolvendo um só veículo. Nomeadamente,

- $\{x_{11}, x_{21}, x_{31}, y_1\}$, utilizando o veículo 1, com valores objetivo (50, 1);
- $\{x_{12}, x_{22}, x_{32}, y_2\}$, utilizando o veículo 2, com valores objetivo (40, 1);
- $\{x_{13}, x_{23}, x_{33}, y_3\}$, utilizando o veículo 3, com valores objetivo (20, 1).

Intuitivamente a última destas duas soluções é melhor do que as duas primeiras. Contudo, a solução ótima relativamente a z_1 tem valores objetivo (0, 3). Este exemplo torna evidente que as funções z_1 e z_2 são conflituosas, na medida em que não existe uma solução que as otimize simultaneamente. Uma abordagem bicritério permitirá, então, estender o conceito de otimalidade a problemas com duas funções objetivo, como os definidos anteriormente.

2.3. Conceitos de Programação Linear Biobjetivo

Consideremos o seguinte programa linear:

$$\begin{aligned} \min \quad & z_1(x) \\ \min \quad & z_2(x) \\ \text{s. a} \quad & x \in S \end{aligned} \tag{2.17}$$

com $x \in \mathbb{R}^n$, $z = (z_1, z_2)$ um par de funções lineares e onde o conjunto S é definido por restrições lineares e designado por região admissível. Uma vez que envolve duas funções objetivo, diz-se que este problema é um programa linear biobjetivo (PLBO). Se as variáveis são todas inteiras o problema diz-se um programa linear inteiro biobjetivo (PLIBO).

Uma solução que satisfaça todas as restrições do problema, designa-se por solução admissível do problema. Na presença de várias soluções admissíveis para um problema pretende-se encontrar a melhor, isto é, a solução admissível que tem o valor mais favorável da função objetivo. Esta solução é designada por solução ótima. Contudo, em geral, quando existem vários objetivos o conceito de otimalidade não é aplicável. Como referido acima, nem sempre uma solução admissível que otimiza uma das funções objetivo otimiza também as restantes. Deste modo, é necessário definir uma forma de comparar soluções admissíveis de um PLBO, [6, 14]. Consideremos, então, as seguintes definições.

Definição 1 (Solução dominada, Soluções equivalentes). Sejam $x = (x_1, \dots, x_n)$ e $y = (y_1, \dots, y_n)$ duas soluções admissíveis de (2.17). A solução y é dominada pela solução x se e só se

$$z_i(x) \leq z_i(y), \quad \forall i \in \{1, 2\}$$

As soluções x e y dizem-se equivalentes quando

$$z_i(x) = z_i(y), \quad \forall i \in \{1, 2\}$$

Definição 2 (Solução eficiente, Ponto não dominado). Uma solução admissível $x = (x_1, \dots, x_n)$ de (2.17) diz-se eficiente se não existir qualquer outra solução $y = (y_1, \dots, y_n) \in S$, tal que y domine x . Neste caso, $z(x)$ diz-se um ponto não dominado.

Definição 3 (Solução fracamente eficiente). Uma solução admissível $x = (x_1, \dots, x_n)$ de (2.17) denomina-se por fracamente eficiente se e só se não existe uma outra solução $y = (y_1, \dots, y_n) \in S$, tal que $z_i(x) < z_i(y)$, para todo o $i \in \{1, 2\}$.

Portanto, o conceito de solução ótima da Programação Linear é substituído pelo de “conjunto de soluções eficientes”, quando lidamos com mais do que uma função objetivo. As soluções eficientes também são designadas por soluções não inferiores ou ótimas de Pareto. O cálculo de soluções ótimas alternativas em otimização monoobjetivo é geralmente moroso e, por vezes, pouco útil. O mesmo pode acontecer

com o cálculo de soluções equivalentes quando existem várias funções objetivo. Por este motivo é comum a determinação de um subconjunto de soluções eficientes que não contém soluções equivalentes, assim como a associação entre solução eficiente e ponto não dominado.

No exemplo anterior, a solução $\{x_{13}, x_{23}, x_{33}, y_3\}$, de imagem $(z_1(x, y), z_2(x, y)) = (20, 1)$, domina a solução $\{x_{12}, x_{22}, x_{32}, y_2\}$, de imagem $(z_1(x, y), z_2(x, y)) = (40, 1)$, e esta última é uma solução fracamente eficiente. Além disso, utilizando dois veículos, o PCIVM tem as seguintes soluções:

- $\{x_{11}, x_{31}, x_{22}, y_1, y_2\}$, com valores objetivo $(20, 2)$;
- $\{x_{11}, x_{32}, x_{22}, y_1, y_2\}$, com valores objetivo $(20, 2)$;
- $\{x_{11}, x_{21}, x_{33}, y_1, y_3\}$, com valores objetivo $(30, 2)$;
- $\{x_{11}, x_{23}, x_{33}, y_1, y_3\}$, com valores objetivo $(10, 2)$;
- $\{x_{12}, x_{22}, x_{33}, y_2, y_3\}$, com valores objetivo $(20, 2)$;
- $\{x_{13}, x_{22}, x_{33}, y_2, y_3\}$, com valores objetivo $(10, 2)$.

É de notar que $\{x_{11}, x_{31}, x_{22}, y_1, y_2\}$, $\{x_{11}, x_{32}, x_{22}, y_1, y_2\}$ e $\{x_{12}, x_{22}, x_{33}, y_2, y_3\}$ são soluções equivalentes. Além disso, das soluções listadas acima destacam-se $\{x_{11}, x_{23}, x_{33}, y_1, y_3\}$ e $\{x_{13}, x_{22}, x_{33}, y_2, y_3\}$, por dominarem as restantes.

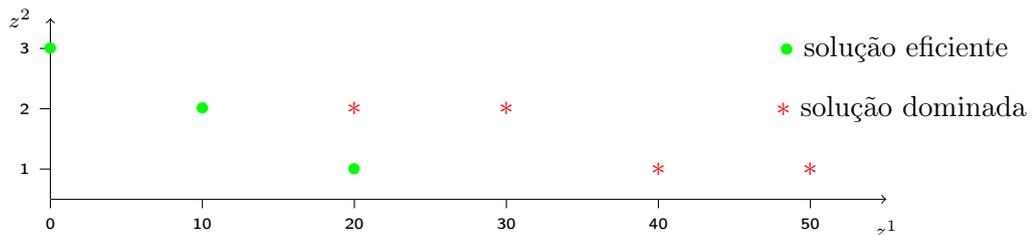


Figura 2.2: Imagens das soluções eficientes do PCIVM na rede da Figura 2.1

Finalmente, $\{x_{11}, x_{22}, x_{33}, y_1, y_2, y_3\}$ é a solução do PCIVM quando se utilizam três veículos e tem valores objetivo $(0, 3)$. Esta solução não é dominada por nenhuma das anteriores.

Após esta análise exaustiva, conclui-se que o PCIVM considerado tem quatro soluções eficientes: $\{x_{11}, x_{22}, x_{33}, y_1, y_2, y_3\}$, $\{x_{11}, x_{23}, x_{33}, y_1, y_3\}$, $\{x_{13}, x_{22}, x_{33}, y_2, y_3\}$ e $\{x_{13}, x_{23}, x_{33}, y_3\}$. Os pontos não dominados são $(0, 3)$, $(10, 2)$, $(20, 1)$, e correspondem aos círculos na Figura 2.2.

Quando o programa linear biobjetivo tem variáveis inteiras, podemos ainda distinguir dois tipos de soluções eficientes: suportadas e não suportadas.

Definição 4 (Solução eficiente suportada e solução eficiente não suportada). Uma solução eficiente de um PLBO diz-se suportada se é solução ótima de um problema monoobjetivo, em que a função objetivo resulta de uma soma ponderada das funções objetivo do PLBO. A solução diz-se não suportada, caso contrário.

Adiante serão dados mais detalhes sobre estes dois tipos de soluções. No caso do exemplo anterior todas as soluções eficientes são suportadas.

Capítulo 3

Resolução de problemas de *carpooling* bicritério

O presente capítulo incidirá em métodos para calcular as soluções eficientes de problemas lineares inteiros com duas funções objetivo, sem necessidade de introdução de informação adicional, ditos métodos geradores. Em geral tais processos consistem em reduzir o problema biobjetivo a outro da mesma classe mas em que se pretende otimizar uma única função objetivo. Este é o caso dos métodos descritos em seguida, e que são alguns dos mais utilizados: o método NISE, o método das restrições, e o método de Tchebycheff.

3.1. Método NISE

Provavelmente o processo mais intuitivo, e mais generalizado, de transformar duas funções objetivo numa só resulta de agregá-las, adicionando-as. Se existir um método para resolver um programa linear monoobjetivo, então será possível otimizar a função agregada. Além disso, quando é conhecida a importância relativa das funções objetivo para o problema que se pretende resolver, cada uma é ponderada em conformidade antes da agregação. Com ou sem ponderação o novo problema é redutor relativamente ao original, uma vez que tem uma solução única, ou várias mas com o mesmo valor objetivo. No entanto, essa é uma solução eficiente do primeiro problema, como mostra o Teorema 1.

Teorema 1 ([6]). *Se x^* é solução de*

$$\begin{aligned} \min \quad & \lambda_1 z_1(x) + \lambda_2 z_2(x) \\ \text{s. a} \quad & x \in S \\ & \lambda_1 + \lambda_2 = 1 \\ & \lambda_i > 0, \quad i \in \{1, 2\} \end{aligned}$$

então x^ é uma solução eficiente do problema (2.17).*

Demonstração. Proceda-se por redução ao absurdo. Seja x^* uma solução satisfazendo as condições do teorema e suponha-se que não é uma solução eficiente. Então existe outra solução \bar{x} que domina x^* , ou seja, tal que

$$z_i(\bar{x}) \leq z_i(x^*), \quad i = 1, 2, \quad \text{e} \quad z_k(\bar{x}) < z_k(x^*), \quad \text{para algum } k.$$

Como $\lambda_1, \lambda_2 > 0$, tem-se

$$\lambda_i z_i(\bar{x}) \leq \lambda_i z_i(x^*), \quad i = 1, 2, \quad \text{e} \quad \lambda_k z_k(\bar{x}) < \lambda_k z_k(x^*),$$

o que contradiz a otimalidade de x^* . □

Este resultado garante que as soluções de problemas como o enunciado no Teorema 1 são eficientes¹. No entanto, não especifica que valores λ_i utilizar, por forma a obter todas as soluções eficientes do problema. Poder-se-ia optar pela resolução de problemas para uma escolha arbitrária de valores, mas tal processo não garantiria a obtenção do conjunto completo das soluções eficientes suportadas do problema.

O método *NonInferior Solutions Estimation* (NISE), proposto por Cohon [7], efetua a otimização de uma sequência de problemas obtidos de (2.17) mas com funções objetivo que são uma soma pesada das duas funções objetivo, considerando uma determinada escolha de pesos (λ_i). Os problemas desta sequência agregam as duas funções objetivo iniciais. Os novos problemas mantêm a estrutura do problema inicial, mas têm uma única função objetivo, pelo que conhecendo-se um método para os resolver será possível encontrar soluções eficientes do primeiro problema.

O método NISE começa por determinar as duas soluções, x_A e x_B , que minimizam isoladamente cada uma das funções objetivo z_1 e z_2 , respetivamente. Uma vez que estes problemas podem ter soluções múltiplas, algumas correspondentes a soluções fracamente eficientes, e portando dominadas, as soluções x_A e x_B devem corresponder às soluções lexicograficamente ótimas relativamente a (z_1, z_2) e a (z_2, z_1) , respetivamente. Pode verificar-se que x_A e x_B são soluções eficientes.

No que se segue, A denotará o ponto pertencente ao espaço das imagens das soluções associado à solução x_A , isto é, $A = z(x_A)$. A próxima solução eficiente a ser determinada tem como imagem o ponto mais distante do segmento de reta \overline{AB} , de forma a pertencer à região admissível e ao semi-plano definido por AB que contém a origem. Assim, o segmento de reta \overline{AB} servirá de base ao cálculo da próxima solução

¹Por definição, essas soluções serão suportadas.

eficiente. Com efeito, consideram-se os pesos λ_1 e λ_2 tais que

$$-\frac{\lambda_1}{\lambda_2} = \alpha,$$

em que α é o declive da reta AB . Como no cálculo dos pesos apenas interessa a relação entre eles, e sendo conhecido o declive da reta AB , basta igualar um dos pesos a 1 para se obter o outro. Com efeito,

$$\alpha = \frac{z_2(x_B) - z_2(x_A)}{z_1(x_B) - z_1(x_A)}.$$

Tomando $\lambda_2 = 1$, obtém-se

$$-\frac{\lambda_1}{1} = \alpha = \frac{z_2(x_B) - z_2(x_A)}{z_1(x_B) - z_1(x_A)} \implies \lambda_1 = \frac{z_2(x_A) - z_2(x_B)}{z_1(x_B) - z_1(x_A)}$$

O problema a resolver é então

$$\begin{aligned} \min \quad & \lambda_1 z_1(x) + z_2(x) \\ \text{s. a } & x \in S \end{aligned} \tag{3.1}$$

e, pelo Teorema 1, a sua solução ótima, x_C , é solução eficiente do problema (2.17). Cada novo ponto assim obtido forma duas novas retas com os pontos adjacentes, AC e BC . Este processo é repetido até que nenhum programa linear apresente apenas novas soluções. O método NISE é resumido no Algoritmo 1.

Algorithm 1: Método NISE

```

1  $x_A \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_1, z_2)$ 
2  $x_B \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_2, z_1)$ 
3  $L \leftarrow \{A, B\}; X \leftarrow \{(A, B)\}$ 
4 while  $X \neq \emptyset$  do
5      $(A, B) \leftarrow$  elemento de  $X; X \leftarrow X - \{(A, B)\}$ 
6      $\lambda_1 \leftarrow \frac{z_2(x_A) - z_2(x_B)}{z_1(x_B) - z_1(x_A)}$ 
7      $Y \leftarrow$  lista das imagens das soluções ótimas de (3.1)
8      $hasNew \leftarrow$  false
9     for  $C \in Y$  do
10         if  $C \notin L$  then
11              $hasNew \leftarrow (C = \text{último elemento de } Y)$ 
12              $L \leftarrow L \cup \{C\}$ 
13     if  $hasNew$  then
14          $X \leftarrow X \cup \{(A, \text{primeiro elemento de } Y)\} \cup \{(\text{último elemento de } Y, B)\}$ 

```

No Algoritmo 1, a variável L representa um conjunto que armazena os pontos não dominados encontrados (poder-se-ia guardar as soluções eficientes propriamente ditas de modo análogo) e X é um conjunto auxiliar que armazena pares de pontos não dominados determinados e ainda por analisar. A variável Y representa uma lista auxiliar que armazena as imagens das soluções de (3.1), em cada iteração. Enquanto que L contém todos os pontos não dominados, Y é substituída, em cada iteração, pelos pontos não dominados do segmento de reta considerado nessa iteração.

Os programas lineares resolvidos ao longo do Algoritmo 1, incluindo os iniciais, podem ter múltiplas soluções ótimas. Neste caso, como se trata de um problema linear, as soluções obtidas estarão situadas sobre o mesmo segmento de reta (associado à imagem da função objetivo). Ocorrendo esta situação no cálculo dos pontos iniciais, A e B , quando se consideram as retas verticais e horizontais, respetivamente, só um dos extremos é eficiente. A verificação passa por comparar as imagens das soluções obtidas e verificar qual delas produz um ponto não dominado.

No caso de ocorrer após o cálculo daqueles pontos, todas as soluções no segmento de reta correspondem a soluções eficientes do PLBO original e deverão ser consideradas para continuar o algoritmo.

Como referido anteriormente, por definição, o método NISE encontra apenas soluções eficientes suportadas. Por este motivo, é frequente a resolução de problemas biobjetivo em duas fases: uma primeira em que se utiliza o método NISE para determinar as soluções eficientes suportadas; e uma segunda em que é aplicado outro método que permite encontrar as soluções eficientes não suportadas [15]. Os métodos descritos em seguida têm uma aplicação mais ampla do que o método NISE, dado que permitem calcular todas as soluções eficientes (suportadas e não suportadas).

3.2. Método das restrições

Outra forma conhecida para “reduzir” um problema biobjetivo a um problema mono-objetivo é otimizar apenas uma das funções objetivo e incluir a restante (as restantes, no caso multiobjetivo geral) como uma restrição, limitando-a por um valor constante. A nova restrição pode alterar a estrutura do problema, por exemplo em problemas de otimização combinatoria, e dificultar a sua resolução. No entanto, de um modo geral, em Programação Linear, ou Programação Linear Inteira, continua a ser possível resolver o novo problema, que tem um único objetivo. Tal como sucedeu na secção anterior, as soluções assim obtidas são soluções eficientes do problema inicial,

resultado apresentado no Teorema 2.

Teorema 2 ([6]). *Seja x^* a solução ótima única, para algum k , de*

$$\begin{aligned} \min \quad & z_k(x) \\ \text{s. a} \quad & x \in S \\ & z_i(x) \leq w_i, \quad i \neq k \end{aligned}$$

Então x^ é uma solução eficiente do problema (2.17).*

Demonstração. Proceda-se por redução ao absurdo. Seja x^* uma solução satisfazendo as condições do enunciado, ou seja, $x^* \in S$, $z_i(x^*) \leq w_i$, $i \neq k$, e

$$z_k(x^*) \leq z_k(x), \forall x \in S.$$

Suponha-se que x^* não é uma solução eficiente. Então existe outra solução de (2.17), \bar{x} , que domina x^* . Isto significa que $\bar{x} \in S$ e

$$z_i(\bar{x}) \leq z_i(x^*) \leq w_i, \quad i \neq k,$$

pelo que \bar{x} é solução admissível do problema enunciado. Além disso,

$$z_k(\bar{x}) \leq z_k(x^*),$$

donde \bar{x} é uma solução ótima daquele problema, alternativa a x^* , ou x^* não é solução ótima do problema. Qualquer dos casos contraria as hipóteses iniciais. Sendo assim, x^* tem de ser eficiente. \square

O método das restrições foi proposto em 1971, por Haimes, Lasdon e Wismer [10]. Baseado no resultado acima, o método consiste em otimizar uma função objetivo e restringir a restante, atribuindo-lhe limites máximos (w_i). De um modo geral os limites considerados são escolhidos de forma uniforme, no entanto, pode não ser garantida a obtenção de todas as soluções eficientes do problema.

O caso em que a solução do problema no enunciado do Teorema 2 não é única não invalida o método. De acordo com a demonstração, temos a garantia que existe pelo menos uma solução eficiente e consideramo-la. Dadas as soluções, comparamo-las de forma a verificar qual delas é eficiente. O problema tem sempre soluções no âmbito que estamos a considerar no algoritmo, como será explicado no que se segue.

Sem perda de generalidade, vai ser escolhida a função objetivo z_1 para otimizar, sendo z_2 a função objetivo a restringir. Admitindo que as funções objetivo tomam apenas valores inteiros², o método das restrições pode servir-se do seguinte resultado.

Teorema 3. *Sejam x^* uma solução eficiente de (2.17) e \bar{x} uma solução ótima de*

$$\begin{aligned} \min \quad & z_1(x) & (3.2) \\ \text{s. a.} \quad & x \in S \\ & z_2(x) \leq w_2 \end{aligned}$$

com $w_2 = z_2(x^*) - 1$. *Nestas condições, não existe nenhuma solução eficiente x tal que $z_2(\bar{x}) < z_2(x) < z_2(x^*)$.*

Demonstração. Sejam x^* e \bar{x} duas soluções nas condições do enunciado. Por absurdo, suponhamos que existe uma solução eficiente de (2.17), x , tal que $z_2(\bar{x}) < z_2(x) < z_2(x^*)$. Por hipótese, $z_2(x) < z_2(x^*)$, logo $z_2(x) \leq w_2$ e x é solução admissível de (3.2). Como \bar{x} é solução ótima deste problema temos igualmente $z_1(\bar{x}) \leq z_1(x)$, donde concluímos que x não pode ser solução eficiente, o que contraria a hipótese. \square

O Teorema 3 fornece uma forma de atualizar w_i , ao resolver programas lineares com imagens inteiras e duas funções objetivo através do método das restrições. O valor inicial de w_2 será $z_2(x_A) - 1$. Caso o problema (3.2) tenha múltiplas soluções ótimas, consideramos aquela que é solução eficiente do problema (2.17) como próxima solução deste método. Seja x_C esta solução. O próximo valor de w_2 será, então, dado por $z_2(x_C) - 1$. Procedemos deste modo até que $z_2(x_C) \leq z_2(x_B)$, caso em que o algoritmo termina. O processo descrito é resumido no Algoritmo 2.

Tal como anteriormente, a variável L representa um conjunto que armazena os pontos não dominados encontrados ao longo do Algoritmo 2.

3.3. Método de Tchebycheff

Apresentamos agora o método de Tchebycheff. À semelhança dos anteriores, este processo calcula soluções eficientes de (2.17), resolvendo uma sequência de problemas monoobjetivo relacionados com o original. Começemos por introduzir alguns conceitos, essenciais para uma melhor compreensão daquele método.

²É o que acontece nos problemas de *carpooling* introduzidos no Capítulo 2 se os coeficientes forem inteiros.

Algorithm 2: Método das restrições

```

1  $x_A \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_1, z_2)$ 
2  $x_B \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_2, z_1)$ 
3  $L \leftarrow \{A\}$ 
4  $x_C \leftarrow x_A$ 
5 while  $z_2(x_C) > z_2(x_B)$  do
6    $w_2 \leftarrow z_2(x_C) - 1$ 
7    $x_C \leftarrow$  solução ótima de (3.2) eficiente para (2.17)
8    $L \leftarrow L \cup \{C\}$ 

```

Definição 5 (Região admissível no espaço das funções objetivo). A região admissível no espaço das funções objetivo pode ser definida da seguinte forma:

$$Z = \{\bar{z} \in \mathbb{R}^2 : \bar{z} = (z_1(x), z_2(x)), x \in S\}$$

Definição 6 (Solução ideal). A solução ideal z^* é o ponto no espaço dos objetivos, cujas componentes são o ótimo de cada função objetivo na região admissível, quando otimizadas separadamente.

Em geral, a solução ideal, z^* , não pertence a Z . Se tal acontecesse, z^* otimizaria simultaneamente todas as funções objetivo, o que significaria que as funções não estariam em conflito.

A ideia base do método de Tchebycheff é utilizar um ponto de referência, possivelmente escolhido pelo agente de decisão e frequentemente representado por q . Uma vez fixado o ponto de referência, pretende-se calcular soluções eficientes com imagens próximas desse ponto, através da minimização de uma distância a esse ponto. Deste modo, utilizando a família de métricas L_β e sendo a solução ideal, z^* , o ponto de referência, o problema de encontrar uma solução eficiente de (2.17) consiste em

$$\begin{aligned} \min \quad & \|z(x) - z^*\|_\beta \\ \text{s. a} \quad & x \in S \end{aligned}$$

Tal como no método NISE, se se admitir que as diferenças em cada objetivo têm diferentes importâncias, consideramos uma métrica L_β^λ ponderada, com $\lambda \in \mathbb{R}^2$, $\lambda \geq 0$.

$$\begin{aligned} \min \quad & \|z(x) - z^*\|_\beta^\lambda \\ \text{s. a} \quad & x \in S \end{aligned}$$

onde a norma L_β^λ ponderada, com $\lambda \in \mathbb{R}^2$, $\lambda \geq 0$ de x é definida por

$$\|x\|_\beta^\lambda = (|\lambda_1 x_1|^\beta + |\lambda_2 x_2|^\beta)^{\frac{1}{\beta}}$$

O parâmetro β permite especificar a métrica utilizada, e assim captar diferentes formas de escolher as soluções. Para $\beta = 1$, métrica denominada de Manhattan, são considerados os desvios das funções objetivo em relação a z^* , de forma diretamente proporcional. Para $2 < \beta < \infty$ maiores desvios têm maior importância. Por fim, para $\beta = \infty$, métrica designada de Tchebycheff, apenas o maior desvio, que por conseguinte é o pior, é tido em conta. Além do tipo de métrica a utilizar, a forma como são encontradas as soluções é determinada pelo ponto de referência e os pesos associados à métrica, [5].

No método que aqui apresentamos será utilizada a métrica de Tchebycheff. Para obter a importância relativa dos diferentes objetivos, são utilizados vetores de pesos $\lambda \in \bar{\Lambda}$, onde

$$\bar{\Lambda} = \left\{ \lambda \in \mathbb{R}^2 : \lambda_1, \lambda_2 \geq 0, \sum_{i=1}^2 \lambda_i = 1 \right\},$$

e as métricas ponderadas de Tchebycheff, definidas como

$$\|z - z^*\|_{\infty}^{\lambda} = \max_{i=1,2} \{\lambda_i |z_i - z_i^*|\}.$$

Em alternativa podemos medir a distância entre z e z^* , utilizando métricas ponderadas e aumentadas de Tchebycheff,

$$\|z - z^*\|_{\infty}^{\lambda} = \|z - z^*\|_{\infty}^{\lambda} + \rho \sum_{i=1}^2 z_i,$$

ou

$$\|z - z^*\|_{\infty}^{\lambda} = \max_{i=1,2} \{\lambda_i (z_i - z_i^*)\} + \rho \sum_{i=1}^2 z_i$$

com $\rho > 0$ suficientemente pequeno. O termo acrescentado tem como objetivo evitar a determinação de soluções ótimas alternativas, o que poderia conduzir a soluções fracamente eficientes do problema biobjetivo inicial. No entanto, se ρ não for pequeno o suficiente, poderão não ser encontradas todas as soluções eficientes. O cálculo de valores de ρ adequados a cada problema é uma questão complexa. Algumas condições para a sua determinação são deduzidas em [8]. No Capítulo 4 encontra-se um exemplo que ilustra as possíveis situações. O método de Tchebycheff apoia-se no seguinte teorema.

Teorema 4 ([6]). *Se x^* é solução de*

$$\begin{aligned} \min \quad & \|z(x) - z^*\|_{\infty}^{\lambda} \\ \text{s. a} \quad & x \in S \\ & \lambda \in \bar{\Lambda} \end{aligned} \tag{3.3}$$

então x^* é uma solução eficiente do problema (2.17).

A solução ideal é de extrema importância. Todavia, neste texto, o nosso interesse incidirá num ponto de referência mais geral, q , ao qual se minimizará uma distância de Tchebycheff para encontrar soluções eficientes de (2.17). Consideremos (3.3) com o ponto de referência q :

$$\begin{aligned} \min \quad & \|z(x) - q\|_\infty^\lambda \\ \text{s. a} \quad & x \in S \\ & \lambda \in \bar{\Lambda} \end{aligned} \tag{3.4}$$

Apesar de a função objetivo de (3.3) ser não linear, se assumirmos que o ponto de referência q se encontra abaixo da região admissível (isto é, $z_i(x) > q_i$, para i), o problema pode ser transformado num programa linear da seguinte forma:

$$\begin{aligned} \min \quad & v + \rho \sum_{i=1}^2 z_i(x) \\ \text{s. a} \quad & \lambda_i(z_i(x) - q_i) \leq v \quad i = 1, 2 \\ & v \geq 0 \\ & x \in S \\ & \lambda \in \bar{\Lambda} \end{aligned} \tag{3.5}$$

Deste modo, consideremos a versão seguinte do Teorema 4.

Teorema 5 ([6]). *Se x^* é solução de (3.5) então x^* é uma solução eficiente do problema (2.17).*

Demonstração. Consideremos que x^* é uma solução ótima de (3.5), para a qual $v = v^*$, e suponhamos, por redução ao absurdo, que x^* não é solução eficiente de (2.17). Então, existe outra solução \bar{x} que domina x^* , ou seja,

$$z_i(\bar{x}) \leq z_i(x^*), \quad i = 1, 2, \quad \text{e} \quad z_k(\bar{x}) < z_k(x^*), \quad \text{para algum } k.$$

Assim, temos igualmente $\lambda_i(z_i(\bar{x}) - q_i) \leq v^*$, $i = 1, 2$, e $\rho \sum_{i=1}^2 z_i(\bar{x}) < \rho \sum_{i=1}^2 z_i(x^*)$, o que significa que \bar{x} é solução admissível de (3.5) e, portanto, x^* não é solução ótima, pelo que chegamos a uma contradição. \square

Note-se que na demonstração anterior, não foi usado o facto de $v \geq 0$. Daqui inferimos que o ponto de referência pode estar, quer abaixo da fronteira de Pareto

(caso em que $v \geq 0$), quer acima da fronteira de Pareto (caso em que $v < 0$), que continuamos a obter soluções eficientes de (2.17). Contudo, nos algoritmos que se seguem, vão ser considerados apenas pontos de referência abaixo da fronteira de Pareto, pelo que podemos dizer que os algoritmos continuam a servir-se da “distância” de Tchebycheff.

Os resultados anteriores motivam a utilização da métrica de Tchebycheff para obter todas as soluções eficientes de um problema biobjetivo. Os algoritmos assim obtidos, conhecidos como método de Tchebycheff [6], resolvem uma sequência de problemas semelhantes a (3.5), podendo variar os pesos ou o ponto de referência. Em seguida descrevemos três variantes deste método.

A primeira versão do método de Tchebycheff baseia-se em pesquisas direcionais. Numa pesquisa direcional altera-se, em cada iteração, apenas a componente do ponto de referência correspondente à função objetivo que se pretende melhorar naquele momento. O ponto de referência é atualizado de forma automática e calcula-se, assim, a solução eficiente “mais próxima” do ponto de referência atualizado segundo a direção seguida [4]. Esta pesquisa por soluções diz-se direcional, pois seguimos na direção de minimizar uma das funções objetivo. À distância entre um ponto de referência e uma atualização deste, damos o nome de *passo* da pesquisa direcional. Se a pesquisa se iniciar no ótimo de uma das funções e a outra função for escolhida para melhorar, o método varre as soluções eficientes entre as lexicograficamente ótimas relativas a (z_1, z_2) e (z_2, z_1) , desde que o passo da pesquisa direcional seja pequeno o suficiente. O resultado está resumido no Algoritmo 3. Tal como os métodos apresentados anteriormente, este começa por determinar x_A e x_B , soluções lexicograficamente ótimas relativamente a (z_1, z_2) e (z_2, z_1) , respetivamente. Os pesos são constantes, $\lambda = (1, 1)$. A próxima solução eficiente é solução ótima do problema (3.5), tendo A como ponto de referência (sem perda de generalidade). Foi escolhido 1 para passo da pesquisa direcional. Como nos encontramos perante um problema com soluções inteiras, este valor garante que sejam encontradas todas as soluções eficientes de (2.17). Em seguida, atualiza-se o ponto de referência, melhorando o segundo objetivo, isto é, decrementando a segunda coordenada e resolve-se novamente o problema (3.5). Se a solução, x , não foi já encontrada, a sua imagem é armazenada e o ponto de referência é atualizado como $z(x)$. O algoritmo termina quando a solução calculada for igual a x_B .

Na segunda e na terceira versões do método de Tchebycheff os valores dos pesos

Algorithm 3: Método de Tchebycheff

```

1  $x_A \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_1, z_2)$ 
2  $x_B \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_2, z_1)$ 
3  $L \leftarrow \{A\}$ 
4  $q \leftarrow A$ 
5 do
6    $q \leftarrow (q_1, q_2 - 1)$ 
7    $x \leftarrow$  solução ótima de (3.5), com  $\lambda = (1, 1)$ 
8   if  $z(x) \notin L$  then
9      $q \leftarrow z(x)$ 
10     $L \leftarrow L \cup \{z(x)\}$ 
11 while  $z(x) \neq B$ 

```

e o ponto de referência são atualizados em cada iteração. De forma semelhante ao método NISE, em [9], e mais recentemente em [13], a atualização destes parâmetros é feita com base em duas soluções eficientes que sejam adjacentes. Sejam x_A e x_B duas dessas soluções. Tendo em conta que a métrica de Tchebycheff implica a determinação de soluções cujas imagens se posicionam sobre um rectângulo, toma-se como ponto de referência o ponto (A_1, B_2) . Os pesos são escolhidos de modo a formar rectângulos com lados proporcionais ao definido por A e B ,

$$\lambda_1 = A_2 - B_2, \quad \lambda_2 = B_1 - A_1,$$

ou valores normalizados destes. A solução ótima, x_C , calculada nestas condições forma dois novos pares de soluções eficientes, (A, C) e (C, B) , que são analisados em iterações futuras. O processo termina quando não existirem novos pares de soluções eficientes por analisar. O Algoritmo 4, que denominamos por método de Tchebycheff com aplicação binária, resume esta versão do método de Tchebycheff.

Uma forma alternativa ao Algoritmo 4 para procurar soluções eficientes no rectângulo definido por x_A e x_B , soluções lexicograficamente eficientes relativamente a (z_1, z_2) e a (z_2, z_1) , respetivamente, consiste em executar um “varrimento” daquela região, resolvendo uma sequência de problemas (3.5) em que se mantém fixo o ponto B e se atualiza o ponto de referência e os pesos dependentes de B e das sucessivas soluções encontradas. Os varrimentos são repetidos até toda a região de pontos não dominados ter sido explorada. O Algoritmo 5, que denominamos por método de Tchebycheff com aplicação sequencial, resume o método descrito.

Algorithm 4: Método de Tchebycheff com aplicação binária

```

1  $x_A \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_1, z_2)$ 
2  $x_B \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_2, z_1)$ 
3  $L \leftarrow \{A, B\}; X \leftarrow \{(A, B)\}$ 
4 while  $X \neq \emptyset$  do
5    $(A, B) \leftarrow$  par em  $X; X \leftarrow X - \{(A, B)\}$ 
6    $\lambda_1 \leftarrow A_2 - B_2; \lambda_2 \leftarrow B_1 - A_1$ 
7    $q \leftarrow (A_1, B_2)$ 
8    $\bar{x} \leftarrow$  solução ótima de (3.5)
9   if  $z(\bar{x}) \notin L$  then
10      $L \leftarrow L \cup \{z(\bar{x})\}$ 
11      $X \leftarrow X \cup \{(A, z(\bar{x}))\} \cup \{(z(\bar{x}), B)\}$ 

```

Algorithm 5: Método de Tchebycheff com aplicação sequencial

```

1  $x_A \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_1, z_2)$ 
2  $x_B \leftarrow$  solução lexicograficamente ótima relativamente a  $(z_2, z_1)$ 
3  $L \leftarrow \{A, B\}; X \leftarrow \{(A, B)\}$ 
4 while  $X \neq \emptyset$  do
5    $(A, B) \leftarrow$  par em  $X; X \leftarrow X - \{(A, B)\}$ 
6    $C \leftarrow A$ 
7   while  $C \neq B \wedge C \neq C'$  do
8      $C' \leftarrow C$ 
9      $\lambda_1 \leftarrow C_2 - B_2; \lambda_2 \leftarrow B_1 - C_1$ 
10     $q \leftarrow (C_1, B_2)$ 
11     $\bar{x} \leftarrow$  solução ótima de (3.5)
12    if  $z(\bar{x}) \notin L$  then
13       $L \leftarrow L \cup \{z(\bar{x})\}$ 
14       $X \leftarrow X \cup \{(C, z(\bar{x}))\}$ 
15       $C \leftarrow z(\bar{x})$ 

```

Capítulo 4

Aplicação dos métodos

Como exemplo, em seguida, mostrar-se-á a aplicação dos métodos apresentados no Capítulo 3, para calcular as soluções eficientes de casos particulares do PCIVM e do PCIEM.

4.1. Método NISE

Começamos por resolver os problemas que resultam de considerar cada objetivo individualmente. Ao otimizar apenas z_1 , obtém-se a solução única $\{x_{11}, x_{22}, x_{33}, y_1, y_2, y_3\}$, que tem por imagem $A = (0, 3)$.

Minimizando a função objetivo z_2 , obtêm-se três soluções: $\{x_{11}, x_{21}, x_{31}, y_1\}$, $\{x_{12}, x_{22}, x_{32}, y_2\}$ e $\{x_{13}, x_{23}, x_{33}, y_3\}$, com imagens $(50, 1)$, $(40, 1)$ e $(20, 1)$, respectivamente. Dado que se trata de um problema de minimização, a solução eficiente corresponde a $B = (20, 1)$.

O declive da reta que passa pelos pontos A e B , encontrados no passo anterior, é dado por:

$$\alpha = \frac{1 - 3}{20 - 0} = -\frac{1}{10}$$

pelo que se calculam os pesos

$$\lambda_1 = -\frac{1}{10}, \quad \lambda_2 = 1$$

Minimizando agora $\lambda_1 z_1 + z_2$, na região admissível original, obtêm-se quatro soluções ótimas. Duas correspondem aos pontos A e B . As restantes duas são $\{x_{11}, x_{23}, x_{33}, y_1, y_3\}$ e $\{x_{13}, x_{22}, x_{33}, y_2, y_3\}$ que correspondem ambas a um novo ponto $C = (10, 2)$. Por fim, o algoritmo termina, pois $X = \emptyset$. O conjunto das soluções eficientes obtido é, portanto,

$$\{\{x_{11}, x_{22}, x_{33}, y_1, y_2, y_3\}, \{x_{13}, x_{23}, x_{33}, y_3\}, \{x_{13}, x_{22}, x_{33}, y_2, y_3\}, \{x_{11}, x_{23}, x_{33}, y_1, y_3\}\},$$

cujas imagens são as representadas na Figura 2.2.

Consideremos agora um PCIVM com dimensão 10, em que todos os veículos têm capacidade para quatro pessoas, e com a matriz de insatisfação W_1 apresentada em

apêndice. Neste caso, o conjunto das imagens das soluções eficientes encontradas, através do método NISE, é dado por

$$\{(0, 10), (1, 9), (3, 8), (7, 7), (14, 6), (26, 5), (54, 3)\},$$

ilustrado na Figura 4.1a.

Consideremos agora um PCIEM com dimensão 10, com a matriz de insatisfação e as capacidades referidas no exemplo anterior. Admitamos ainda que as emissões dos veículos são $e = [2, 2, 4, 4, 4, 5, 5, 5, 5, 5]$. Através do método NISE, obtém-se o seguinte conjunto das imagens das soluções eficientes:

$$\{(0, 41), (128, 10), (46, 15), (14, 25), (75, 11), (4, 32), (26, 20), (2, 36), (11, 27)\}.$$

4.2. Método das restrições

Consideremos, agora, a aplicação do método das restrições para resolução do mesmo problema. Tal como no método NISE, começamos por determinar os ótimos relativamente a z_1 e relativamente a z_2 . Como anteriormente, $A = (0, 3)$ e $B = (20, 1)$. O conjunto de soluções eficientes já encontradas é então inicializado com $L = \{(0, 3)\}$.

Seguindo o Algoritmo 2, o próximo problema a resolver é

$$\begin{aligned} \min \quad & z_1(x, y) = 20x_{12} + 10x_{13} + 30x_{21} + 10x_{23} + 20x_{31} + 20x_{32} \\ \text{s. a} \quad & (2.7) - (2.16) \\ & y_1 + y_2 + y_3 \leq 2 \end{aligned}$$

e obtêm-se as soluções ótimas $\{x_{13}, x_{22}, x_{33}, y_2, y_3\}$ e $\{x_{11}, x_{23}, x_{33}, y_1, y_3\}$. Ambas têm imagem $(10, 2)$, pelo que são eficientes. Atualizemos $L = L \cup \{(10, 2)\}$. Em seguida, considera-se

$$\begin{aligned} \min \quad & z_1(x, y) = 20x_{12} + 10x_{13} + 30x_{21} + 10x_{23} + 20x_{31} + 20x_{32} \\ \text{s. a} \quad & (2.7) - (2.16) \\ & y_1 + y_2 + y_3 \leq 1 \end{aligned}$$

que tem solução ótima única $\{x_{13}, x_{23}, x_{33}, y_3\}$, com imagem $(20, 1)$. Atualizamos $L = L \cup \{(20, 1)\}$. Consideremos, agora,

$$\begin{aligned} \min \quad & z_1(x, y) = 20x_{12} + 10x_{13} + 30x_{21} + 10x_{23} + 20x_{31} + 20x_{32} \\ \text{s. a} \quad & (2.7) - (2.16) \\ & y_1 + y_2 + y_3 \leq 0 \end{aligned}$$

Dado que não tem soluções, o algoritmo termina. Deste modo, obtemos

$$L = \{(0, 3), (10, 2), (20, 1)\}.$$

Neste caso, o conjunto das soluções eficientes calculadas coincide com o conjunto obtido pelo método NISE.

Consideremos agora o mesmo PCIVM que utilizámos no método NISE, com dimensão 10. Neste caso, por aplicação do método das restrições, o conjunto das imagens das soluções é dado por:

$$\{(0, 10), (1, 9), (3, 8), (7, 7), (14, 6), (26, 5), (41, 4), (54, 3)\},$$

apresentado na Figura 4.1b.

Note-se que a solução com imagem $(41, 4)$ foi encontrada apenas com o método das restrições. Significa, portanto, que é uma solução eficiente não suportada. Gráficamente, esta situação pode ser observada através das chamadas zonas de desnível de dualidade.

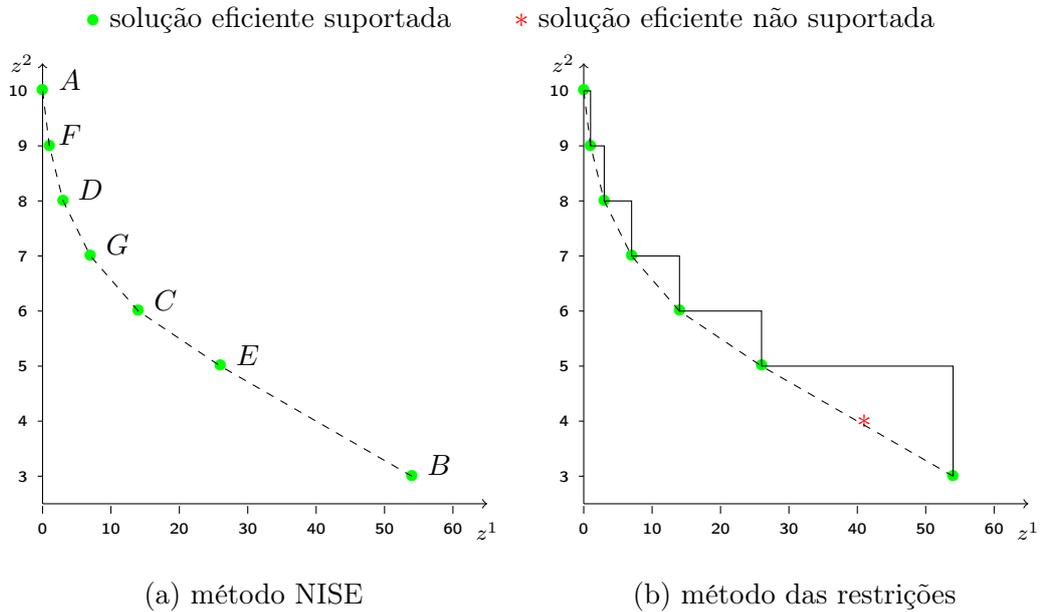


Figura 4.1: Imagens das soluções eficientes para o problema com dimensão 10

Definição 7 (Zona de desnível de dualidade). Uma zona de desnível de dualidade é definida pelo triângulo formado pelas imagens de duas soluções eficientes suportadas adjacentes e pelo ponto definido pelos valores máximos das duas funções objetivo, relativamente àquelas soluções.

As soluções eficientes não suportadas só podem aparecer em zonas de desnível de dualidade, as quais estão representadas na Figura 4.1b, [15]. Esta também proporciona uma melhor visualização e comparação entre os métodos NISE e das restrições.

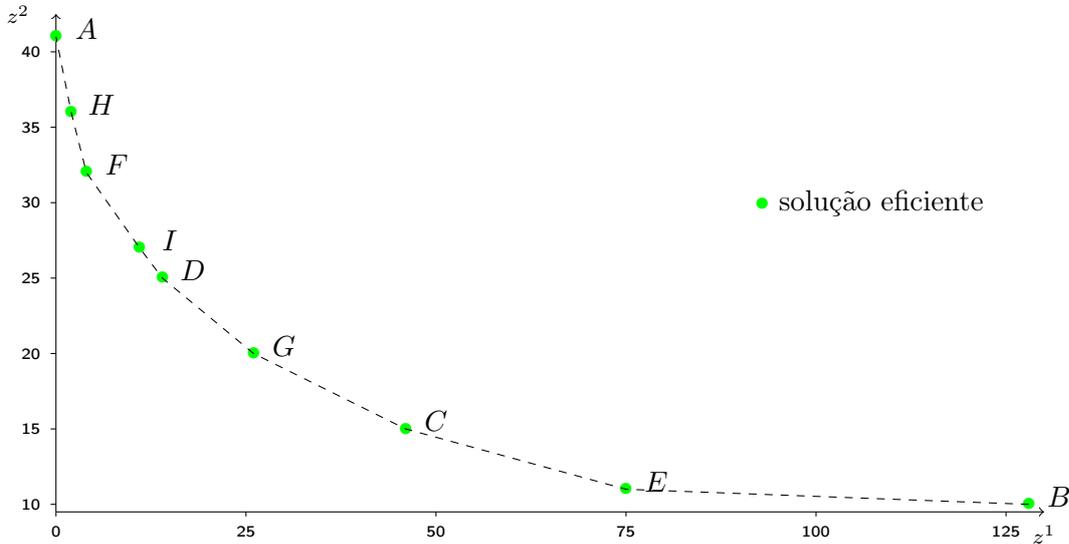


Figura 4.2: Imagens das soluções eficientes para o PCIEM, obtidas através do método NISE

Consideremos agora o mesmo PCIEM que utilizámos no método NISE, com dimensão 10. Para além das imagens das soluções eficientes suportadas, obtidas através do método NISE, o método das restrições calculou ainda as imagens:

$$\{(70, 13), (54, 14), (34, 19), (23, 22), (22, 24), (7, 30), (3, 35), (2, 36), (1, 39)\}.$$

Tal como no exemplo anterior, estes últimos pontos correspondem a soluções eficientes não suportadas, que aparecem dentro das zonas de desnível de dualidade na Figura 4.3.

4.3. Método de Tchebycheff

Em seguida, mostrar-se-á a aplicação do método de Tchebycheff, para calcular as soluções eficientes do PCIVM na rede representada na Figura 2.1.

Começamos, por escrever o problema (3.5) e tomar $\rho = 0,001$ e $\lambda = (1, 1)$. Em seguida, resolvemos os problemas que resultam de considerar cada objetivo individualmente. Ao otimizar apenas z_1 obtém-se a solução $\{x_{11}, x_{22}, x_{33}, y_1, y_2, y_3\}$, que tem por imagem $A = (0, 3)$. Minimizando a função objetivo z_2 obtém-se a solução $\{x_{13}, x_{23}, x_{33}, y_3\}$, que tem por imagem $B = (20, 1)$. Seguindo o Algoritmo 3, iniciamos o nosso conjunto de soluções eficientes encontradas, com $L = \{(0, 3)\}$. Em

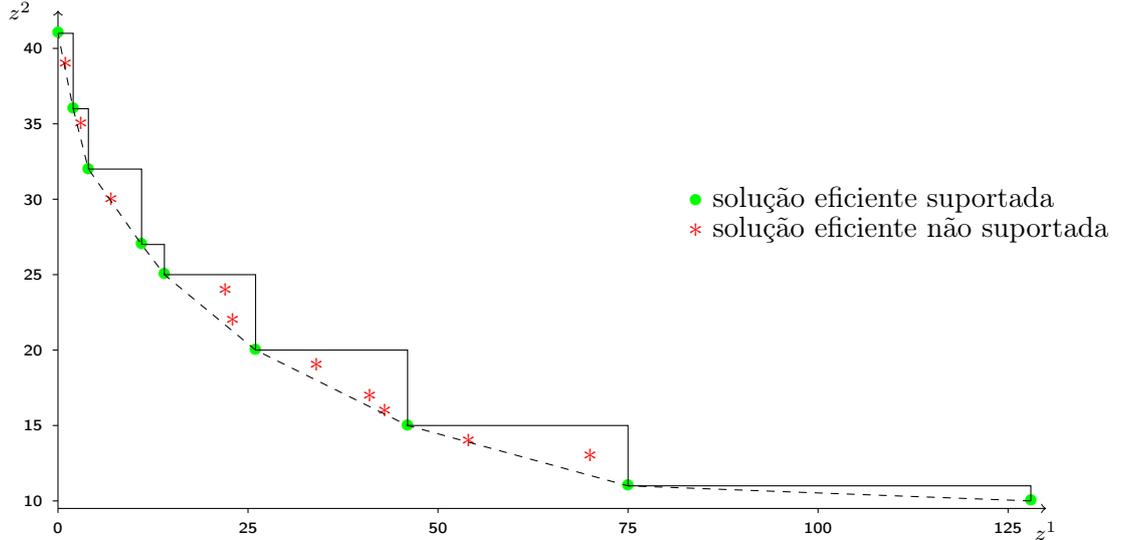


Figura 4.3: Imagens das soluções eficientes para o PCIEM, obtidas através do método das restrições

seguida, fixamos o ponto de referência $q = (0, 3)$. Vamos proceder de forma cíclica até que a última solução eficiente encontrada corresponda a $(20, 1)$. Nas primeiras 10 iterações, onde fixamos $q_1 = 0$ e decrementamos q_2 , foi resolvido o problema (3.5), obtendo sempre a mesma solução, com imagem $(0, 3)$. Ao resolver o problema (3.5) com $q = (0, -8)$, foi obtida a primeira solução não armazenada em L , cuja imagem corresponde a $(10, 2)$. Assim, atualizamos $L = L \cup \{(10, 2)\}$ e escolhemos $q = (10, 2)$ para novo ponto de referência. Também nas próximas 10 passagens do ciclo, onde fixamos $q_1 = 10$ e decrementamos q_2 , foi resolvido o problema (3.5), obtendo sempre a mesma solução, com imagem $(10, 2)$. Ao resolver o problema (3.5) com $q = (10, -9)$, foi obtida a solução com imagem $(20, 1)$, correspondente à solução x_B , terminando assim o ciclo e, conseqüentemente, o algoritmo. Todas as soluções eficientes correspondem então às do conjunto L , dado por

$$L = \{(0, 3), (20, 1), (10, 2)\}.$$

Consideremos agora o PCIEM com dimensão 10 utilizado nos métodos anteriores. Começamos por escrever o problema (3.5), e consideramos $\rho = 0,001$ e $\lambda = (1, 1)$. Em seguida, resolvemos os problemas que resultam de considerar cada objetivo individualmente. Ao otimizar apenas z_1 , obtém-se a solução x_A que tem por imagem $A = (0, 10)$. Minimizando a função objetivo z_2 , obtém-se a solução x_B , que tem por imagem $B = (54, 3)$. De forma análoga ao problema descrito anteriormente, seguindo

o Algoritmo 3, encontrou-se o seguinte conjunto de soluções eficientes ordenadas pela ordem encontrada:

$$L = \{(0, 10), (54, 3), (1, 9), (3, 8), (7, 7), (14, 6), (26, 5), (41, 4)\},$$

o mesmo que obtido pelo método das restrições, representado na Figura 4.1b.

Para terminar, consideremos o PCIEM com dimensão 10 referido nos métodos anteriores. Começamos por escrever o problema (3.5), e consideramos $\rho = 0,001$ e $\lambda = (1, 1)$. Resolvem-se os problemas que resultam de considerar cada função objetivo de forma individual. Ao otimizar apenas z_1 , obtém-se a solução x_A que tem por imagem $A = (0, 41)$. Minimizando a função objetivo z_2 , obtém-se a solução x_B , que tem por imagem $B = (128, 10)$. Seguindo o Algoritmo 3 encontrou-se o seguinte conjunto de soluções eficientes, ordenadas pela ordem encontrada:

$$L = \{(0, 41), (128, 10), (1, 39), (2, 36), (3, 35), (4, 32), (7, 30), (11, 27), (14, 25), (22, 24), \\ (23, 22), (26, 20), (34, 19), (41, 17), (43, 16), (46, 15), (54, 14), (70, 13), (75, 11)\},$$

o mesmo obtido pelo método das restrições e que se encontra representado na Figura 4.3.

4.4. Método de Tchebycheff com aplicação binária

Em seguida, mostrar-se-á a aplicação do método de Tchebycheff com aplicação binária, para calcular as soluções eficientes do PCIVM na rede representada na Figura 2.1.

Começamos por escrever o problema (3.5), com $\rho = 0,001$. Em seguida, resolvemos os problemas que resultam de considerar cada objetivo individualmente. Ao otimizar apenas z_1 , obtém-se a solução $\{x_{11}, x_{22}, x_{33}, y_1, y_2, y_3\}$, que tem por imagem $A = (0, 3)$. Minimizando a função objetivo z_2 , obtém-se a solução $\{x_{13}, x_{23}, x_{33}, y_3\}$, que tem por imagem $B = (20, 1)$. Seguindo o Algoritmo 4, iniciamos o nosso conjunto de soluções eficientes encontradas, com $L = \{(0, 3), (20, 1)\}$, e o nosso conjunto auxiliar com $X = \{((0, 3), (20, 1))\}$. Vamos proceder de forma cíclica até que X seja vazio. Retiramos o primeiro par de X , $((0, 3), (20, 1))$. Calculamos o valor de $\lambda_1 = 3 - 1 = 2$ e $\lambda_2 = 20 - 0 = 20$. Fixamos o ponto de referência $q = (0, 1)$. É resolvido o problema (3.5) obtendo a solução $(10, 2)$. Como esta não pertence a L , este passa a ser $L = \{(0, 3), (20, 1), (10, 2)\}$ e X passa a ser $X = \{((10, 2), (20, 1)), ((0, 3), (10, 2))\}$. Na iteração seguinte é removido de X o par $((10, 2), (20, 1))$ e é obtida a solução $(10, 2)$, que já pertence a L . Na seguinte iteração

é removido de X o par $((0, 3), (10, 2))$ e é obtida a solução $(0, 3)$ que já pertence a L . Neste momento X encontra-se vazio terminando assim o ciclo e, conseqüentemente, o algoritmo. Todas as soluções eficientes correspondem então às do conjunto L , dado por

$$L = \{(0, 3), (20, 1), (10, 2)\}.$$

Consideremos agora o PCIVM com dimensão 10 utilizado nos métodos anteriores. Começamos, primeiro, por escrever o problema (3.5), e consideremos $\rho = 0,001$. Em seguida, resolvemos os problemas que resultam de considerar cada objetivo individualmente. Ao otimizar apenas z_1 , obtém-se a solução x_A que tem por imagem $A = (0, 10)$. Minimizando a função objetivo z_2 , obtém-se a solução x_B , que tem por imagem $B = (54, 3)$. De forma análoga ao problema descrito anteriormente, seguindo o Algoritmo 4, encontra-se o seguinte conjunto de soluções eficientes ordenadas pela ordem encontrada:

$$L = \{(0, 10), (54, 3), (14, 6), (26, 5), (41, 4), (3, 8), (7, 7), (1, 9)\},$$

o mesmo obtido pelo método das restrições, representado na Figura 4.1b.

Para terminar, consideremos o PCIVM com dimensão 10 referido nos métodos anteriores. Começamos por escrever o problema (3.5), com $\rho = 0,001$. Resolvem-se os problemas que resultam de considerar cada função objetivo de forma individual. Ao otimizar apenas z_1 , obtém-se a solução x_A que tem por imagem $A = (0, 41)$. Minimizando a função objetivo z_2 , obtém-se a solução x_B , que tem por imagem $B = (128, 10)$. Seguindo o Algoritmo 4 encontra-se o seguinte conjunto de soluções eficientes, ordenadas pela ordem encontrada:

$$L = \{(0, 41), (128, 10), (34, 19), (70, 13), (75, 11), (46, 15), (54, 14), (41, 17), (43, 16), \\ (11, 27), (23, 22), (26, 20), (14, 25), (22, 24), (4, 32), (7, 30), (2, 36), (3, 35), (1, 39)\},$$

o mesmo obtido anteriormente – Figura 4.3.

4.5. Método de Tchebycheff com aplicação sequencial

Em seguida, mostrar-se-á a aplicação do método de Tchebycheff com aplicação sequencial, para calcular as soluções eficientes do PCIVM na rede representada na Figura 2.1.

Começamos por escrever o problema (3.5) e consideramos $\rho = 0,001$. Em seguida, resolvemos os problemas que resultam de considerar cada objetivo individualmente. Ao otimizar apenas z_1 , obtém-se a solução $\{x_{11}, x_{22}, x_{33}, y_1, y_2, y_3\}$, que

tem por imagem $A = (0, 3)$. Minimizando a função objetivo z_2 , obtém-se a solução $\{x_{13}, x_{23}, x_{33}, y_3\}$, que tem por imagem $B = (20, 1)$. Seguindo o Algoritmo 5, iniciamos o nosso conjunto de soluções eficientes encontradas, com $L = \{(0, 3), (20, 1)\}$, e o nosso conjunto auxiliar com $X = \{((0, 3), (20, 1))\}$. Vamos proceder de forma cíclica até que X seja vazio. Retiramos o primeiro par de X , $((0, 3), (20, 1))$, fazemos $C = (0, 3)$ e procedemos de forma cíclica até $C = (20, 1)$ ou $C = C'$. Guardamos o valor atual de C em C' , portanto C' passa a ser $(0, 3)$. Calculamos os pesos $\lambda_1 = 3 - 1 = 2$ e $\lambda_2 = 20 - 0 = 20$. Fixamos o ponto de referência $q = (0, 1)$. Resolvemos o problema (3.5), obtendo a solução $(10, 2)$. Como esta não pertence a L , este é atualizado como $L = \{(0, 3), (20, 1), (10, 2)\}$ e X com $X = \{((0, 3), (10, 2))\}$. C é atualizado com a solução encontrada $(10, 2)$. Na iteração seguinte, é encontrada novamente a solução $(10, 2)$, pelo que termina este ciclo interno. Na iteração seguinte do ciclo externo é removido de X o par $((0, 3), (10, 2))$ e, no interno é obtida a solução $(0, 3)$ que já pertence a L . Neste momento, $X = \emptyset$, terminando assim o ciclo e, conseqüentemente, o algoritmo. Todas as soluções eficientes correspondem então às do conjunto L , dado por

$$L = \{(0, 3), (20, 1), (10, 2)\}.$$

Consideremos agora o PCIVM com dimensão 10 utilizado nos métodos anteriores. Começamos por escrever o problema (3.5), com $\rho = 0,001$. Em seguida, resolvemos os problemas que resultam de considerar cada objetivo individualmente. Ao otimizar apenas z_1 , obtém-se a solução x_A que tem por imagem $A = (0, 10)$. Minimizando a função objetivo z_2 , obtém-se a solução x_B , que tem por imagem $B = (54, 3)$. De forma análoga ao problema descrito anteriormente, seguindo o Algoritmo 5, encontram-se os seguintes soluções eficientes ordenadas pela ordem encontrada:

$$L = \{(0, 10), (54, 3), (14, 6), (26, 5), (41, 4), (3, 8), (7, 7), (1, 9)\},$$

as mesmas obtidas pelo método das restrições – Figura 4.1b.

Para terminar, consideremos o PCIEM com dimensão 10 referido nos métodos anteriores. Começamos por escrever o problema (3.5), com $\rho = 0,001$. Resolvem-se os problemas que resultam de considerar cada função objetivo de forma individual. Ao otimizar apenas z_1 obtém-se a solução x_A , que tem por imagem $A = (0, 41)$. Minimizando a função objetivo z_2 obtém-se a solução x_B , que tem por imagem

$B = (128, 10)$. Seguindo o Algoritmo 5 encontram-se as seguintes soluções eficientes, ordenadas pela ordem encontrada:

$$L = \{(0, 41), (128, 10), (34, 19), (70, 13), (75, 11), (46, 15), (54, 14), (41, 17), (43, 16), \\ (11, 27), (23, 22), (26, 20), (14, 25), (22, 24), (4, 32), (7, 30), (2, 36), (3, 35), (1, 39)\},$$

as mesmas obtidas anteriormente – Figura 4.3.

Segue-se um exemplo do impacto da escolha do valor de ρ no caso da métrica de Tchebycheff aumentada. Este exemplo expõe a necessidade de uma escolha adequada do parâmetro ρ ao utilizar a métrica de Tchebycheff aumentada.

Consideremos um PCIEM com capacidades e emissões iguais às utilizadas na experiência computacional apresentada no Capítulo 5, e com a matriz de insatisfação W_2 em apêndice.

O método das restrições aplicado a este problema calcula 18 soluções,

$$\{(0, 41), (12, 36), (24, 32), (28, 31), (39, 28), (40, 27), (44, 26), (55, 23), \\ (56, 22), (72, 21), (74, 19), (75, 18), (88, 17), (99, 14), (114, 13), (128, 12), \\ (154, 11), (192, 10)\}.$$

Por sua vez, o método de Tchebycheff binário com $\rho = 0,001$ obtém 19 soluções,

$$\{(0, 41), (192, 10), (72, 21), (114, 13), (154, 11), (128, 12), (88, 17), (99, 14), \\ (90, 17), (75, 18), (74, 19), (28, 31), (44, 26), (55, 23), (56, 22), (39, 28), \\ (40, 27), (12, 36), (24, 32)\}$$

Verificamos que o valor $\rho = 0,001$ é demasiado pequeno, produzindo uma solução fracamente eficiente, não encontrada com o processo anterior.

Considerando agora o método de Tchebycheff binário com $\rho = 0,002$, obtemos as mesmas 18 soluções calculadas pelo método das restrições, ou seja, são obtidas todas as soluções eficientes e apenas estas.

Capítulo 5

Resultados computacionais

Neste capítulo apresentaremos os resultados de uma comparação computacional entre os métodos descritos anteriormente, quando aplicados ao PCIEM.

Os métodos foram implementados em Java 8u45 e os programas lineares inteiros foram resolvidos utilizando o pacote de *software* CPLEX 12.6.0.0, [2,12]. Os códigos desenvolvidos estão disponíveis em <http://www.mat.uc.pt/~mat1254/codigos/>. Os testes foram executados num Intel(R) Core(TM)2 Quad CPU Q9550, com velocidade de processador 2.83GHz (morsa6).

Para cada dimensão n do problema foram gerados 20 casos e aplicados os Algoritmos 1 – 5. Para avaliar a execução dos métodos foi registado o tempo de CPU efetivo utilizado na execução do processo, o número de subproblemas resolvidos (isto é, de chamadas do *solver* CPLEX), o número de soluções eficientes determinadas e o de subproblemas resolvidos por segundo.

Para que possa ser despistado qualquer tipo de enviesamento, a seleção dos dados da matriz de insatisfação, W , foi feita aleatoriamente, com entradas inteiras

$$\begin{cases} w_{ij} \in [a, b], & i \neq j \\ w_{ii} = 0 \end{cases}, \text{ para } i, j = 1, \dots, n,$$

com $a = 10$ e $b = 100$. Os valores das capacidades dos veículos, c , e das suas emissões, e , foram fixados por forma a se obterem situações realistas. Os diferentes métodos vão resolver o mesmo conjunto de problemas.

Sabemos que o transporte de objetos ou pessoas, num veículo, influencia a sua resistência aerodinâmica, sendo esta a principal responsável pelo aumento de consumos. Por exemplo, um peso adicional de 50 kg pode representar um aumento de consumo de combustível entre 1% e 3% e, conseqüentemente, um aumento das emissões de gases. Além disso, nos percursos urbanos o consumo pode ser ainda mais significativo devido às variações frequentes de velocidade. Para os problemas de *car-pooling* utilizados na experiência computacional, considerámos apenas uma categoria de veículos a motor, concebidos e construídos para o transporte de passageiros, com

5.0 Método de Tchebycheff com aplicação sequencial

eficientes suportadas e não suportadas. Relativamente aos métodos de Tchebycheff, as implementações utilizando aplicação binária e sequencial são as que apresentam melhores tempos médios.

Tabela 5.2: Número médio de soluções

Método \ n	10	20	30	40	50
NISE	9,15	15,15	21,20	28,80	34,05
Restrições	17,40	38,90	57,75	84,00	102,65
Tchebycheff	17,40	38,90	57,75	84,00	102,65
Tchebycheff binário	17,45	38,95	57,75	84,00	102,70
Tchebycheff sequencial	17,45	38,95	57,75	84,00	102,70

Tabela 5.3: Número médio de subproblemas resolvidos

Método \ n	10	20	30	40	50
NISE	49,25	77,75	94,00	114,80	117,65
Restrições	32,80	75,80	113,50	166,00	203,30
Tchebycheff	215,60	264,35	336,35	431,40	496,90
Tchebycheff binário	31,90	74,90	112,50	165,00	202,40
Tchebycheff sequencial	31,90	74,95	112,50	165,00	203,30

Tabela 5.4: Número de médios de subproblemas/seg.

Método \ n	10	20	30	40	50
NISE	108,72	38,73	21,91	12,17	8,16
Restrições	95,91	34,56	18,36	10,91	7,86
Tchebycheff	49,89	24,03	16,59	10,19	7,58
Tchebycheff binário	52,30	19,33	12,31	7,43	5,43
Tchebycheff sequencial	53,26	19,16	12,31	7,42	5,45

Observando a Tabela 5.2, verificamos, como já era expetável, que o método NISE

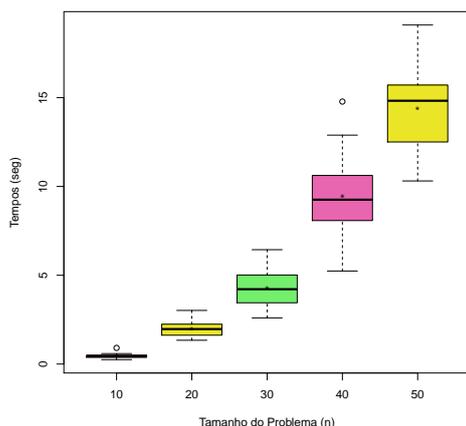


Figura 5.1: Tempos de execução: Método NISE

apresenta um menor número médio de soluções, dado que calcula apenas as soluções eficientes suportadas. Os métodos de Tchebycheff e das restrições apresentam exatamente o mesmo número médio de soluções. No geral, aqueles quando comparados com os métodos de Tchebycheff com aplicação binária e com aplicação sequencial também apresentam os mesmos valores. No entanto, para $n = 10, 20, 50$ nem todos os valores coincidem. Esta situação deve-se ao facto de o valor do ρ ser demasiado pequeno, para algumas matrizes. Apesar de ser possível escolher um novo ρ , como as matrizes são geradas de forma aleatória, não teríamos garantia de que o mesmo não voltaria a acontecer.

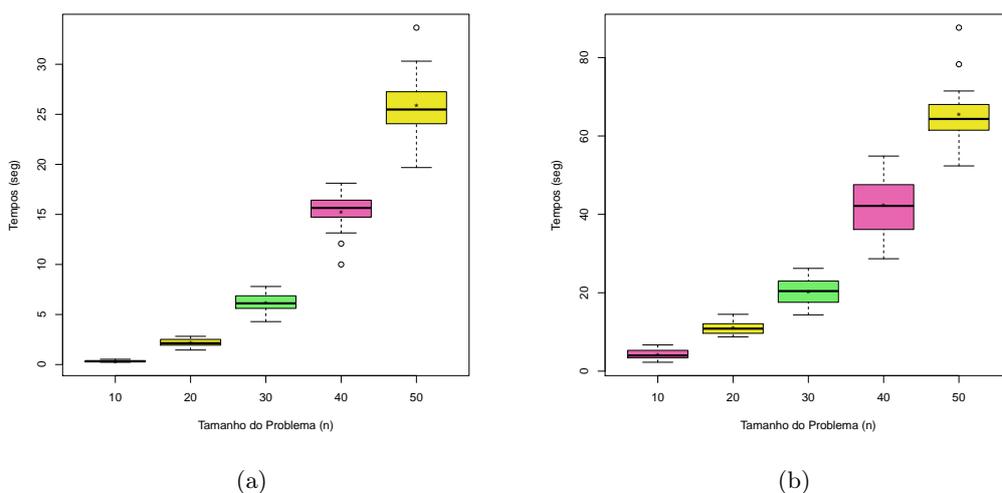


Figura 5.2: Tempos de execução: (a) Método das restrições; (b) Método de Tchebycheff

Podemos concluir que existe uma correlação entre o tamanho da fronteira de

Pareto e o tempo de execução.

Atendendo, agora, ao número médio de subproblemas resolvidos, nota-se uma discrepância acentuada no número de subproblemas considerado no método de Tchebycheff, quando comparado com os restantes métodos. O número médio de subproblemas dos métodos de Tchebycheff com aplicação binária e sequencial, e das restrições são bastante semelhantes, alguns resultados são mesmo iguais.

De acordo com a Tabela 5.4, o método NISE é aquele que permite a resolução de um maior número de subproblemas por segundo. Relativamente aos restantes métodos, destaca-se o método das restrições por ser, em média, o mais rápido.

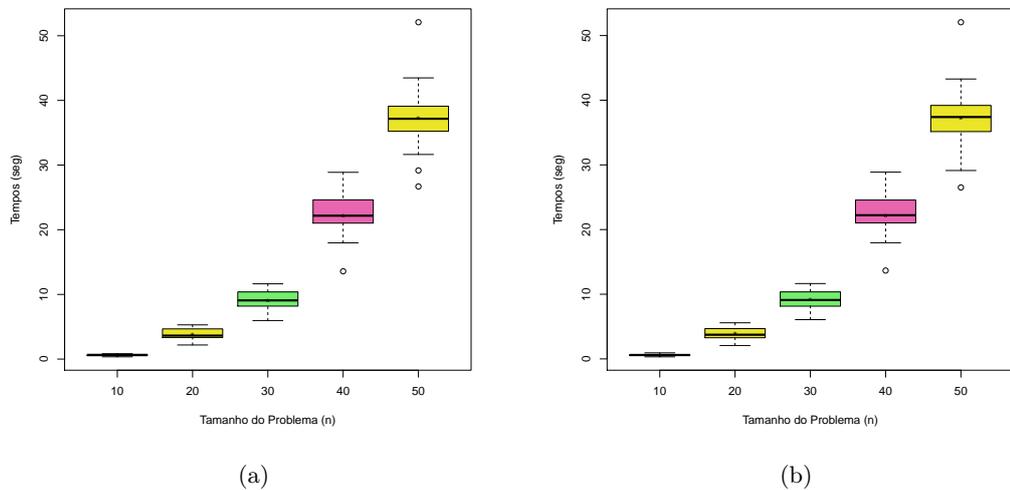


Figura 5.3: Tempos de execução: (a) Método de Tchebycheff com aplicação binária; (b) Método de Tchebycheff com aplicação sequencial

As Figuras 5.1 a 5.3 apresentam os gráficos de extremos e quartis, dos tempos médios de execução dos vários métodos quando aplicados ao PCIEM. Podemos, também, observar alguns valores atípicos, ou seja, *outliers*, que são representados através de uma circunferência.

Capítulo 6

Conclusão

Este trabalho possibilitou-nos o estudo de problemas de otimização linear inteira biobjetivo. Os métodos conhecidos para lidar com problemas de Programação Linear Inteira/Inteira-Mista Multiobjetivo podem classificar-se em métodos iterativos e em métodos geradores, sendo estes o objeto do nosso trabalho. Em particular, os métodos geradores foram aplicados a um problema de *carpooling* com duas funções objetivo. Por forma a ter um modelo realista, foram apresentadas duas formulações alternativas do problema, que incluíam, cada uma, uma função objetivo adicional. Assim sendo, considerámos uma formulação para o problema de *carpooling* com insatisfação e número de veículos mínimos; e um problema de *carpooling* com insatisfação e emissões de gases poluentes mínimas de cada veículo.

Apresentaram-se métodos geradores para problemas lineares biobjetivo que permitem o cálculo do conjunto de soluções eficientes sem a necessidade de introdução de informação adicional. Nomeadamente, o método NISE, o método das restrições, e o método de Tchebycheff. Depois de exemplificados os métodos para o problema de *carpooling*, estes foram testados por meio de experiência computacional, para o caso do problema de *carpooling* em que se pretendem minimizar a insatisfação dos passageiros e a emissão de gases poluentes. Os diferentes métodos resolveram o mesmo conjunto de problemas, para evitar enviesamentos. Os resultados médios recolhidos, em termos de tempos de execução, do número de soluções eficientes calculadas, do número de subproblemas resolvidos e do número de subproblemas resolvidos por segundo, sobre o desempenho dos diferentes métodos foram discutidos, de modo a validar os resultados teóricos e a comparar as abordagens estudadas.

Podemos concluir que os métodos geradores possibilitam o cálculo de todas as soluções eficientes sem, em cada iteração, necessitarem da interação do agente de decisão.

Em termos de tempo de execução, foi o método NISE que obteve o melhor desempenho em termos de tempo. Verificámos, no entanto, que este processo determina

apenas soluções eficientes suportadas. O método das restrições foi o que apresentou um melhor desempenho, comparativamente com os outros métodos que calculam todas as soluções eficientes (suportadas e não suportadas). Concluimos ainda que existe uma correlação entre o tamanho da fronteira de Pareto e o tempo de execução. Assim, diferentes problemas com a mesma dimensão, poderão apresentar diferentes tempos de execução. O número médio de subproblemas do método das restrições foi o menor de todos os métodos. Desta forma, o método das restrições “dominou” os outros métodos com igual capacidade a nível de desempenho.

Bibliografia

- [1] Carpooling. <http://www.en.wikipedia.org/wiki/Carpool>. Acedido: 2015-05-25.
- [2] Manual da API Java para o CPLEX. http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.1/ilog.odms.ide.help/refjavaopl/html/overview-summary.html. Acedido: 2015-05-25.
- [3] Transportes partilhados. http://www.conferenciamobilidade.imtt.pt/pacmob/transpartilhados/Transportes_Partilhados_Marco2011.pdf.
- [4] M. J. Alves and J. Clímaco. An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound. *European Journal of Operational Research*, 124:478–494, 2000.
- [5] M. J. Alves and J. P. Costa. Programação linear inteira e inteira-mista multiobjetivo: Conceitos fundamentais e métodos. In *Actas do Congresso Latino-Iberoamericano de Investigación Operativa*, pages 4713–4724, Rio de Janeiro, Brasil, September 2012. CLAIO/SBPO.
- [6] J. Clímaco, C. Antunes, and M. J Alves. *Programação Linear Multiobjetivo*. Imprensa da Universidade de Coimbra, 2003.
- [7] J. Cohon. *Multiobjective Programming and Planning*. Academic Press, New York, 1978.
- [8] K. Dachert, J. Gorski, and K. Klamroth. An augmented weighted tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. *Computers & Operations Research*, 39:2929–2943, May 2012.

- [9] P. Eswaran, A. Ravindran, and H. Moskowitz. Algorithms for nonlinear integer bicriterion problems. *Journal of Optimization Theory and Applications*, 63(2):261–279, 1989.
- [10] Y. Haimes, U. Lasdon, and D. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 1(3):296–297, 1971.
- [11] I. Hartman. Optimal assignment for carpooling - draft. <http://www.uhasselt.be/Documents/datasim/Papers/%28D3.1%29%281%29BAH13.pdf>, July 2013. Acedido: 2015-05-25.
- [12] IBM. IBM ILOG CPLEX Optimization Studio. <http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/>. Acedido: 2015-05-25.
- [13] T. Ralphs, M. Saltzman, and M. Wiecek. An improved algorithm for solving bi-objective integer programs. *Annals of Operations Research*, 147(1):43–70, 2006.
- [14] R. Steuer. *Multiple Criteria Optimization Theory, Computation and Application*. John Wiley, 1986.
- [15] M. Visée, J. Teghem, M. Pirlot, and E. Ulungu. Two-phases method and branch-and-bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, 12:139–155, 1998.

Apêndice A

Matrizes de insatisfação

Apresentam-se em seguida as matrizes W_1 e W_2 referidas no texto.

$$W_1 = \begin{bmatrix} 0 & 9 & 74 & 6 & 47 & 72 & 15 & 96 & 76 & 33 \\ 95 & 0 & 69 & 3 & 84 & 39 & 7 & 3 & 33 & 1 \\ 60 & 49 & 0 & 81 & 43 & 16 & 11 & 15 & 23 & 9 \\ 15 & 77 & 2 & 0 & 67 & 22 & 93 & 72 & 97 & 84 \\ 67 & 29 & 62 & 64 & 0 & 56 & 87 & 29 & 12 & 92 \\ 13 & 93 & 22 & 25 & 60 & 0 & 32 & 42 & 61 & 4 \\ 7 & 24 & 12 & 86 & 27 & 52 & 0 & 71 & 12 & 56 \\ 97 & 69 & 45 & 10 & 85 & 25 & 72 & 0 & 38 & 78 \\ 23 & 97 & 12 & 81 & 56 & 99 & 71 & 71 & 0 & 35 \\ 56 & 2 & 60 & 77 & 96 & 9 & 31 & 4 & 10 & 0 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0 & 32 & 42 & 72 & 20 & 37 & 91 & 29 & 82 & 50 \\ 89 & 0 & 85 & 56 & 31 & 81 & 71 & 60 & 78 & 59 \\ 54 & 15 & 0 & 59 & 83 & 17 & 23 & 51 & 81 & 72 \\ 93 & 48 & 23 & 0 & 47 & 12 & 78 & 52 & 33 & 34 \\ 32 & 89 & 42 & 16 & 0 & 83 & 22 & 75 & 90 & 19 \\ 48 & 24 & 82 & 37 & 79 & 0 & 37 & 40 & 50 & 32 \\ 69 & 63 & 16 & 62 & 74 & 64 & 0 & 53 & 17 & 25 \\ 57 & 92 & 80 & 29 & 69 & 40 & 84 & 0 & 87 & 16 \\ 72 & 21 & 16 & 60 & 33 & 12 & 20 & 94 & 0 & 32 \\ 27 & 36 & 99 & 95 & 24 & 73 & 48 & 18 & 78 & 0 \end{bmatrix}$$