

Marisol Ferreira Martins

INTRODUÇÃO À PROGRAMAÇÃO LINEAR MULTI OBJETIVO

Dissertação de Mestrado em Matemática, Área de Especialização em Estatística, Otimização e Matemática Financeira,
orientada pelo Professor Doutor José Luís Esteves dos Santos
e apresentada ao Departamento de Matemática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Julho 2017



UNIVERSIDADE DE COIMBRA

Introdução à programação linear multiobjetivo

Marisol Ferreira Martins



UNIVERSIDADE DE COIMBRA

Mestrado em Matemática

Master in Mathematics

Dissertação de Mestrado | MSc Dissertation

Julho 2017

Agradecimentos

Gostaria de começar por agradecer ao meu orientador Professor José Luis Santos por toda a ajuda fornecida ao longo deste semestre. Sem a sua ajuda e os conhecimentos por si transmitidos a realização deste trabalho não seria possível. Ao departamento agradeço as condições físicas e humanas que foram essenciais para o meu crescimento académico.

Agradeço também à minha família por todo o apoio disponibilizado, sem eles não conseguiria chegar até aqui.

Por fim, ao André agradeço todo o apoio demonstrado ao longo destes cinco anos, o seu apoio incondicional, o seu encorajamento e os seus ensinamentos foram muito importantes ao longo de todo o meu percurso académico e pessoal.

Resumo

A programação linear multiobjetivo é um caso particular de programação multiobjetivo, onde se otimiza simultaneamente múltiplas funções lineares sujeitas a um conjunto de restrições também lineares. Este tipo de problemas, tipicamente, não admite uma única solução mas um conjunto de soluções incomparáveis. Este conjunto de soluções representa o melhor resultado possível entre os objetivos conflitantes, visto que não é possível melhorar um critério sem piorar algum dos outros.

Neste trabalho iremos introduzir as principais definições de otimização multiobjetivo, analisar as principais ideias da programação linear multiobjetivo, sumarizar os resultados mais importantes de programação linear e mostrar como usar os programas lineares paramétricos para resolver programas lineares com múltiplos objetivos, utilizando exemplos em \mathbb{R}^2 que ilustrem esses resultados. Provaremos alguns resultados importantes, como por exemplo o principal teorema de programação linear multiobjetivo que afirma que todas as soluções eficientes são também propriamente eficiente.

Estudaremos também dois métodos para resolver programas lineares biobjetivo, o método da soma ponderada e o método simplex, ambos implementados em *Matlab*. Faremos ainda uma breve generalização do método simplex para o caso de programação linear multiobjetivo. Concluímos este trabalho com um breve estudo computacional comparativo dos dois métodos que permite concluir que o método simplex é mais rápido que o método da soma ponderada nos exemplos testados.

Conteúdo

Lista de Figuras	ix
1 Introdução	1
2 Introdução à otimização multiobjetivo	5
3 Programação linear multiobjetivo	13
3.1 Propriedades do MOLP	14
3.2 Geometria da programação linear	16
3.3 Método da soma ponderada	17
3.3.1 Método da soma ponderada para programas lineares biobjetivo	22
3.4 Método Simplex para programas lineares	28
3.4.1 Método Simplex para programas lineares monobjetivo	28
3.4.2 Método Simplex para programas lineares biobjetivo	33
3.4.3 Método simplex para programas lineares multiobjetivo	40
4 Comparação dos métodos	43
Bibliografia	47
Apêndice A Códigos	49

Lista de Figuras

1.1	Região admissível \mathcal{X} do problema de otimização multiobjetivo e região admissível \mathcal{Y} no espaço dos objetivos, respetivamente.	2
1.2	Região admissível \mathcal{X} do problema de otimização multiobjetivo.	3
1.3	Região admissível \mathcal{Y} no espaço de objetivos.	4
1.4	Gráfico da função em \mathcal{X}_E , com $x_1 \in [0.5; 1.5]$ e $x_2 = 1.5 - x_1$	4
2.1	Conjuntos eficiente, propriamente eficiente, não dominado e propriamente não dominado do exemplo 1.2.	7
2.2	Conjunto admissível \mathcal{X} do problema de otimização multiobjetivo.	8
2.3	Conjuntos \mathcal{Y} e $\mathcal{Y} + \mathbb{R}_{\geq}^2$	10
2.4	Relação entre \mathcal{Y}_N e $\mathcal{S}_0(\mathcal{Y})$	12
3.1	Região admissível no espaço de objetivos de $LP(\lambda)$	19
3.2	Conjunto das soluções eficientes.	25
3.3	Região admissível \mathcal{X} do problema.	25
3.4	Primeira iteração da Fase II do método soma ponderada.	26
3.5	Segunda iteração da Fase II do método soma ponderada.	27
4.1	Resolução gráfica do PL para $n = 6$	44
4.2	Número de iterações até $n = 40$	44
4.3	Tempo até $n = 40$	45

Capítulo 1

Introdução

Em diversas situações do quotidiano temos que lidar com diferentes problemas, perante os quais temos que tomar decisões. Cada solução encontrada, pode ser avaliada de uma ou mais formas com vista a seleccionar a melhor forma possível de resolver o problema. Quando cada solução do problema é avaliada por mais do que um critério em simultâneo, estamos na presença de otimização multiobjetivo [6].

O exemplo seguinte, relativamente à escolha dos ingredientes de um pequeno almoço, irá servir como motivação ao longo do trabalho desenvolvido.

Exemplo 1.1. *Pretende-se preparar um pequeno almoço saudável, isto é, com um número reduzido de calorias mas, ao mesmo tempo, com o menor custo possível. O pequeno almoço tem que conter maçãs e/ou leite, sendo que a quantidade total de maçãs e leite não pode ser superior a 500g e a quantidade de vitamina C tem que ser superior ou igual a 5 mg. Consideremos que o custo, a quantidade de calorias e vitamina C destes alimentos são dados pela seguinte tabela:*

<i>por 100 g</i>	<i>Custo</i>	<i>Calorias</i>	<i>Vitamina C</i>
<i>Maçã</i>	<i>20 cent.</i>	<i>60 Kcal</i>	<i>5 mg</i>
<i>Leite</i>	<i>10 cent.</i>	<i>90 Kcal</i>	<i>2.5 mg</i>

Neste problema, pretendemos uma solução que minimize o número de calorias e o custo, sendo portanto de natureza multiobjetivo. Deste modo, o problema de otimização multiobjetivo a resolver é:

$$\begin{aligned} \min & 20x_1 + 10x_2 \\ \min & 60x_1 + 90x_2 \\ \text{s.a} & 5x_1 + 2.5x_2 \geq 5 \\ & x_1 + x_2 \leq 5 \\ & x_1, x_2 \geq 0 \end{aligned}$$

em que x_1 e x_2 são as nossas variáveis de decisão pois estas representam as quantidades (em centenas de gramas) de maçã e leite, respetivamente.

Este problema linear multiobjetivo tem duas funções objetivo, duas restrições lineares, duas restrições de sinal e duas variáveis, x_1 e x_2 , por isso podemos representar o problema graficamente em \mathbb{R}^2 da seguinte forma:

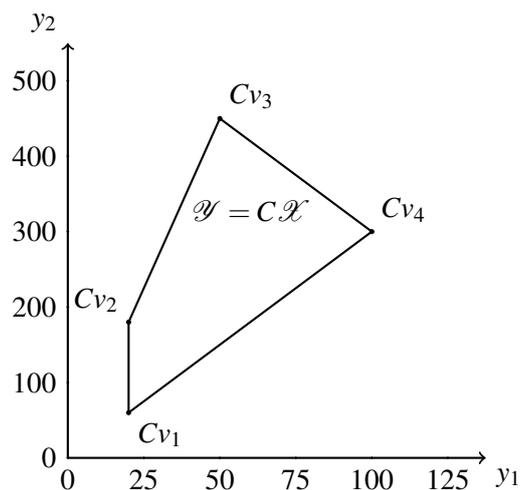
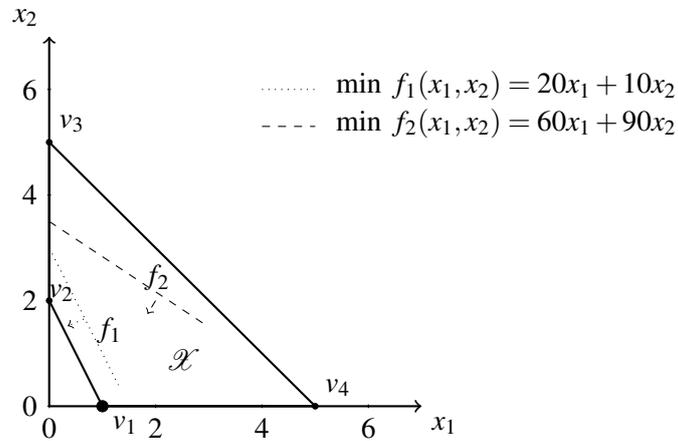


Figura 1.1 Região admissível \mathcal{X} do problema de otimização multiobjetivo e região admissível \mathcal{Y} no espaço dos objetivos, respetivamente.

A figura 1.1 mostra a região admissível do problema de otimização multiobjetivo no espaço das soluções e a região admissível no espaço dos objetivos. Observando esta figura conclui-se que o problema linear multiobjetivo tem uma única solução, o ponto v_1 . Deste modo, perante as restrições impostas, a melhor solução para este problema seria comer apenas 100g de maçã ao pequeno almoço.

No exemplo anterior, a solução encontrada otimiza, simultaneamente, os dois objetivos. Contudo esta situação não ocorre em geral, uma vez que este tipo de soluções não está disponível na região admissível. Deste modo, para tornar o exemplo mais realista, acrescentaremos uma nova restrição associada à quantidade de açúcar.

Exemplo 1.2. Consideremos o exemplo anterior ao qual adicionamos a seguinte restrição: a quantidade de açúcar ingerida não pode ser inferior a 3 gramas.

A tabela seguinte apresenta o custo, a quantidade de calorias, vitamina C e açúcares de 100 gramas de maçã e leite.

por 100 g	custo	calorias	vitamina C	açúcares
maçã	20 cent.	60 Kcal	5 mg	2g
leite	10 cent.	90 Kcal	2.5 mg	2g

Assim, temos um novo problema linear multiobjetivo:

$$\begin{aligned}
 \min \quad & 20x_1 + 10x_2 \\
 \min \quad & 60x_1 + 90x_2 \\
 \text{s.a} \quad & 5x_1 + 2.5x_2 \geq 5 \\
 & x_1 + x_2 \leq 5 \\
 & 2x_1 + 2x_2 \geq 3 \\
 & x_1, x_2 \geq 0.
 \end{aligned}$$

Analogamente ao exemplo anterior o problema pode ser resolvido graficamente em \mathbb{R}^2 , como podemos observar nas figuras 1.2 e 1.3. Desta forma, conclui-se que este novo problema linear multiobjetivo não tem apenas uma solução mas sim um conjunto de soluções incomparáveis, visto que não é possível reduzir as calorias do pequeno almoço, sem aumentar o custo do mesmo e vice-versa.

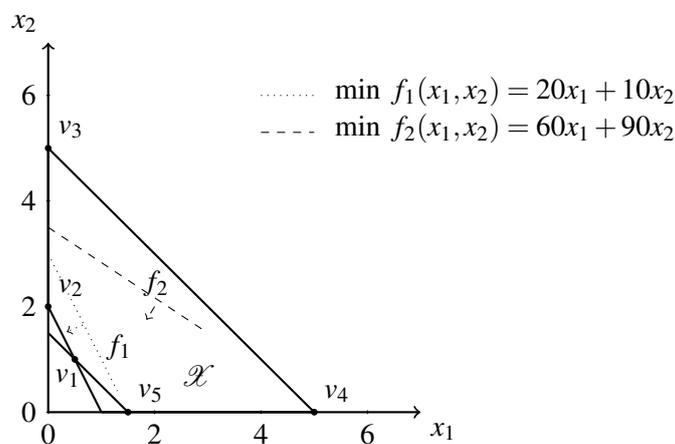


Figura 1.2 Região admissível \mathcal{X} do problema de otimização multiobjetivo.

Neste caso esse conjunto de soluções é dado por:

$$\{(x_1, x_2) \in \mathbb{R}^2 : x_1 + x_2 = 1.5, x_1 \geq 0.5, x_2 \geq 0\}$$

e o conjunto dos valores da função objetivo correspondente é:

$$\{(y_1, y_2) \in \mathbb{R}^2 : y_2 = -3y_1 + 180, 20 \leq y_1 \leq 30\}.$$

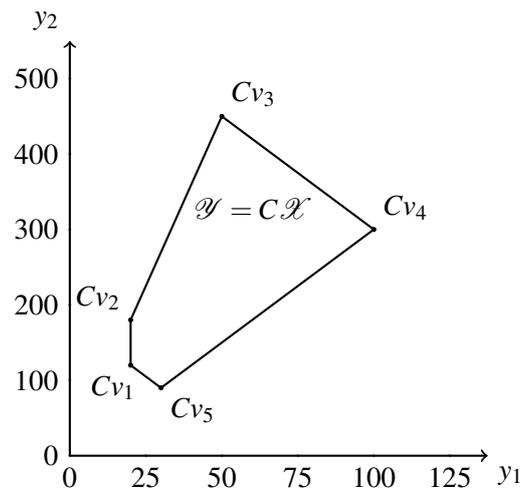


Figura 1.3 Região admissível \mathcal{Y} no espaço de objetivos.

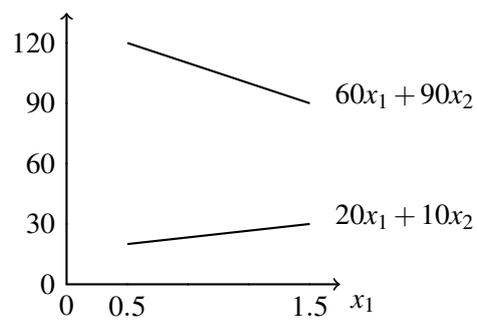


Figura 1.4 Gráfico da função em \mathcal{X}_E , com $x_1 \in [0.5; 1.5]$ e $x_2 = 1.5 - x_1$.

A figura 1.4 ilustra as soluções são incomparáveis. A função $60x_1 + 90x_2$ decresce ao longo de x_1 , enquanto que a função $20x_1 + 10x_2$ cresce ligeiramente ao longo de x_1 , pelo que os dois objetivos são conflituosos.

Capítulo 2

Introdução à otimização multiobjetivo

Nesta secção iremos apresentar alguns formalismos importantes para o estudo deste problema. Começemos por analisar alguns conceitos e propriedades da otimização multiobjetivo que vão ser utilizados ao longo do texto, bem como, exemplos que ilustrem os mesmos.

Um problema geral de otimização multiobjetivo é usualmente formulado, sem perda de generalidade, da seguinte forma:

$$\begin{array}{ll} \min & (f_1(x), \dots, f_p(x)) \\ \text{sujeito a} & x \in \mathcal{X}, \end{array} \quad (2.1)$$

onde $f_i : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, \dots, p$, denominam-se por funções objetivo e $\mathcal{X} \subset \mathbb{R}^n$ é o conjunto das soluções admissíveis.

Ao longo deste trabalho, utilizaremos as seguintes relações de ordem em \mathbb{R}^p , definidos do seguinte modo. Dado $y^1, y^2 \in \mathbb{R}^2$:

$$\begin{aligned} y^1 > y^2 & \text{ se } y_k^1 > y_k^2 \quad k = 1, \dots, p \\ y^1 \geq y^2 & \text{ se } y_k^1 \geq y_k^2 \quad k = 1, \dots, p \\ y^1 \geq y^2 & \text{ se } y^1 \geq y^2, \quad y^1 \neq y^2. \end{aligned}$$

Estas relações são todas transitivas, mas apenas a segunda é uma relação de ordem parcial [3]. Ao longo do trabalho iremos também usar as notações $y^1 < y^2$, $y^1 \leq y^2$ e $y^1 \leq y^2$ para denotar $y^2 > y^1$, $y^2 \geq y^1$ e $y^2 \geq y^1$, respetivamente.

Destas relações surgem naturalmente as seguintes notações:

$$\begin{aligned} \mathbb{R}_{\geq}^p & := \{y \in \mathbb{R}^p : y \geq 0\}, \text{ o conjunto dos vetores não negativos de } \mathbb{R}^p; \\ \mathbb{R}_{\leq}^p & := \{y \in \mathbb{R}^p : y \leq 0\} = \mathbb{R}_{\geq}^p \setminus \{0\}; \\ \mathbb{R}_{>}^p & := \{y \in \mathbb{R}^p : y > 0\}, \text{ o conjunto dos vetores positivos de } \mathbb{R}^p. \end{aligned}$$

O conjunto das soluções ótimas do problema linear mono-objetivo em cada uma das p funções objetivo é $\mathcal{X}_i = \{\hat{x} \in \mathcal{X} : f_i(\hat{x}) \leq f_i(x), \quad x \in \mathcal{X}\}$, $i = 1, \dots, p$.

Ao longo do trabalho iremos admitir que $\bigcap_{i=1}^p \mathcal{X}_i = \emptyset$. Esta suposição garante que não existe nenhuma solução que minimize todas as funções objetivo ao mesmo tempo.

A decisão neste tipo de problemas não pode ser reduzida à procura da solução ótima no sentido tradicional, visto que na maioria dos casos do dia-a-dia não existe uma solução ótima. Uma vez que a relação \geq é uma relação de ordem parcial e não total, e atendendo a que $\bigcap_{i=1}^p \mathcal{X}_i = \emptyset$, não existe uma única solução mas um conjunto de soluções incomparáveis entre si. Assim o conceito de solução ótima é substituído pelo conceito de eficiência, que está relacionado com o conceito de não dominância, estando o primeiro associado ao espaço das soluções e o segundo espaço dos objetivos.

Definição 2.1. *Seja $\hat{x} \in \mathcal{X}$ uma solução admissível do problema (2.1) e $\hat{y} = f(\hat{x})$.*

1. *\hat{x} é fracamente eficiente se não existe nenhum $x \in \mathcal{X}$ tal que $f(x) < f(\hat{x})$; nesse caso, \hat{y} é fracamente não dominado.*
2. *\hat{x} é eficiente se não existe nenhum $x \in \mathcal{X}$ tal que $f(x) \leq f(\hat{x})$; nesse caso, \hat{y} é não dominado.*
3. *\hat{x} é propriamente eficiente se é eficiente e existe um número real M , não negativo tal que $\forall i \in \{1, \dots, p\}, \forall x \in \mathcal{X}$, que satisfaça $f_i(x) < f_i(\hat{x}), \exists j \in \{1, \dots, p\}$ tal que $f_j(x) > f_j(\hat{x})$ e*

$$\frac{f_i(\hat{x}) - f_i(x)}{f_j(x) - f_j(\hat{x})} \leq M.$$

Desta forma podemos então definir os seguintes conjuntos:

- O conjunto de todas as soluções eficientes $\hat{x} \in \mathcal{X}$ é denotado por \mathcal{X}_E e designado por conjunto eficiente. O conjunto de todos os pontos não dominados, isto é $f(\mathcal{X}_E)$ é denotado por \mathcal{Y}_N e designado por conjunto não dominado.
- O conjunto de soluções fracamente eficiente é denotado por \mathcal{X}_{wE} e $\mathcal{Y}_{wN} = f(\mathcal{X}_{wE})$ é o conjunto fracamente não dominado.
- O conjunto de soluções propriamente eficiente é denotado por \mathcal{X}_{pE} e $\mathcal{Y}_{pN} = f(\mathcal{X}_{pE})$ é o conjunto propriamente não dominado.

Consideremos novamente o exemplo 1.2 para ilustrar estes conceitos.

Na Figura 2.1 são realçados os conjuntos

$$\mathcal{X}_E = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 + x_2 = 1.5, x_1 \geq 0.5, x_2 \geq 0\};$$

$$\mathcal{X}_{wE} = \{(x_1, x_2) \in \mathbb{R}^2 : 2x_1 + x_2 = 2, 0 \leq x_1 \leq 0.5\} \cup \mathcal{X}_E;$$

$$\mathcal{Y}_N = \{(y_1, y_2) \in \mathbb{R}^2 : y_2 = -3y_1 + 180, 20 \leq y_1 \leq 30\};$$

$$\mathcal{Y}_{wN} = \{(y_1, y_2) \in \mathbb{R}^2 : y_1 = 20, 120 \leq y_2 \leq 180\} \cup \mathcal{Y}_N$$

com um traço preto mais grosso. Na figura de baixo observamos que o ponto y^1 é dominado uma vez que $\exists x \in \mathcal{X}$ tal que $y^1 < f(x)$. De facto, todos os pontos que se encontram em $(y^1 - \mathbb{R}_{\geq}^p) \cap \mathcal{Y} = \{y \in \mathcal{Y} : y = y^1 - y^*, \text{ para algum } y^* \in \mathbb{R}_{\geq}^p\}$, que são representados pelos pontos de \mathcal{Y} dentro do ângulo reto com vértice em y^1 , dominam este ponto. Por outro lado o ponto y^2 é não dominado, pois $(y^2 - \mathbb{R}_{\geq}^p) \cap \mathcal{Y}$ é vazio.

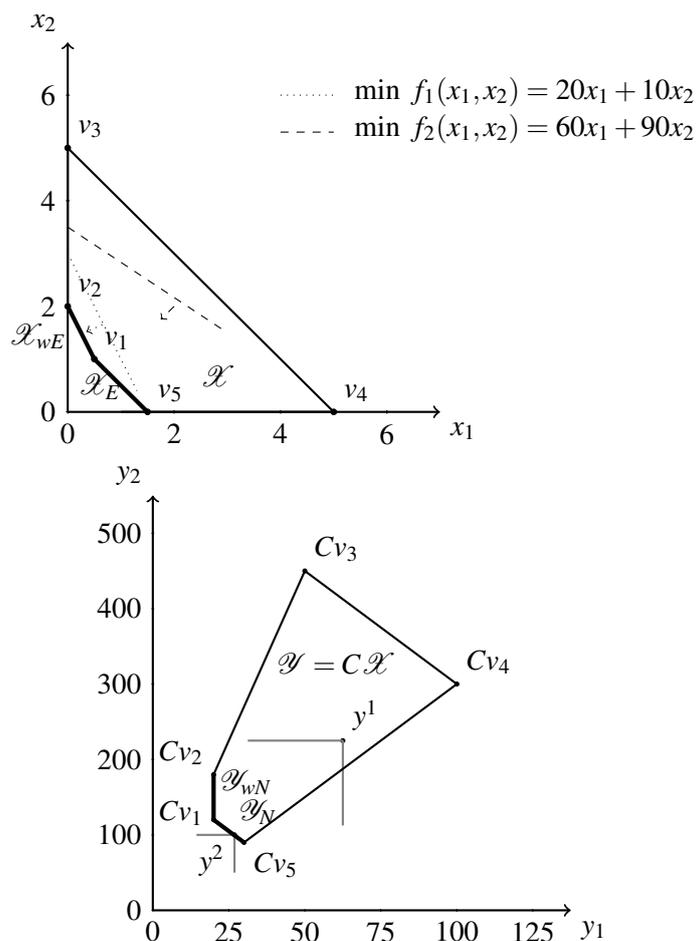


Figura 2.1 Conjuntos eficiente, propriamente eficiente, não dominado e propriamente não dominado do exemplo 1.2.

Note-se que pode haver situações em que \mathcal{X}_E é vazio sem o conjunto \mathcal{X} o ser. De facto, se no problema anterior retirarmos a restrição $x_1 + x_2 \leq 5$ e alterarmos as funções objetivo para o seu simétrico, obtemos um conjunto eficiente vazio, isto é, $\mathcal{X}_E = \emptyset$, sem que \mathcal{X} não o seja. Este resultado ocorre visto que a procura de soluções ótimas é feita no sentido em que o conjunto admissível do espaço de decisão é ilimitado.

Pode também acontecer que \mathcal{X} seja ilimitado e $\mathcal{X}_E \neq \emptyset$. De facto, basta considerar o exemplo 1.2, mantendo as mesmas funções objetivo e removendo apenas a restrição $x_1 + x_2 \leq 5$.

É possível também ilustrar uma solução eficiente que não é propriamente eficiente mas para isso já não podemos utilizar um problema linear multiobjetivo, como veremos posteriormente. Para ilustrar esta observação vejamos o seguinte exemplo.

Exemplo 2.2. Consideremos o seguinte programa:

$$\begin{aligned} & \min && x_1 \\ & \min && x_2 \\ & \text{s.a} && (x_1 - 3)^2 + (x_2 - 3)^2 \leq 2 \end{aligned}$$

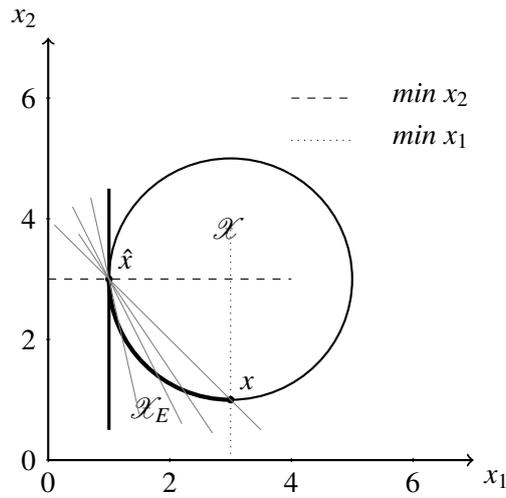


Figura 2.2 Conjunto admissível \mathcal{X} do problema de otimização multiobjetivo.

Na figura 2.2 está representado o conjunto admissível \mathcal{X} e o conjunto das soluções eficientes $\mathcal{X}_E = \{(x_1 - 3)^2 + (x_2 - 3)^2 = 2, 3 - \sqrt{2} \leq x_1 \leq 3, 3 - \sqrt{2} \leq x_2 \leq 3\}$, estando destacada a solução eficiente $\hat{x} = (3 - \sqrt{2}, 3)$, que mostraremos não ser propriamente eficiente, e uma solução eficiente genérica. Seja $x = (x_1, 3 - \sqrt{2 - (x_1 - 3)^2}) \in \mathcal{X}_E$, com $x_1 \in]3 - \sqrt{2}, 3]$ uma solução eficiente genérica. Estas soluções verificam $f_2(x) < f_2(\hat{x})$ e $f_1(x) > f_1(\hat{x})$ mas o limite

$$\begin{aligned} & \lim_{x \rightarrow \hat{x}} \frac{f_2(\hat{x}) - f_2(x)}{f_1(x) - f_1(\hat{x})} \\ &= \lim_{x_1 \rightarrow (3 - \sqrt{2})^+} \frac{3 - 3 + \sqrt{2 - (x_1 - 3)^2}}{x_1 - 3 + \sqrt{2}} \quad (\text{Utilizando a regra de Cauchy}) \\ &= \lim_{x_1 \rightarrow (3 - \sqrt{2})^+} \frac{1/2 - 2(x_1 - 3)(2 - (x_1 - 3)^2)^{-1/2}}{1} \\ &= \lim_{x_1 \rightarrow (3 - \sqrt{2})^+} -\frac{x_1 - 3}{\sqrt{2 - (x_1 - 3)^2}} \\ &= \infty. \end{aligned}$$

Assim, concluímos que $\lim_{\substack{x \rightarrow \hat{x} \\ x \in \mathcal{X}_E}} \frac{f_2(\hat{x}) - f_2(x)}{f_1(x) - f_1(\hat{x})} = \infty$, logo não existe $M > 0$ tal que $\forall i \in \{1, 2\}, \forall x \in \mathcal{X}$, que satisfaça $f_i(x) < f_i(\hat{x}), \exists j \in \{1, 2\}$ tal que $f_j(x) > f_j(\hat{x})$ e

$$\frac{f_i(\hat{x}) - f_i(x)}{f_j(x) - f_j(\hat{x})} \leq M.$$

e portanto \hat{x} não é propriamente eficiente.

Existem vários métodos para encontrar soluções multiobjetivo, um dos quais é resolver um problema paramétrico, com $\lambda \in \mathbb{R}^P$:

$$\min_{x \in \mathcal{X}} \sum_{i=1}^P \lambda_i f_i(x).$$

Este tipo de problemas é muito mais simples de resolver pois trata-se de um problema monocritério. Além disso, variando o valor de λ , podemos obter várias soluções eficientes do problema multiobjetivo.

Com o intuito de aplicar este método, vamos ao longo do trabalho tentar responder às seguintes questões:

- A solução desse problema é sempre uma solução eficiente?
- Todas as soluções eficientes são soluções do problema anterior para algum λ ?

No que se segue vamos sumarizar alguns resultados e notações importantes que envolvem os conjuntos admissíveis \mathcal{X} e \mathcal{Y} . Para tal comecemos por considerar:

$$\mathcal{S}(\mathcal{Y}) = \{\hat{y} \in \mathcal{Y} : \exists \lambda > 0 \text{ para o qual } \lambda^T \hat{y} \leq \lambda^T y, \forall y \in \mathcal{Y}\}$$

e

$$\mathcal{S}_0(\mathcal{Y}) = \{\hat{y} \in \mathcal{Y} : \exists \lambda \geq 0 \text{ para o qual } \lambda^T \hat{y} \leq \lambda^T y, \forall y \in \mathcal{Y}\}$$

De seguida realcemos algumas definições. Sejam $\mathcal{Y} \subset \mathbb{R}^P$:

- Seja $y \in \mathcal{Y}$, o conjunto $\mathcal{Y}_y = (y - \mathbb{R}_{\geq}^P) \cap \mathcal{Y}$ diz-se uma secção de \mathcal{Y} . Se for compacto diz-se uma secção compacta.
- \mathcal{Y} é \mathbb{R}_{\geq}^P -compacto se $\forall y \in \mathcal{Y}$ a secção \mathcal{Y}_y é compacta.
- \mathcal{Y} é \mathbb{R}_{\geq}^P -convexo se $\mathcal{Y} + \mathbb{R}_{\geq}^P$ é convexo.
- \mathcal{Y} é \mathbb{R}_{\geq}^P -fechado se $\mathcal{Y} + \mathbb{R}_{\geq}^P$ é fechado.
- \mathcal{Y} é conexo se não puder ser escrito como $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2$, com $\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_1 \cap \mathcal{Y}_2 \neq \emptyset$.

Vejamos agora um exemplo que ilustra estas definições, mostrando também que é possível termos um conjunto \mathcal{Y} que não é nem convexo, nem fechado, nem compacto, mas pode ser \mathbb{R}_{\geq}^P -convexo, \mathbb{R}_{\geq}^P -fechado e \mathbb{R}_{\geq}^P -compacto.

Exemplo 2.3. Na figura 2.3 está representado o conjunto \mathcal{Y} , que não é convexo, nem fechado, nem compacto. Este conjunto é definido por $\mathcal{Y} = \{(y_1, y_2) \in \mathbb{R}^2 : y_1 \geq 1 \wedge y_2 \geq 1 \wedge (y_1 < 2 \vee y_2 < 2)\}$. Ora,

o conjunto $\mathcal{Y} + \mathbb{R}_{\geq}^2$ (figura da direita da figura 2.3) é definido por $\{(y_1, y_2) \in \mathbb{R}^2 : y_2 \geq 1 \wedge y_2 \geq 1\}$. No entanto este segundo conjunto já é convexo, fechado pelo que \mathcal{Y} é \mathbb{R}_{\geq}^p -convexo e \mathbb{R}_{\geq}^p -fechado. Além disso, dado $(y_1, y_2) \in \mathcal{Y}$, a secção $(y - \mathbb{R}_{\geq}^p) \cap \mathcal{Y} = \{y^* \in \mathcal{Y} : y^* \leq y\}$ é compacta pelo que \mathcal{Y} é \mathbb{R}_{\geq}^p -compacto.

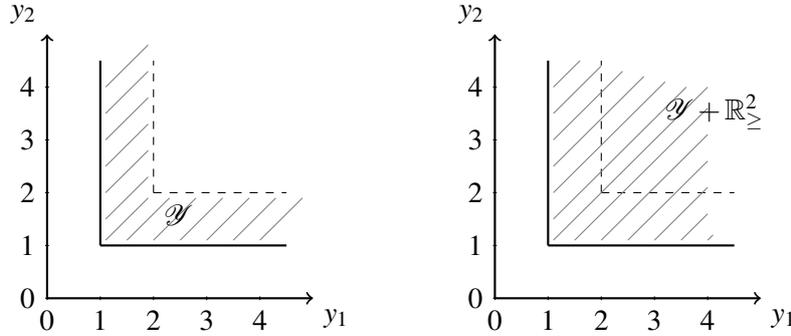


Figura 2.3 Conjuntos \mathcal{Y} e $\mathcal{Y} + \mathbb{R}_{\geq}^2$.

Definição 2.4. \mathcal{Y} é limitado inferiormente se $\exists y \in \mathbb{R}^p$ tal que $\mathcal{Y} \subset y + \mathbb{R}_{\geq}^p$ e \mathcal{Y} é limitado superiormente se $\exists y \in \mathbb{R}^p$ tal que $\mathcal{Y} \subset y - \mathbb{R}_{\geq}^p$.

Desta forma estamos em condições de concluir os seguintes resultados:

Teorema 2.5. Para qualquer conjunto $\mathcal{Y} \subset \mathbb{R}^p$ tem-se $\mathcal{S}_0(\mathcal{Y}) \subset \mathcal{Y}_{wN}$.

Demonstração. Seja $\lambda \in \mathbb{R}_{\geq}^p$ e \hat{y} uma solução ótima de $\mathcal{S}_0(\mathcal{Y})$, então

$$\sum_{k=1}^p \lambda_k \hat{y}_k \leq \sum_{k=1}^p \lambda_k y_k, \text{ para todo o } y \in \mathcal{Y}.$$

Por redução absurdo, suponhamos que $\hat{y} \in \mathcal{Y}_{wN}$, então existe algum $y' \in \mathcal{Y}$ tal que $y'_k < \hat{y}_k$, $k = 1, \dots, p$. Somando em k e multiplicando por λ obtém-se

$$\sum_{k=1}^p \lambda_k y'_k < \sum_{k=1}^p \lambda_k \hat{y}_k, \text{ para todo o } y \in \mathcal{Y},$$

pois pelo menos um dos pesos λ_k é positivo. Chegamos assim a uma contradição. \square

O inverso também possível ser provado, mas é necessário que \mathcal{Y} seja \mathbb{R}_{\geq}^p -convexo.

Teorema 2.6. Se \mathcal{Y} é \mathbb{R}_{\geq}^p -convexo, então $\mathcal{Y}_{wN} = \mathcal{S}_0(\mathcal{Y})$.

Relacionemos no que se segue $\mathcal{S}(\mathcal{Y})$ e $\mathcal{S}_0(\mathcal{Y})$ com \mathcal{Y}_N .

Teorema 2.7. Consideremos $\mathcal{S}(\mathcal{Y})$ e \mathcal{Y}_N definidos anteriormente, então $\mathcal{S}(\mathcal{Y}) \subset \mathcal{Y}_N$.

Demonstração. Seja $\hat{y} \in \mathcal{S}(\mathcal{Y})$, então existe algum $\lambda \in \mathbb{R}_{>}^p$ tal que $\sum_{i=1}^p \lambda_i \hat{y}_i \leq \sum_{i=1}^p \lambda_i y_i$, $y \in \mathcal{Y}$.

Suponhamos que $\hat{y} \notin \mathcal{Y}_N$, então existe $y' \in \mathcal{Y}$ tal que $y' \leq \hat{y}$. Multiplicando ambos os membros pelos pesos vem $\lambda_i y'_i \leq \lambda_i \hat{y}_i$, $i = 1, \dots, p$ e $\lambda_i y'_i < \lambda_i \hat{y}_i$ para um i . Da desigualdade estrita e do facto de λ_i ser positivo conclui-se que, $\sum_{i=1}^p \lambda_i \hat{y}_i > \sum_{i=1}^p \lambda_i y'_i$, o que contradiz $\hat{y} \in \mathcal{S}(\mathcal{Y})$. \square

Se \mathcal{Y} for \mathbb{R}_{\geq}^p -convexo o inverso pode ser provado.

Corolário 2.8. *Se \mathcal{Y} é \mathbb{R}_{\geq}^p -convexo, então $\mathcal{Y}_N \subset \mathcal{S}_0(\mathcal{Y})$.*

Demonstração. Este resultado é consequência imediata do teorema 2.6 uma vez que

$$\mathcal{Y}_N \subset \mathcal{Y}_{wN} = \mathcal{S}_0(\mathcal{Y}).$$

□

Assim, relacionando os teoremas enunciados anteriormente, vem que se \mathcal{Y} for \mathbb{R}_{\geq}^p -convexo vem ainda que

$$\mathcal{S}(\mathcal{Y}) \subset \mathcal{Y}_N \subset \mathcal{Y}_{wN} = \mathcal{S}_0(\mathcal{Y}) \quad (2.2)$$

Observemos que se \mathcal{Y} não for \mathbb{R}_{\geq}^p -convexo então \mathcal{Y}_N não está necessariamente contido em $\mathcal{S}_0(\mathcal{Y})$. De facto, a figura 2.4 ilustra um exemplo onde $\mathcal{Y}_N \not\subset \mathcal{S}_0(\mathcal{Y})$.

No que se segue pretende-se relacionar $\mathcal{S}(\mathcal{Y})$ com \mathcal{Y}_{pN} , de forma a obter a mesma conclusão que em (2.2), para isso enunciemos o teorema de *Geoffrion(1968)* que relaciona as soluções ótimas do problema paramétrico com as soluções propriamente eficientes de problema 2.1.

Teorema 2.9. (Geoffrion(1968)) *Sejam $\lambda_k > 0$, $k = 1, \dots, p$, com $\sum_{k=1}^p \lambda_k = 1$, pesos positivos. Se \hat{x} é uma solução ótima do problema paramétrico então \hat{x} é uma solução propriamente eficiente do problema 2.1.*

Corolário 2.10. *Seja $\mathcal{Y} \subset \mathbb{R}^p$, então $\mathcal{S}(\mathcal{Y}) \subset \mathcal{Y}_{pN}$.*

O inverso também possível ser provado, desde que \mathcal{Y} seja \mathbb{R}_{\geq}^p -convexo.

Teorema 2.11. *Se \mathcal{Y} é \mathbb{R}_{\geq}^p -convexo então $\mathcal{Y}_{pN} \subset \mathcal{S}(\mathcal{Y})$.*

Assim, relacionando os corolários e teoremas enunciados anteriormente vem que

$$\mathcal{S}(\mathcal{Y}) = \mathcal{Y}_{pN} \subset \mathcal{Y}_N \subset \mathcal{Y}_{wN} = \mathcal{S}_0(\mathcal{Y})$$

se \mathcal{Y} for \mathbb{R}_{\geq}^p -convexo.

No que se segue são indicados uma série de resultados que serão necessários para a secção seguinte, e cujas demonstrações podem ser consultadas em [4].

Teorema 2.12. (Borwein(1983)) *Seja \mathcal{Y} um conjunto não vazio e suponhamos que \mathcal{Y} contém uma secção compacta. Então \mathcal{Y}_N é não vazio.*

Corolário 2.13. *Se \mathcal{Y} é \mathbb{R}_{\geq}^p -convexo então $\mathcal{Y}_N \subset \mathcal{S}(\mathcal{Y})$.*

Estamos agora em condições de garantir que os conjuntos das soluções eficientes, fracamente eficientes e não dominadas são conexos.

Teorema 2.14. (Naccache(1978)) *Se \mathcal{Y} é fechado, convexo e \mathbb{R}_{\geq}^p -compacto então \mathcal{Y}_N é conexo.*

Teorema 2.15. *Seja $\mathcal{X} \subset \mathbb{R}^n$ um conjunto convexo e compacto. Suponha que todas as funções objetivo f_k são convexas. Então \mathcal{X}_E e \mathcal{X}_{wE} são conjuntos conexos.*

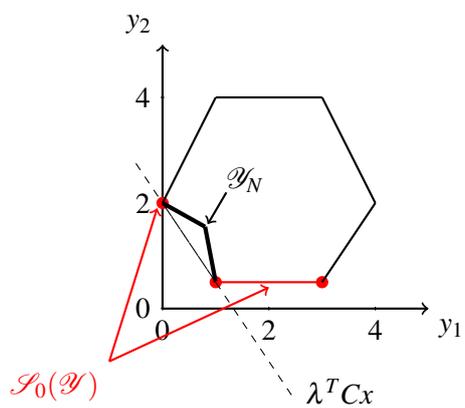


Figura 2.4 Relação entre \mathcal{Y}_N e $\mathcal{S}_0(\mathcal{Y})$.

Capítulo 3

Programação linear multiobjetivo

Um problema linear multiobjetivo é um caso particular de programação multiobjetivo, onde as várias funções objetivo e as restrições são lineares. Estes problemas são também designados, de forma abreviada por MOLP (Multicriteria Optimization Linear Problem). Sem perda de generalidade, iremos admitir que o conjunto das restrições se escreve da forma $Ax = b$. As funções objetivo são da forma $f_k(x) = c_k^T x$, $k = 1, \dots, p$, com $c_k \in \mathbb{R}^n$. Por último, restringe-se ainda que o vetor das variáveis seja não negativo em \mathbb{R}^n , isto é $x \geq 0$.

Desta forma, um programa linear multiobjetivo pode então ser escrito como o seguinte problema de otimização

$$\begin{array}{ll} \min & Cx \\ \text{sujeito a} & Ax = b \\ & x \geq 0, \end{array} \quad (3.1)$$

onde $C \in \mathbb{R}^{p \times n}$ se denomina por matriz critério e é construída com as linhas c_k^T , $k = 1, \dots, p$.

O conjunto admissível no espaço de decisão é

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

definido pela matriz das restrições $A \in \mathbb{R}^{m \times n}$, e pelo vetor $b \in \mathbb{R}^m$. A matriz A deverá satisfazer ainda $\text{car}(A) = m$ de modo ao sistema ser possível e não ter restrições redundantes. O conjunto admissível no espaço de objetivos é $\mathcal{Y} = C\mathcal{X} = \{Cx : x \in \mathcal{X}\}$.

O exemplo 1.2 que nos tem acompanhado é um MOLP e, portanto pode ser escrito na forma (3.1). Para o efeito, podemos reescreve-lo usando variáveis de folga:

$$\begin{array}{ll} \min & 20x_1 + 10x_2 \\ \min & 60x_1 + 90x_2 \\ \text{s.a} & 5x_1 + 2.5x_2 - x_3 = 5 \\ & x_1 + x_2 + x_4 = 5 \\ & 2x_1 + 2x_2 - x_5 = 3 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{array}$$

Deste modo, fazendo $C = \begin{bmatrix} 20 & 10 & 0 & 0 & 0 \\ 60 & 90 & 0 & 0 & 0 \end{bmatrix}$, $x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T$, $A = \begin{bmatrix} 5 & 2.5 & -1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 2 & 2 & 0 & 0 & -1 \end{bmatrix}$

e $b = \begin{bmatrix} 5 \\ 5 \\ 3 \end{bmatrix}$, obtemos a notação pretendida.

Com o problema formalmente definido, na próxima secção serão introduzidos os primeiros resultados.

3.1 Propriedades do MOLP

Nesta secção serão apresentados resultados sobre a existência de soluções eficientes e a caracterização do conjunto de soluções.

Lema 3.1. *Os conjuntos admissíveis \mathcal{X} no espaço de decisão e \mathcal{Y} no espaço dos objetivos do MOLP (3.1) são convexos e fechados.*

Demonstração. O conjunto $\{x \in \mathbb{R}^n : x \geq 0\}$ é um conjunto fechado, por ser a interseção dos semi-espacos fechados $x_i \geq 0$, $i = 1, \dots, n$. Além disso, $\{x \in \mathbb{R}^n : Ax = b\}$ é um conjunto fechado, pois sabemos que $A_i x = b_i$ é um hiperplano (que é fechado) e $Ax = b$ é a interseção de vários hiperplanos, todos fechados. Como a interseção de conjuntos fechados é um conjunto fechado, conclui-se que \mathcal{X} é fechado.

Provemos agora que \mathcal{X} é convexo, sejam $x_1, x_2 \in \mathcal{X}$, isto é, $Ax_1 = b$, $Ax_2 = b$ e $x_1, x_2 \geq 0$. Seja $\alpha \in (0, 1)$ então, $\alpha x_1 + (1 - \alpha)x_2 \geq 0$ e

$$\begin{aligned} A[\alpha x_1 + (1 - \alpha)x_2] &= A[\alpha x_1 + x_2 - \alpha x_2] \\ &= A\alpha x_1 + Ax_2 - A\alpha x_2 \\ &= \alpha b + b - \alpha b = b. \end{aligned}$$

Assim, $\alpha x_1 + (1 - \alpha)x_2 \in \mathcal{X}$ e portanto \mathcal{X} é um conjunto convexo.

Provemos agora que \mathcal{Y} é um conjunto fechado, para isso é necessário provar que para toda a sucessão $(y_n)_{n \in \mathbb{N}} \in \mathcal{Y}$ $\lim_{n \rightarrow \infty} y_n = y \implies y \in \mathcal{Y}$.

Ora, seja $y_n = Cx_n$, $x_n \in \mathcal{X} \ \forall n \in \mathbb{N}$

$$y = \lim_n y_n = \lim_n Cx_n = C \lim_n x_n.$$

Como $\lim_n x_n \in \mathcal{X}$, pois \mathcal{X} é um conjunto fechado, então $y \in \mathcal{Y}$ e portanto \mathcal{Y} é fechado.

Por último demonstramos a convexidade de \mathcal{Y} , sejam $y_1, y_2 \in \mathcal{Y}$, então existem $x_1, x_2 \in \mathcal{X}$ tal que $y_1 = Cx_1$ e $y_2 = Cx_2$. Deste modo, para qualquer $\alpha \in (0, 1)$:

$$\begin{aligned} & \alpha y_1 + (1 - \alpha)y_2 \\ &= \alpha Cx_1 + (1 - \alpha)Cx_2 \\ &= C[\alpha x_1 + (1 - \alpha)x_2]. \end{aligned}$$

Como \mathcal{X} é convexo, $\alpha x_1 + (1 - \alpha)x_2 \in \mathcal{X}$, logo \mathcal{Y} é convexo. \square

No que segue, serão enunciados alguns resultados sobre o conjunto admissível no espaço dos objetivos, \mathcal{Y} , que serão necessários posteriormente.

Corolário 3.2. *Se \mathcal{X} é limitado então \mathcal{Y} é compacto.*

Demonstração. Pelo teorema anterior temos que \mathcal{X} é fechado, e por hipótese temos que também é limitado, então \mathcal{X} é compacto. Deste modo, e uma vez que as funções lineares são contínuas, $\mathcal{Y} = C\mathcal{X}$ é compacto por ser a imagem de um compacto através de uma função contínua. \square

Lema 3.3. *\mathcal{Y} é \mathbb{R}_{\geq}^p -convexo.*

Demonstração. Pela definição \mathcal{Y} é \mathbb{R}_{\geq}^p -convexo se $\mathcal{Y} + \mathbb{R}_{\geq}^p$ é convexo. Sejam $y_1, y_2 \in \mathcal{Y} + \mathbb{R}_{\geq}^p$, isto é, $y_i = \bar{y}_i + w_i$, $i = 1, 2$, onde $\bar{y}_i \in \mathcal{Y}$ e $w_i \geq 0$. Consideremos ainda $\alpha \in [0, 1]$. Então:

$$\begin{aligned} \alpha y_1 + (1 - \alpha)y_2 &= \alpha(\bar{y}_1 + \bar{w}_1) + (1 - \alpha)(\bar{y}_2 + \bar{w}_2) \\ &= \alpha\bar{y}_1 + (1 - \alpha)\bar{y}_2 + \alpha\bar{w}_1 + (1 - \alpha)\bar{w}_2 \in \mathcal{Y} + \mathbb{R}_{\geq}^p, \end{aligned}$$

pois $\alpha\bar{y}_1 + (1 - \alpha)\bar{y}_2 \in \mathcal{Y}$ que é convexo e $\alpha\bar{w}_1 + (1 - \alpha)\bar{w}_2 \geq 0$. Logo, $\mathcal{Y} + \mathbb{R}_{\geq}^p$ é convexo. \square

Lema 3.4. *Se \mathcal{Y} é limitado inferiormente então \mathcal{Y} é \mathbb{R}_{\geq}^p -compacto.*

Demonstração. Pela definição, \mathcal{Y} é \mathbb{R}_{\geq}^p -compacto se toda a secção $\mathcal{Y}_y = (y - \mathbb{R}_{\geq}^p) \cap \mathcal{Y}$, $y \in \mathcal{Y}$ é compacta. Inicialmente, notemos que \mathcal{Y}_y é fechado, pois é a interseção de dois conjuntos fechados, faltando apenas provar que \mathcal{Y} é limitado.

Por hipótese, temos que \mathcal{Y} é limitado inferiormente, logo \mathcal{Y}_y também é limitada inferiormente. Por outro lado, $\mathcal{Y}_y \subset y - \mathbb{R}_{\geq}^p$, logo concluímos que também é limitado superiormente. Portanto, \mathcal{Y}_y é limitado e consequentemente compacto, para todo $y \in \mathcal{Y}$. \square

Vejamos a seguir que $\mathcal{S}(\mathcal{Y})$ é de facto essencial para calcular as soluções não dominadas que por sua vez formam um conjunto conexo.

Teorema 3.5. *Suponhamos que $\mathcal{Y} \neq \emptyset$ e que \mathcal{Y} é limitado inferiormente. Então:*

1. $\mathcal{Y}_N \neq \emptyset$.
2. $\mathcal{S}(\mathcal{Y}) = \mathcal{Y}_{pN} \subset \mathcal{Y}_N$.
3. \mathcal{Y}_N é conexo.

4. Se \mathcal{X} é limitado então \mathcal{X}_{wE} e \mathcal{X}_E são conexos.

Demonstração. 1. Aplicando o lema 3.4 vem que \mathcal{Y} é \mathbb{R}_{\geq}^p -compacto. Como $\mathcal{Y} \neq \emptyset$ então pelo teorema 2.12 conclui-se que $\mathcal{Y}_N \neq \emptyset$.

2. Por definição, $\mathcal{Y}_{pN} \subset \mathcal{Y}_N$. Além disso, o corolário 2.10 garante que $\mathcal{S}(\mathcal{Y}) \subset \mathcal{Y}_{pN}$. Finalmente, o lema 3.3 assegura que \mathcal{Y} é \mathbb{R}_{\geq}^p -conexo e, portanto, pelo teorema 2.11 obtemos $\mathcal{Y}_{pN} \subset \mathcal{S}(\mathcal{Y})$.

3. Atendendo a que \mathcal{Y} é limitado inferiormente, pelo lema 3.4 tem-se que \mathcal{Y} é \mathbb{R}_{\geq}^p -compacto. Por outro lado, o lema 3.1 refere que \mathcal{Y} é convexo e fechado donde se conclui, usando o teorema 2.14, que \mathcal{Y}_N é conexo.

4. Pelo lema 3.1 vem que \mathcal{X} é fechado e convexo. Além disso, pela hipótese, \mathcal{X} é limitado e portanto \mathcal{X} é compacto. Aplicando o teorema 2.15 concluímos que \mathcal{X}_{wE} e \mathcal{X}_E são conexos. \square

No que se segue apresentam-se resultados sobre a geometria da programação linear.

3.2 Geometria da programação linear

Nesta secção irão ser apresentados os conceitos de hiperplano, poliedro e ponto extremo, bem como as suas propriedades, podendo ser encontrados com mais detalhe em [1] e [4].

Seja $a \in \mathbb{R}^n$ e $b \in \mathbb{R}$. O conjunto

$$\mathcal{H}_{a,b} := \{x \in \mathbb{R}^n : a^T x = b\}$$

é um hiperplano. Um hiperplano define dois semi-espacos, o espaço que constituído por $a^T x \leq b$ e $a^T x \geq b$. Definamos o primeiro conjunto por

$$\tilde{\mathcal{H}}_{a,b} := \{x \in \mathbb{R}^n : a^T x \leq b\}$$

pelo que o segundo será $\tilde{\mathcal{H}}_{-a,-b}$.

Seja $\mathcal{X} \subset \mathbb{R}^n$ um conjunto não-vazio. Um hiperplano $\mathcal{H}_{a,b}$ é chamado hiperplano de suporte de \mathcal{X} em \hat{x} se $\hat{x} \in \mathcal{X} \cap \mathcal{H}_{a,b}$ e $\mathcal{X} \subset \tilde{\mathcal{H}}_{a,b}$. Também se diz que $\mathcal{H}_{a,b}$ suporta \mathcal{X} em \hat{x} .

Um conjunto \mathcal{X} é chamado de poliedro se for a interseção de um número finito de semi-espacos. Por exemplo, o conjunto admissível nos espaço de decisão, $\mathcal{X} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$, de um LP é um poliedro, pois é a interseção dos semi-espacos \tilde{H}_{a_i, b_i} e $\tilde{H}_{-a_i, -b_i}$ com os semi-espacos \tilde{H}_{-e_i} , onde e_i é o i -ésimo vetor da base canónica.

Diz-se que $x \in \mathcal{X}$ é um vértice de \mathcal{X} se existir $a \in \mathbb{R}^n$ tal que

$$ax > a\bar{x}, \quad \text{para todo o } \bar{x} \in \mathcal{X} \setminus \{x\}.$$

Seja $\mathcal{X} \neq \emptyset$ um poliedro escrito da seguinte forma $\mathcal{X} = \{x : Ax \leq b\}$. Se $d \in \mathbb{R}^n$ tal que $Ad \leq 0$, então d é chamado de raio de \mathcal{X} . Um raio d é designado de raio extremo se não existem raios $d^1, d^2, d^1 \neq \alpha d^2$ para todo o $\alpha \in \mathbb{R}_{>}$, tal que $d = (1/2)(d^1 + d^2)$.

Estas definições levam-nos ao seguinte lema que nos diz que o conjunto dos raios, \mathcal{R} é um cone convexo, ou seja \mathcal{R} é convexo e para qualquer $p \in \mathcal{R}$ e $\alpha \geq 0$ então $\alpha p \in \mathcal{R}$.

Lema 3.6. *Seja \mathcal{X} um poliedro definido anteriormente e $d \in \mathbb{R}^n$ um raio de \mathcal{X} . Então $x + \alpha d \in \mathcal{X}$ para todo o $\alpha \geq 0$ e o conjunto $\{d : Ad \leq 0\}$ é um cone convexo.*

No que se segue enunciaremos algumas notações que nos levarão ao conceito de face e algumas das suas propriedades.

O conjunto de pontos $\{a^{(i)} \in \mathbb{R}^n : i \in K\}$ é independente afim se $\sum_i \lambda_i a^{(i)} = 0$ e $\sum_i \lambda_i = 0$ apenas quando $\lambda_i = 0$ para todo $i \in K$.

A dimensão de um poliedro, $\dim(\mathcal{X})$, é o número máximo de pontos independentes afim em \mathcal{X} menos 1. A dimensão pode ser: -1 (se \mathcal{X} for vazio); 0 (se \mathcal{X} for um ponto único); 1 (se \mathcal{X} for um segmento de reta) e até n quando $\mathcal{X} \subseteq \mathbb{R}^n$. No último caso dizemos que \mathcal{X} é totalmente dimensional.

Consideremos $\mathcal{H}_{a,b}$ um hiperplano de suporte do poliedro \mathcal{X} . Então $\mathcal{F} = \mathcal{X} \cap \mathcal{H}_{a,b}$ é denominado por face de \mathcal{X} . Por definição, todas as faces são poliedros. Faces de dimensão 0 são chamados pontos extremos, faces de dimensão 1 são chamados arestas e faces de dimensão $\dim(\mathcal{X}) - 1$ são chamados facetas.

Na secção seguinte estabelecem-se relações que permitem obter as soluções de \mathcal{Y}_N à custa das soluções de $\mathcal{S}(\mathcal{Y})$, ou seja, vamos ver como podemos chegar às soluções eficientes do problemas biobjetivo através da resolução de problemas monobjetivo.

3.3 Método da soma ponderada

Como referido anteriormente, um processo de determinar as soluções de MOLP é o método da soma ponderada. Este método consiste na transformação de um problema multiobjetivo num problema mono-objetivo através da atribuição de pesos a cada objetivo. Com a atribuição de diferentes pesos para cada objetivo temos uma nova função que representa uma combinação linear entre os objetivos.

Seja $\lambda \in \mathbb{R}_{\geq}^p$, o programa linear de soma ponderada denotado por $LP(\lambda)$ define-se da seguinte forma:

$$\begin{aligned} \min \quad & \lambda^T Cx \\ \text{sujeito a} \quad & Ax = b, \\ & x \geq 0. \end{aligned} \tag{3.2}$$

com $b \geq 0$. Mostraremos, de seguida, algumas relações entre a solução dos problemas MOLP e $LP(\lambda)$.

Teorema 3.7. *Seja $\hat{x} \in \mathcal{X}$ a solução ótima do $LP(\lambda)$ indicado em (3.2).*

1. *Se $\lambda \geq 0$ então \hat{x} é fracamente eficiente.*
2. *Se $\lambda > 0$ então \hat{x} é eficiente.*

Demonstração. 1. Suponhamos, por absurdo, que \hat{x} não é fracamente eficiente. Então, existe $x \in \mathcal{X}$ tal que

$$c_k^T x < c_k^T \hat{x}, \quad k = 1, \dots, p. \tag{3.3}$$

Multiplicando ambos os membros por λ_k vem,

$$\lambda_k c_k^T x \leq \lambda_k c_k^T \hat{x}, \quad k = 1, \dots, p, \quad (3.4)$$

em que a desigualdade estrita acontece pelo menos uma vez porque $\lambda \neq 0$. Somando para todos os valores de k temos $\lambda^T Cx < \lambda^T C\hat{x}$ e chegamos a uma contradição!

2. Supondo agora, por absurdo, que \hat{x} não é eficiente. Então existe $x \in \mathcal{X}$ tal que

$$c_k^T x \leq c_k^T \hat{x}, \quad k = 1, \dots, p,$$

sendo, pelo menos, uma das desigualdades estrita.

Multiplicando ambos os membros por $\lambda_k > 0$ vem,

$$\lambda_k c_k^T x \leq \lambda_k c_k^T \hat{x}, \quad k = 1, \dots, p,$$

onde pelo menos uma das desigualdades é estrita. Somando para todos os valores de k temos $\lambda^T Cx < \lambda^T C\hat{x}$ e chegamos novamente a uma contradição!

□

Este teorema dá-nos uma forma de encontrar soluções eficientes do MOLP. Consolidemos este resultado com um exemplo que ilustra como podemos encontrar graficamente pontos não dominados. Este facto é importante porque encontrando pontos não dominados ou pontos fracamente não dominados, conseguimos concluir se a solução ótima é eficiente ou apenas fracamente eficiente.

Exemplo 3.8. Consideremos o exemplo 1.2 que tem servido de base neste estudo. A figura 3.1 ilustra o conjunto \mathcal{Y} e curvas de nível da função $\lambda^T Cx$ para os seguintes valores de λ : $\lambda^1 = [3 \ 1]$, $\lambda^2 = [1 \ 1]$, $\lambda^3 = [0 \ 1]$ e $\lambda^4 = [1 \ 0]$.

Com λ^1 obtemos que todos os pontos em $\overline{Cv_1 Cv_5}$ são valores ótimos de $LP(\lambda^1)$ e correspondem também a pontos não dominados. Com λ^2 e λ^3 obtemos o único ponto não dominado Cv_5 e com λ^4 obtemos que todos os pontos em $\overline{Cv_1 Cv_2}$ são fracamente não dominados mas apenas Cv_1 é não dominado.

Da análise dos exemplos apresentados, podemos observar:

- Observação 3.9.**
1. Um único ponto não dominado pode ser identificado por diferentes vetores de ponderação λ , como é o caso de Cv_5 , como se pode observar no exemplo 3.8, pode ser identificado pelos vetores de ponderação λ^2 e λ^3 .
 2. Um vetor de ponderação λ pode identificar vários pontos não dominados, como é o caso de λ^1 , como se pode observar no exemplo 3.8, identifica todos os pontos de $\overline{Cv_1 Cv_5}$ como pontos não dominados.
 3. Com $\lambda \geq 0$ e com algum $\lambda_i = 0$ podemos ter:

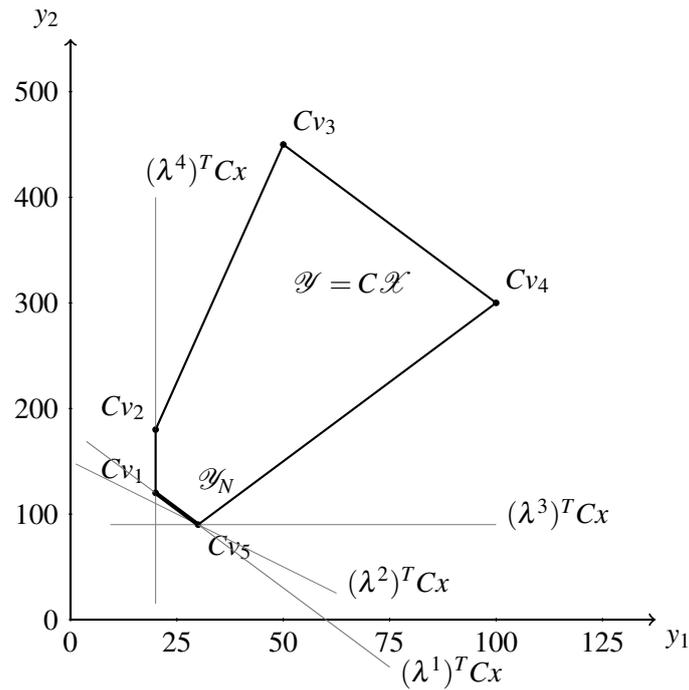


Figura 3.1 Região admissível no espaço de objetivos de LP(λ).

- (a) apenas pontos não dominados, como se pode observar no exemplo 3.8 usando o vetor de ponderação λ^3 .
- (b) pontos fracamente não dominados e pontos não dominados, como se pode observar no exemplo 3.8 usando o vetor de ponderação λ^4 .

Com o objetivo de provar o resultado fundamental da programação linear multiobjetivo, teorema 3.13 precisamos de provar alguns resultados auxiliares. Em primeiro lugar, necessitamos de identificar se uma dada solução admissível é eficiente ou não.

Lema 3.10. *Uma solução admissível $x^0 \in \mathcal{X}$ é eficiente se e só se o programa linear*

$$\begin{aligned}
 & \max && e^T z \\
 & \text{sujeito a} && Ax = b \\
 & && Cx + Iz = Cx^0 \\
 & && x, z \geq 0,
 \end{aligned} \tag{3.5}$$

onde $e^T = (1, \dots, 1) \in \mathbb{R}^p$ e I é a matriz identidade $p \times p$, tem valor ótimo nulo.

Demonstração. Notemos em primeiro lugar que o problema (3.5) é admissível uma vez que $(x^0, 0)$ é uma solução admissível desse problema.

Seja $(x, z) \in \mathcal{X} \times \mathbb{R}_{\geq}^p$ uma qualquer solução admissível de (3.5). Então

$$Cx + Iz = Cx^0 \Leftrightarrow z = Cx^0 - Cx \geq 0, \tag{3.6}$$

pelo facto de $z \geq 0$.

(\Rightarrow) O valor ótimo ser nulo significa que $e^T z = 0 \Leftrightarrow z = 0$, pois $z \geq 0$. Suponhamos, por redução ao absurdo, que o valor ótimo é maior que 0. Então (x, z) é admissível em (3.5). Como $z \geq 0$ e $z \neq 0$ vem que $z > 0$. Então de (3.6) vem $Cx \leq Cx^0$ e portanto x^0 não é eficiente, chegamos a uma contradição!

(\Leftarrow) Suponhamos, por redução ao absurdo, que \hat{x} não é eficiente. Então $\exists x \in \mathcal{X}$ tal que $Cx \leq Cx^0$. Fazendo $z = Cx^0 - Cx$ e usando (3.6) vem que $z \geq 0 \Rightarrow e^T z > 0$, chegamos a uma contradição! \square

Para provar que todas as soluções eficientes do MOLP (3.1) podem ser encontradas resolvendo o LP(λ) (3.2) necessitamos de relembrar o conceito de dualidade de programação linear.

Consideremos o seguinte programa linear de objetivo único (LP) denominado por primal:

$$\begin{aligned} \min \quad & c^T x \\ \text{sujeito a} \quad & Ax = b. \\ & x \geq 0 \end{aligned}$$

A cada LP primal define-se um outro LP associado, denominado por dual, que é definido por:

$$\begin{aligned} \max \quad & u^T b \\ \text{sujeito a} \quad & u^T A \leq c^T. \\ & u \in \mathbb{R}^m \end{aligned}$$

A relação entre os programas lineares primal e dual é estabelecida no seguinte teorema:

Teorema 3.11. (Dualidade de programação linear)

1. (Dualidade fraca) Seja $x \in \mathcal{X}$ e $u \in \mathcal{U}$, onde $\mathcal{U} = \{u \in \mathbb{R}^m : A^T u \leq c\}$, isto é, x e u são as soluções admissíveis do primal e dual, respetivamente. Então

$$b^T u \leq c^T x.$$

2. Se um dos problemas é ilimitado então o outro é inadmissível.

3. É possível que o primal e o dual sejam ambos inadmissíveis.

4. (Dualidade forte) Se um dos problemas tem solução ótima, então o outro também tem e o valor ótimo dos dois coincide.

Deste modo, conseguimos caracterizar as soluções eficientes através do dual de (3.5).

Lema 3.12. Uma solução admissível $x^0 \in \mathcal{X}$ é eficiente se e só se o programa linear

$$\begin{aligned} \min \quad & u^T b + w^T Cx^0 \\ \text{sujeito a} \quad & u^T A + w^T C \geq 0 \\ & w \geq e \\ & u \in \mathbb{R}^m, w \in \mathbb{R}^p \end{aligned} \tag{3.7}$$

tem a solução ótima (\hat{u}, \hat{w}) com $\hat{u}^T b + \hat{w}^T Cx^0 = 0$.

Demonstração. Recorrendo ao conceito de dualidade referido anteriormente temos que (3.7) é o dual do programa linear (3.5).

Deste modo, usando o lema 3.10, $x^0 \in \mathcal{X}$ é eficiente se e só se o valor ótimo de (3.5) é nulo. Aplicando o teorema 3.11 temos que o problema (3.7) também tem valor ótimo nulo, pelo que as suas soluções ótimas verificam $\hat{u}^T b + \hat{w}^T Cx^0 = 0$. \square

Estamos agora em condições de provar que todas as soluções do MOLP (3.1) podem ser encontradas resolvendo vários LP(λ) (3.2). Nesta prova consideramos uma solução eficiente x^0 e construímos o $\lambda \in \mathbb{R}_{>}^p$ tal que a solução eficiente x^0 seja a solução ótima do LP(λ) (3.2).

Teorema 3.13. (Isermann(1974)) *Uma solução admissível $x^0 \in \mathcal{X}$ é solução eficiente do MOLP (3.1) se e só se existe $\lambda \in \mathbb{R}_{>}^p$ tal que*

$$\lambda^T Cx^0 \leq \lambda^T Cx \quad (3.8)$$

para todo $x \in \mathcal{X}$.

Demonstração. (\Leftarrow) Do teorema 3.7 tem-se que, se $x^0 \in \mathcal{X}$ é a solução do LP(λ) (3.2) e $\lambda > 0$ então x^0 é eficiente.

(\Rightarrow) Seja $x^0 \in \mathcal{X}_E$. Pelo lema anterior vem que o LP (3.7) tem solução ótima (\hat{u}, \hat{w}) tal que $\hat{u}^T b + \hat{w}^T Cx^0 = 0 \Leftrightarrow \hat{u}^T b = -\hat{w}^T Cx^0$.

Fixando $w = \hat{w}$, vem $\min u^T b + \hat{w}^T Cx^0 = \min u^T b$, pois $\hat{w}^T Cx^0$ é uma constante.

Assim, \hat{u} também é solução ótima do LP:

$$\begin{aligned} \min \quad & u^T b \\ \text{sujeito a} \quad & u^T A + \hat{w}^T C \geq 0 \end{aligned}$$

este LP corresponde ao LP (3.7) com $w = \hat{w}$ fixo.

Então recorrendo à *dualidade forte*, vem que a solução ótima do dual do LP anterior,

$$\begin{aligned} \max \quad & -\hat{w}^T Cx \\ \text{sujeito a} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

existe. Pelo *dualidade fraca*, vem $-\hat{w}^T Cx \leq u^T b$ para todas as soluções admissíveis u do primal e para todas as soluções admissíveis x do dual.

Como vimos anteriormente $\hat{u}^T b = -\hat{w}^T Cx^0$, portanto x^0 é solução ótima do LP dual anterior.

Por outro lado temos a seguinte equivalência

$$\begin{aligned} \max \{ & -\hat{w}^T Cx : Ax = b, x \geq 0 \} \\ \Leftrightarrow & -\min \{ \hat{w}^T Cx : Ax = b, x \geq 0 \}. \end{aligned}$$

Das restrições do LP (3.7) temos $\hat{w} \geq e > 0$. Portanto x^0 é a solução ótima do LP (3.2) para $\lambda = \hat{w}$, ou seja $\hat{w}^T Cx^0 \leq \hat{w}^T Cx, \forall x \in \mathcal{X}$. Deste modo, tomando $\lambda = \hat{w}$, chegamos ao resultado pretendido. \square

Este teorema permite concluir o seguinte corolário.

Corolário 3.14. $\mathcal{X}_E = \mathcal{X}_{pE}$.

Sobre a primeira afirmação do teorema 3.5 temos as seguintes condições para a existência de soluções eficientes, respetivamente pontos não dominados.

Proposição 3.15. *Seja $x^0 \in \mathcal{X}$. Então o LP (3.5) é admissível e as seguintes afirmações são válidas.*

1. *Se (\hat{x}, \hat{z}) é a solução ótima de (3.5) então \hat{x} é uma solução eficiente do MOLP (3.1).*
2. *Se (3.5) é ilimitado então $\mathcal{X}_E = \emptyset$.*

Demonstração. 1. Se (\hat{x}, \hat{z}) é solução ótima de (3.5), então $C\hat{x} + I\hat{z} = Cx^0$ e $\hat{z} \geq 0$. Donde $\hat{z} = Cx^0 - C\hat{x}$ e portanto $C\hat{x} \leq Cx^0$.

Suponhamos que \hat{x} não é solução eficiente do MOLP (3.1). Então $\exists \tilde{x} \in \mathcal{X}$ tal que $C\tilde{x} \leq C\hat{x}$.

Defina-se $\tilde{z} = Cx^0 - C\tilde{x}$. Como $C\tilde{x} \leq C\hat{x}$ e $C\hat{x} \leq Cx^0$ então, por transitividade, $C\tilde{x} \leq Cx^0$. Portanto $\tilde{z} = Cx^0 - C\tilde{x} \geq 0$.

Logo $\tilde{z} \geq 0$ então (\tilde{x}, \tilde{z}) é uma solução admissível de (3.5).

Finalmente, de $C\tilde{x} \leq C\hat{x}$ temos $Cx^0 - C\tilde{x} \geq Cx^0 - C\hat{x} \Rightarrow \tilde{z} \geq \hat{z}$, em que $\tilde{z}_i \geq \hat{z}_i$, $i = 1, \dots, p$ e, pelo menos, uma das desigualdades é estrita. Somando todos os valores de i obtemos $e^T \tilde{z} > e^T \hat{z}$ e portanto (\hat{x}, \hat{z}) não ótimo para (3.5).

2. Suponhamos que $\mathcal{X}_E \neq \emptyset$, então $\exists \hat{x} \in \mathcal{X}_E$. Pelo teorema 3.13 $\exists \lambda \in \mathbb{R}_{>}^p : \lambda^T C\hat{x} \leq \lambda^T Cx$, $\forall x \in \mathcal{X}$.

Seja $\hat{w} = \lambda / \min_{i \in \{1, \dots, p\}} \{\lambda_i\}$, então $\hat{w} \geq e$ e $\hat{w}^T C\hat{x} \leq \hat{w}^T Cx$, $\forall x \in \mathcal{X}$.

Deste modo \hat{x} é solução ótima de

$$\min\{\hat{w}^T Cx : Ax = b, x \geq 0\} = -\max\{-\hat{w}^T Cx : Ax = b, x \geq 0\}.$$

Logo o dual, isto é,

$$\min\{u^T b : u^T A \geq -\hat{w}^T C\}$$

também tem solução ótima \hat{u} tal que $\hat{u}^T A + \hat{w}^T C \geq 0$. Deste modo (\hat{u}, \hat{w}) é uma solução admissível de (3.7). Uma vez que (3.7) é o dual de (3.5), pelo teorema 3.11, o problema (3.5) não é ilimitado. □

3.3.1 Método da soma ponderada para programas lineares biobjetivo

O método da soma ponderada é um método que se baseia numa resolução geométrica e como tal vamos resolver problemas lineares representados da seguinte forma:

$$\begin{array}{ll} \min & Cx \\ \text{sujeito a} & Ax \leq b \\ & x \geq 0, \end{array}$$

com $C \in \mathbb{R}^{p \times n}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ e $x \in \mathbb{R}^n$.

Como sabemos, pelo teorema 2.15, o conjunto das soluções eficientes é conexo e consegue-se demonstrar que é composto por faces adjacentes de \mathcal{X} , a ideia geral do método consiste em encontrar todos os vértices pertencentes a \mathcal{X}_E . Para isso começamos por encontrar soluções que são extremidades do conjunto conexo das soluções eficientes e de seguida encontrar as restantes soluções eficientes que, de certa forma, se encontram entre as duas iniciais. As soluções iniciais correspondem às soluções eficientes que minimiza cada objetivo, ou seja, para obter a primeira solução temos que começar por resolver LP $\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$ obtendo o conjunto de soluções fracamente eficientes, \mathcal{X}_{wE}^1 , e de seguida resolver o LP $\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$ sujeito a \mathcal{X}_{wE}^1 . Por outras palavras, temos de minimizar a segunda função objetivo no conjunto das soluções ótimas da primeira função objetivo. Para encontrar a segunda solução basta repetir o processo por ordem inversa. Este processo consiste em resolver os seguintes problemas lexicográficos:

$$\begin{array}{ll} \min & (c^2)^T x \\ \text{sujeito a} & (c^1)^T x = f, \\ & Ax \leq b \\ & x \geq 0 \end{array} \quad \text{com } f = \min\{(c^1)^T x : Ax \leq b, x \geq 0\}$$

$$\begin{array}{ll} \min & (c^1)^T x \\ \text{sujeito a} & (c^2)^T x = g, \\ & Ax \leq b \\ & x \geq 0 \end{array} \quad \text{com } g = \min\{(c^2)^T x : Ax \leq b, x \geq 0\}. \quad (3.9)$$

Encontradas as soluções iniciais, x^1 e x^2 , resta agora encontrar as restantes, para tal é necessário resolver o LP(λ) definido em (3.2), com $\lambda = \lambda^* > 0$ um vetor perpendicular à reta formada por $y_1 = Cx^1$ e $y_2 = Cx^2$ pois neste caso $\lambda^* Cx^1 = \lambda^* Cx^2$. Seja x_3 a solução obtida para LP(λ^*), se esta solução se encontrar no segmento de reta $\overline{y_1 y_2}$, obtemos que, como x_3 não é nenhum vértice de \mathcal{X} , todos os pontos no segmento de reta $\overline{y_1 y_2}$ são soluções eficientes. Por outro lado, como x_1 e x_2 são pontos extremos de \mathcal{X}_E concluímos então que $\overline{y_1 y_2}$ contém todas as soluções eficientes e o processo termina. Caso contrário a solução ótima do LP(λ) irá encontrar-se a baixo do segmento $\overline{y_1 y_2}$ e x_3 será uma nova solução eficiente.

O próximo passo é repetir este processo para os segmentos de reta $\overline{y_2 y_3}$ e $\overline{y_1 y_3}$. Enquanto algum novo vértice for encontrado então é necessário repetirmos o processo aos novos segmentos de reta procurando novas soluções eficientes. Este processo termina quando todos os segmentos de reta forem verificados e nenhum ponto for adicionado, ou seja quando todas as soluções eficientes já forem encontradas.

O algoritmo 1 esquematiza o processo anteriormente descrito. Apresenta-se no que se segue um exemplo ilustrativo do algoritmo da soma ponderada para programas lineares biobjetivo.

Algorithm 1: Algoritmo soma ponderada para programação linear biobjetivo.

Input : Dados $A, b, C = (c^1, c^2)$ do LP biobjetivo.

Fase I: Resolver os problemas lexicográficos (3.9).

Fase II: Seja S o conjunto das soluções ordenadas relativamente a x_1

while Um ponto for adicionado **do**

for $i = 1$ até $\dim(S) - 1$ **do**

 Seja $\lambda > 0$ tal que $\lambda^T (C(S(i+1)) - S(i)) = 0$.

 Seja x uma solução do LP(λ).

if x não pertence à reta formada por $C S(i+1)$ e $C S(i)$. **then**

 | Acrescentar x a S , mantendo a ordenação relativamente a x_1

end

end

end

Output : Conjunto S de todos os vértices pertencentes a \mathcal{X}_E .

Exemplo 3.16. Relembremos o programa linear biobjetivo que temos vindo a acompanhar ao longo do trabalho.

$$\begin{aligned} \min \quad & 20x_1 + 10x_2 \\ \min \quad & 60x_1 + 90x_2 \\ \text{s.a} \quad & 5x_1 + 2.5x_2 \geq 5 \\ & x_1 + x_2 \leq 5 \\ & 2x_1 + 2x_2 \geq 3 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Ao resolver o LP $\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)$ obtém-se o conjunto das soluções

$$\mathcal{X}_{wE}^1 = \{(x_1, x_2) : 2x_1 + x_2 = 2, x_1 \in [0, 0.5]\},$$

e todas têm valor mínimo de $\lambda^T Cx = 20$. Resolvendo agora o LP $\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$ s.a $x \in \mathcal{X}_{wE}^1$, isto é $20x_1 + 10x_2 = 20$, obtém-se a primeira solução pretendida, isto é, $x = (0.5, 1)$.

Repetindo este processo, mas agora começando por resolver o LP $\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$ obtém-se o conjunto das soluções

$$\mathcal{X}_{wE}^2 = \{(1.5, 0)\},$$

obtendo 90 como valor mínimo de $\lambda^T Cx$. Ao resolver o LP $\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)$ s.a $x \in \mathcal{X}_{wE}^2$, isto é, $60x_1 + 90x_2 = 90$ obtém-se a segunda solução pretendida $x = (1.5, 0)$.

Fazendo $\lambda^T Cx^1 = \lambda^T Cx^2$, com $\lambda > 0$ vem que $\lambda = (3, 1)$ e o valor ótimo do problema é 180, coincidindo com $\lambda^T Cx^1$ e $\lambda^T Cx^2$, de seguida, como todas as soluções ótimas de LP(λ) se encontram na reta $\overline{Cx^1Cx^2}$, podemos terminar o processo e concluir que o conjunto das soluções eficientes é dado por:

$$\{(x_1, x_2) \in \mathbb{R}^2 : x_1 + x_2 = 1.5, x_1 \geq 0.5, x_2 \geq 0\},$$

este conjunto está representado na figura 3.2.

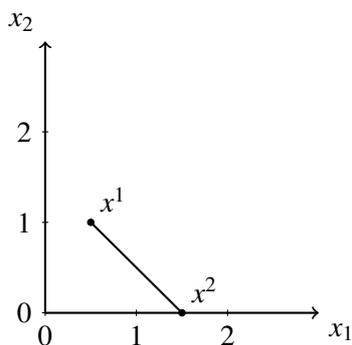


Figura 3.2 Conjunto das soluções eficientes.

Como se pode observar, este exemplo não é muito ilustrativo do algoritmo da soma ponderada, pois o conjunto das soluções eficientes é só um segmento de reta, e por isso o método só necessita de uma iteração para encontrar \mathcal{X}_E .

Com o objetivo de ilustrar todos os passos do algoritmo vejamos o seguinte exemplo.

Exemplo 3.17. Consideremos o seguinte problema linear:

$$\begin{array}{ll} \min & x_1 \\ \min & x_2 \\ \text{s.a} & x_1 + 2x_2 \geq 2. \\ & 2x_1 + x_2 \geq 2 \\ & x_1, x_2 \geq 0 \end{array}$$

Começemos por representar graficamente a região admissível do problema.

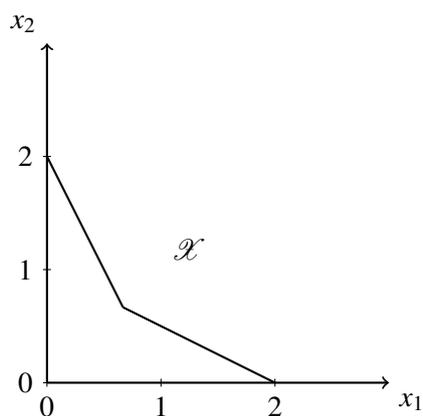


Figura 3.3 Região admissível \mathcal{X} do problema.

Podemos agora começar a Fase I do algoritmo, para tal começemos por calcular f e g , e obter os conjuntos fracamente eficientes:

$$f = \min_{s.a.} \quad x_1 \quad x \in \mathcal{X} \quad = 0 \quad e \quad \mathcal{X}_{wE}^1 = \{(0, 2)\}$$

$$g = \min_{s.a.} \quad x_2 \quad x \in \mathcal{X} \quad = 2 \quad e \quad \mathcal{X}_{wE}^2 = \{(2, 0)\}.$$

Estamos agora em condições de resolver os problemas lexicográficos:

$$\min \quad x_2 \\ \text{sujeito a} \quad x \in \mathcal{X}_{wE}^1,$$

$$\min \quad x_1 \\ \text{sujeito a} \quad x \in \mathcal{X}_{wE}^2,$$

obtendo assim as soluções $x^1 = (0, 2)$ e $x^2 = (2, 0)$.

Tendo as soluções iniciais podemos passar então à Fase II do método. Nesta fase vamos resolver $LP(\lambda)$, para isso necessitamos de calcular λ . Fazendo $\lambda^T Cx^1 = \lambda^T Cx^2$ vem que $\lambda = (0.5, 0.5)$.

Iteração 1: Resolvendo o $LP(\lambda)$ obtemos uma nova solução eficiente, $x^3 = (2/3, 2/3)$, como se pode observar na figura 3.4.

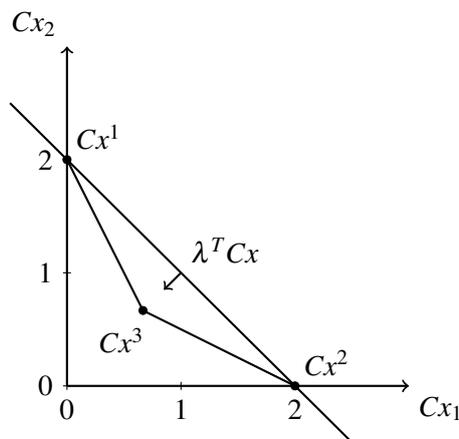


Figura 3.4 Primeira iteração da Fase II do método soma ponderada.

Iteração II: Observando a figura 3.4 conclui-se que se resolvermos o $LP(\lambda)$ mas agora para as retas formadas por Cx^1 e Cx^3 , e Cx^3 e Cx^2 não vamos encontrar novos vértice eficientes.

Como não foram acrescentados novos vértices ao conjunto de soluções eficientes então o algoritmo termina com a lista de vértices eficientes $S = \{(0, 2); (2/3, 2/3); (2, 0); (2, 0)\}$ o que origina o conjunto das soluções eficientes $\{(x_1, x_2); 2x_1 + x_2 = 2 \vee x_1 + 2x_2 = 2, x_1, x_2 \geq 0\}$.

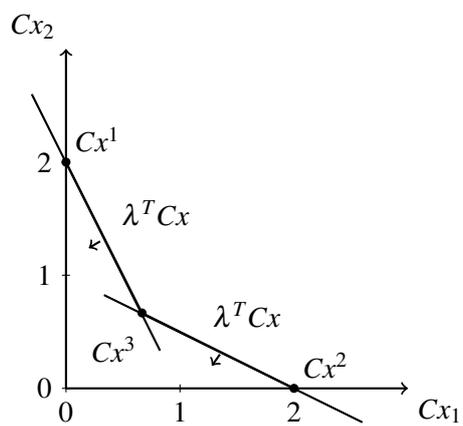


Figura 3.5 Segunda iteração da Fase II do método soma ponderada.

3.4 Método Simplex para programas lineares

O método simplex é um método para resolver programas lineares, determinando soluções ótimas para estes problemas. A sua estratégia consiste em começar com um vértice admissível e, em cada iteração chegar a um dos seguintes casos: um novo vértice que não piore o valor da função objetivo, um raio em que a função objetivo é ilimitada ou ser encontrado um vértice ótimo (solução básica admissível ótima).

Para desenvolver o método simplex para programas lineares multiobjetivo necessitamos, em primeiro lugar, de conhecer o método simplex para programas lineares monobjetivo.

3.4.1 Método Simplex para programas lineares monobjetivo

Para o desenvolvimento deste método precisamos de algumas notações que nos irão acompanhar ao longo desta secção. As demonstrações podem ser consultadas em [2], [4] e [5].

Consideremos o LP

$$\min\{c^T x : Ax = b, x \geq 0\}, \quad (3.10)$$

com $c \in \mathbb{R}^n$, A uma matriz $m \times n$ e $b \in \mathbb{R}^m$, com $b \geq 0$. Assumiremos ainda que $\text{car}(A) = m$ e sem perda de generalidade que $\text{car}(A) = m < n$.

Uma submatriz não-singular $A_{\mathcal{B}} \in \mathbb{R}^{m \times m}$ de A diz-se matriz base, em que \mathcal{B} é o conjunto dos índices de colunas de A que definem $A_{\mathcal{B}}$ e é chamado de base. Seja $\mathcal{N} := \{1, \dots, n\} \setminus \mathcal{B}$ o conjunto dos índices das colunas não-básicas. Uma variável x_i e o índice i são chamados de básicos se $i \in \mathcal{B}$ e não-básicos caso contrário.

Assim, com estas notações podemos dividir A , x e c nas partes básica e não básica usando \mathcal{B} e \mathcal{N} como os índices dos conjuntos. Temos então, $A = (A_{\mathcal{B}}, A_{\mathcal{N}})$, $c^T = (c_{\mathcal{B}}^T, c_{\mathcal{N}}^T)$ e $x^T = (x_{\mathcal{B}}^T, x_{\mathcal{N}}^T)$.

Consequentemente, podemos reescrever as restrições $Ax = b$ em função das partes básica e não-básica,

$$(A_{\mathcal{B}}, A_{\mathcal{N}})(x_{\mathcal{B}}^T, c_{\mathcal{N}}^T)^T = b \Leftrightarrow A_{\mathcal{B}}x_{\mathcal{B}} + A_{\mathcal{N}}x_{\mathcal{N}} = b$$

Uma vez que $A_{\mathcal{B}}$ é invertível, pois é não-singular tem-se,

$$x_{\mathcal{B}} = A_{\mathcal{B}}^{-1}(b - A_{\mathcal{N}}x_{\mathcal{N}}). \quad (3.11)$$

Fixando $x_{\mathcal{N}} = 0$ em (3.11) obtemos $x_{\mathcal{B}} = A_{\mathcal{B}}^{-1}b$, com $b \geq 0$, sendo $(x_{\mathcal{B}}, 0)$ chamada solução básica do LP (3.10). Adicionalmente, se $x_{\mathcal{B}} \geq 0$ é chamada solução básica admissível (BFS) e a base \mathcal{B} é chamada admissível. Por outro lado, se $(x_{\mathcal{B}}, 0)$ é uma BFS ótima então \mathcal{B} é chamada de base ótima.

Relacionemos agora vértices e soluções básicas admissíveis.

Teorema 3.18. *Seja \mathcal{X} um poliedro não vazio e $x \in \mathcal{X}$. Então, as seguintes afirmações são equivalentes:*

1. x é um vértice;

2. x é uma solução básica admissível.

Note-se que várias bases admissíveis podem definir a mesma solução básica admissível e, conseqüentemente, o mesmo vértice. Neste caso as soluções dizem-se degeneradas.

O teorema que se segue justifica o método analisado anteriormente, também conhecido como método simplex para a programação linear, provando que existem soluções básicas admissíveis e soluções básicas admissíveis ótimas.

Teorema 3.19. 1. Se o LP (3.10) é admissível, isto é se $\mathcal{X} \neq \emptyset$, então existe uma solução básica admissível.

2. Se, a função objetivo $c^T x$ é limitada inferiormente em \mathcal{X} , então existe solução básica admissível ótima.

Duas soluções básicas admissíveis dizem-se adjacentes se as bases associadas diferirem apenas numa das variáveis. Vejamos agora como passar de uma solução básica admissível para outra adjacente.

Começemos por calcular o valor da função objetivo usando (3.11) do seguinte modo,

$$\begin{aligned} (c_{\mathcal{B}}^T, c_{\mathcal{N}}^T)(x_{\mathcal{B}}^T, x_{\mathcal{N}}^T)^T &= c_{\mathcal{B}}^T x_{\mathcal{B}} + c_{\mathcal{N}}^T x_{\mathcal{N}} \\ &= c_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} b + (c_{\mathcal{N}}^T - c_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} A_{\mathcal{N}}) x_{\mathcal{N}} \end{aligned} \quad (3.12)$$

Definimos deste modo o vetor $\bar{c}^T = c^T - c_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} A$ que é chamado de vetor dos custos reduzidos e verifica

$$\bar{c}^T = c^T - c_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} A = [0 \quad (c_{\mathcal{N}}^T - c_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} A_{\mathcal{N}})] = [\bar{c}_{\mathcal{B}}^T \quad \bar{c}_{\mathcal{N}}^T]$$

Deste modo, o valor da função objetivo pode escrever-se

$$c^T x = c_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} b + \bar{c}_{\mathcal{N}}^T x_{\mathcal{N}}. \quad (3.13)$$

Seja $(x_{\mathcal{B}}, 0)$ uma solução básica admissível, de (3.12) tem-se que, se existir algum $s \in \mathcal{N}$ tal que $\bar{c}_s < 0$ o valor de $c^T x$ decresce se x_s cresce de 0, no entanto o crescimento de x_s deve ser limitado pela não negatividade $x_{\mathcal{B}} \geq 0$. Resumindo, o nosso próximo objetivo é analisar como devemos alterar o x_s de forma a diminuir o máximo possível o valor da função objetivo.

Para simplificar a notação definimos $\tilde{A} := A_{\mathcal{B}}^{-1} A_{\mathcal{N}}$ e $\tilde{b} := A_{\mathcal{B}}^{-1} b$.

Analisando em (3.11) a variação de uma variável básica x_j , $j \in \mathcal{B}$ com a alteração de x_s , $s \in \mathcal{N}$

$$\begin{aligned} x_j &= (A_{\mathcal{B}}^{-1} b - A_{\mathcal{B}}^{-1} A_{\mathcal{N}} x_{\mathcal{N}})_j, \quad x_{\mathcal{N}} = (0, \dots, x_s, \dots, 0) \\ &= (\tilde{b} - \tilde{A} x_{\mathcal{N}})_j \end{aligned} \quad (3.14)$$

$$= \tilde{b}_j - \tilde{A}_{js} x_s \geq 0 \quad (3.15)$$

em que \tilde{A}_j corresponde à linha j de \tilde{A} e \tilde{A}_{js} o elemento na linha j e coluna s .

Caso exista algum j tal que $\tilde{A}_{js} \leq 0$, a desigualdade (3.15) é válida para todo o $x_s \geq 0$. Isto significa que podemos fazer $x_s \rightarrow +\infty$ e conseqüentemente, segundo a equação (3.13) o valor da função objetivo vai tender para $-\infty$, ou seja o problema é ilimitado. Caso contrário $\tilde{A}_{js} \geq 0, \forall j \in \mathcal{B}$ e x_s deve aumentar de modo a que (3.14) seja satisfeita, ou seja,

$$x_j = \tilde{b}_j - \tilde{A}_{js}x_s \geq 0 \Leftrightarrow \tilde{A}_{js}x_s \leq \tilde{b}_j \Leftrightarrow x_s \leq \frac{\tilde{b}_j}{\tilde{A}_{js}}, \quad \forall j \in \mathcal{B}, \text{ com } \tilde{A}_{js} \neq 0.$$

Como vimos anteriormente queremos atribuir o maior valor admissível de x_s pois esta escolha irá conduzir à maior diminuição da função objetivo. Para isso, como a desigualdade a cima se tem de verificar para todo o $j \in \mathcal{B}$ resta-nos atribuir a x_s o valor da seguinte forma:

$$x_s = \min \left\{ \frac{\tilde{b}_j}{\tilde{A}_{js}}, j \in \mathcal{B}, \tilde{A}_{js} > 0 \right\}. \quad (3.16)$$

Seja $r \in \mathcal{B}$ o índice para o qual o mínimo de (3.16) é alcançado. Denotemos x_s por variável de entrada e x_r variável de saída, pois x_r segundo (3.15) será nula.

O próximo passo será então atualizar a base, pois os índices de \mathcal{B} são agora diferentes. A nova base $\mathcal{B}' = (\mathcal{B} \setminus \{r\}) \cup \{s\}$ vai definir uma nova solução básica admissível $(x_{\mathcal{B}'}, 0)$ com um valor objetivo menor (desde que $\tilde{b}_r \neq 0$).

O teorema que se segue justifica o método analisado anteriormente, também conhecido como método simplex para a programação linear, permitindo caracterizar soluções básicas admissíveis ótimas.

Teorema 3.20. *Seja \mathcal{B} uma base admissível e \bar{x} uma solução básica admissível associada a \mathcal{B} . Se $\bar{c}_{\mathcal{N}} \geq 0$ então \bar{x} é ótima.*

Demonstração. Seja x uma solução admissível e \bar{x} uma solução básica admissível do problema. Escrevendo $x = (x_{\mathcal{B}}, x_{\mathcal{N}})$, vimos em (3.13) que o valor da função objetivo é dada por:

$$c^T x = c_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} b + \bar{c}_{\mathcal{N}} x_{\mathcal{N}}.$$

Uma vez que \bar{x} é uma solução básica admissível então tem-se que $\bar{x}_{\mathcal{B}} = A_{\mathcal{B}}^{-1} b$ e portanto

$$c^T x = c_{\mathcal{B}}^T \bar{x}_{\mathcal{B}} + \bar{c}_{\mathcal{N}} x_{\mathcal{N}}.$$

Como $x_{\mathcal{N}}$ é não negativo pois é a parte não básica da solução admissível e por hipótese $\bar{c}_{\mathcal{N}} \geq 0$, vem

$$c^T x \geq c_{\mathcal{B}}^T \bar{x}_{\mathcal{B}} = c^T \bar{x}.$$

Logo $c^T x \geq c^T \bar{x}$, isto é a solução básica admissível $\bar{x} = (\bar{x}_{\mathcal{B}}, 0)$ é ótima. □

No algoritmo 2 esquematizamos os passos principais do método simplex, para programação linear, que a partir de uma base \mathcal{B} e a solução básica admissível $(x_{\mathcal{B}}, 0)$ inicial, devolve uma base ótima e uma solução básica admissível ótima ou deteta que o problema é ilimitado.

Algorithm 2: Algoritmo simplex para programação linear

Input : Base \mathcal{B} e BFS $(x_{\mathcal{B}}, 0)$.
 Calcular $\tilde{A} := A_{\mathcal{B}}^{-1}A$, $\tilde{b} := A_{\mathcal{B}}^{-1}b$ e $\tilde{c}^T = c^T - c^T A_{\mathcal{B}}^{-1}A$.
while $\{i \in \mathcal{N} : \tilde{c}_i < 0\} \neq \emptyset$ **do**
 escolher $s \in \{i \in \mathcal{N} : \tilde{c}_i < 0\}$.
 if $\tilde{A}_{js} \leq 0$ para algum $j \in \mathcal{B}$ **then**
 STOP, o LP (3.10) é ilimitado.
 else
 escolher $r \in \operatorname{argmin} \left\{ j \in \mathcal{B} : \frac{\tilde{b}_j}{\tilde{A}_{js}}, \tilde{A}_{js} > 0 \right\}$.
 Seja $\mathcal{B} := (\mathcal{B} \setminus \{r\}) \cup \{s\}$ e atualizar \tilde{A} , \tilde{b} e \tilde{c} .
 end
end
Output : Base ótima \mathcal{B} e BFS ótima $(x_{\mathcal{B}}, 0)$.

Falta por último analisar se o algoritmo termina num número finito de iterações. Precisamos agora de analisar os casos em que $\tilde{b}_r > 0$ e $\tilde{b}_r = 0$, separadamente.

Se $\tilde{b}_r > 0$ em todas as iterações, então x_s na nova base é positiva e portanto o valor objetivo decresce de $\tilde{c}_s x_s$ pelo que a nova solução básica é diferente das obtidas anteriormente. Contudo, existem no máximo $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ soluções básicas distintas porque cada solução básica corresponde à escolha de m colunas de A . Assim, conclui-se que o algoritmo termina após um número finito de iterações, obtendo uma solução ótima ou mostrando que o LP é ilimitado.

Se, por outro lado, $\tilde{b}_r = 0$ e, portanto $x_r = 0$, então a nova base vai ter $x_s = 0$. Deste modo, apesar de a base ser diferente, a solução mantém-se inalterada. Para diferenciar estas situações introduzimos a seguinte notação. Se $x_{\mathcal{B}}$ não tem nenhuma componente igual a zero então é chamada de solução básica não-degenerada. Caso contrário, é chamada de solução básica degenerada. É possível que o algoritmo simplex itere entre uma sequência de bases degeneradas sem terminar. Assim, assumiremos que os LPs que consideramos são não-degenerados. Destas notações resultam os seguintes resultados.

Teorema 3.21. *Se todas as bases admissíveis da matriz A estiverem associadas a soluções não-degeneradas, então o algoritmo simplex termina ao fim de um número finito de iterações.*

Demonstração. Pela ideia geral do método simplex sabemos que este método percorre bases admissíveis. Como todas as bases admissíveis da matriz A correspondem a soluções não-degeneradas distintas e, em cada iteração do método o valor da função objetivo decresce, então o método não repete nenhuma solução. Uma vez que o número de soluções básicas admissíveis é finito, então após um número finito obtém-se uma solução básica admissível que é ótima ou a função objetivo diminui ao longo de um raio a partir dele. \square

Lema 3.22. *Seja o LP não-degenerado e \mathcal{B} a base ótima, então $\tilde{c}_{\mathcal{N}} \geq 0$.*

Demonstração. Seja \bar{x} uma solução básica ótima associada à base \mathcal{B} e suponhamos, por absurdo que $\tilde{c}_j < 0$ para algum $j \in \mathcal{N}$. O objetivo do que se segue é encontrar uma nova solução admissível \hat{x} tal que

$$c^T \hat{x} < c^T \bar{x},$$

chegando assim a uma contradição.

Escrevamos \hat{x} na seguinte forma:

$$\hat{x} = (\hat{x}_{\mathcal{B}}, \hat{x}_{\mathcal{N}}), \text{ com } \hat{x}_{\mathcal{N}} = \bar{x}_{\mathcal{N}} + te^j = te^j,$$

onde e^j é o vetor unitário em $\mathbb{R}^{|\mathcal{N}|}$ e t um número positivo escolhido de modo a que \hat{x} seja admissível. Segundo (3.11) $x_{\mathcal{B}}$ pode ser escrito da seguinte forma:

$$\hat{x}_{\mathcal{B}} = \bar{x}_{\mathcal{B}} - tA_{\mathcal{B}}^{-1}A_{\mathcal{N}}e^j.$$

Como o LP é não-degenerado então $\bar{x}_{\mathcal{B}}$ é estritamente positivo pelo que existe $t > 0$ tal que $\hat{x}_{\mathcal{B}} \geq 0$. Deste modo, \hat{x} é admissível. Calculemos então o valor da função objetivo em \hat{x}

$$\begin{aligned} c^T \hat{x} &= c_{\mathcal{B}}^T \hat{x}_{\mathcal{B}} + c_{\mathcal{N}}^T \hat{x}_{\mathcal{N}} \\ &= c_{\mathcal{B}}^T \bar{x}_{\mathcal{B}} - c_{\mathcal{B}}^T t A_{\mathcal{B}}^{-1} A_{\mathcal{N}} e^j + c_{\mathcal{N}}^T t e^j \\ &= c_{\mathcal{B}}^T \bar{x}_{\mathcal{B}} + (-c_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} A_{\mathcal{N}} e^j + c_{\mathcal{N}}^T e^j) t \\ &= c_{\mathcal{B}}^T \bar{x}_{\mathcal{B}} + \bar{c}_j t \\ &< c_{\mathcal{B}}^T \bar{x}_{\mathcal{B}}, \text{ pois } \bar{c}_j < 0 \text{ e } t > 0. \end{aligned}$$

Logo $c^T \hat{x} < c^T \bar{x}$ e portanto chegamos a uma contradição. \square

Apresentamos agora uma tabela que sumariza a informação em cada iteração para o método simplex:

	\bar{c}	$-c_{\mathcal{B}}^T x_{\mathcal{B}}$
\mathcal{B}	$\tilde{\mathcal{A}}$	\tilde{b}

A primeira linha da tabela simplex contem o vetor dos custos reduzidos \bar{c} e o valor negativo da função objetivo restrita à base \mathcal{B} . A segunda linha contem os índices da base \mathcal{B} , bem como a matriz das restrições modificada e à direita o vetor \tilde{b} .

Como podemos deduzir pelo algoritmo apresentado anteriormente, cada iteração do método consiste na determinação das variáveis de entrada e saída, x_s e x_r , e atualizar $\tilde{\mathcal{A}}$, \tilde{b} e \bar{c} , como enunciado anteriormente.

O elemento A_{rs} é denotado por elemento pivot, a linha $A_{r\bullet}$ é a linha pivot, linha da variável, x_r , que se torna não básica. A coluna $A_{\bullet s}$ é a coluna pivot, coluna da variável, x_s , que se torna básica.

Depois de encontrar o pivot aplica-se a eliminação de Gauss para converter a coluna s de $\tilde{\mathcal{A}}$ numa coluna que contém $\tilde{A}_{rs} = 1$, e desta forma consegue-se atualizar $\tilde{\mathcal{A}}$, \tilde{b} e \bar{c} sem ser necessário calcular $A_{\mathcal{B}}^{-1}$ em todas as iterações.

Como foi visto anteriormente, o algoritmo 2 é inicializado com uma base admissível. Podemos determinar uma base inicial resolvendo o seguinte programa linear auxiliar, também designado por método de duas fases:

$$\begin{aligned} \min \quad & e^T z \\ \text{sujeito a} \quad & Ax + z = b \\ & x, z \geq 0. \end{aligned} \quad (3.17)$$

Este LP é sempre admissível e $(x, z) = (0, b)$ é uma solução admissível, porque $b \geq 0$, como foi assumido anteriormente. O teorema que se segue dá-nos uma relação entre os LPs (3.10) e (3.17).

Proposição 3.23. *O LP (3.10) é admissível, isto é $\mathcal{X} \neq \emptyset$, se e só se o LP auxiliar (3.17) tem solução ótima (\hat{x}, \hat{z}) com $\hat{z} = 0$.*

Demonstração. Seja \hat{x} uma solução admissível do LP (3.10). Então (\hat{x}, \hat{z}) com $\hat{z} = 0$ é uma solução admissível do LP auxiliar (3.17) e o valor mínimo da função objetivo é zero. Por outro lado, se o valor ótimo do LP auxiliar é zero então uma solução ótima (\hat{x}, \hat{z}) tem $\hat{z} = 0$ o que implica que \hat{x} é solução admissível do LP (3.10). \square

Desta proposição concluímos também que se o custo ótimo no problema auxiliar for diferente de zero, isto é, $\hat{z} \neq 0$, então o problema original é não admissível.

O teorema que se segue relaciona a álgebra e a geometria da programação linear.

Teorema 3.24. *Se $\mathcal{X} \neq \emptyset$ e o LP (3.10) é limitado, então o conjunto de todas as soluções ótimas do LP ou é o próprio \mathcal{X} ou é uma face de \mathcal{X} .*

Com este teorema conclui-se um breve resumo sobre o método simplex para programação linear monobjetivo, estamos agora em condições de estender o algoritmo 2 para resolver programas lineares biobjetivo.

3.4.2 Método Simplex para programas lineares biobjetivo

Estudemos agora o método simplex para resolver programas lineares com dois objetivos. Consideremos o seguinte programa linear biobjetivo

$$\begin{aligned} \min \quad & ((c^1)^T x, (c^2)^T x) \\ \text{sujeito a} \quad & Ax = b, \\ & x \geq 0. \end{aligned} \quad (3.18)$$

com $b \geq 0$. Pelo teorema 3.13 (Isermann) sabemos que encontrar soluções eficientes de (3.18) é equivalente a resolver o seguinte LP

$$\min \{ \lambda_1 (c^1)^T x + \lambda_2 (c^2)^T x : Ax = b, x \geq 0 \} \quad (3.19)$$

para todo o $\lambda_1, \lambda_2 \in \mathbb{R}_>$. Sem perda de generalidade, podemos dividir a função objetivo por $\lambda_1 + \lambda_2$, obtendo

$$\min \left\{ \frac{\lambda_1}{\lambda_1 + \lambda_2} (c^1)^T x + \frac{\lambda_2}{\lambda_1 + \lambda_2} (c^2)^T x : Ax = b, x \geq 0 \right\},$$

e por sua vez, se fizermos $\lambda = \frac{\lambda_1}{\lambda_1 + \lambda_2} < 1$, então

$$\begin{aligned} \frac{\lambda_2}{\lambda_1 + \lambda_2} &= \frac{\lambda_2 + \lambda_1 - \lambda_1}{\lambda_1 + \lambda_2} \\ &= \frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2} - \frac{\lambda_1}{\lambda_1 + \lambda_2} \\ &= 1 - \lambda, \text{ para todo o } \lambda \in (0, 1). \end{aligned}$$

Assim, podemos assumir $(\lambda_1, \lambda_2) = (\lambda, 1 - \lambda)$ e definir a função objetivo paramétrica

$$c(\lambda) := \lambda c^1 + (1 - \lambda)c^2.$$

Substituindo $c(\lambda)$ na equação (3.19), obtemos o seguinte programa linear paramétrico,

$$\min \{c(\lambda)^T x : Ax = b, x \geq 0\}, \quad (3.20)$$

que precisamos de resolver para todo o $\lambda \in (0, 1)$, ou seja, precisamos de encontrar bases ótimas para todos os λ . No entanto, a mesma base poderá servir para vários λ , pelo que iremos focarmo-nos nos valores de λ relevantes nesta análise. Como tal, comecemos por considerar a base admissível \mathcal{B} e o vetor dos custos reduzidos $\bar{c}(\lambda)$ do programa linear paramétrico,

$$\bar{c}(\lambda) = \lambda \bar{c}^1 + (1 - \lambda)\bar{c}^2. \quad (3.21)$$

Suponhamos que \mathcal{B} é a base ótima do LP (3.20) para algum $\lambda = \hat{\lambda}$ com $\hat{\lambda} \in (0, 1)$. Aplicando o critério de otimalidade do teorema 3.19 alínea 2. e o lema 3.22 tem-se que $\bar{c}(\hat{\lambda}) \geq 0$. Esta informação tem consequências diferentes dependendo de c^1 e c^2 .

Em primeiro lugar, suponhamos que $\bar{c}^2 \geq 0$. Então, por (3.21), vem $\bar{c}(\hat{\lambda}) \geq 0$ para todo o $\lambda \leq \hat{\lambda}$, pois para todo o i

$$\begin{aligned} \bar{c}_i(\lambda) &= \lambda \bar{c}_i^1 + (1 - \lambda)\bar{c}_i^2 \\ &= \lambda(\bar{c}_i^1 - \bar{c}_i^2) + \bar{c}_i^2. \end{aligned}$$

Como $\bar{c}_i(\hat{\lambda}) \geq 0$, $\bar{c}_i(0) \geq 0$ e \bar{c}_i é linear em λ conclui-se que $\bar{c}_i(\lambda) \geq 0, \forall \lambda \in [0, \hat{\lambda}]$.

O segundo caso a considerar é o caso em que existe pelo menos um $i \in \mathcal{N}$ tal que $\bar{c}_i^2 < 0$. Então como $\bar{c}_i(\hat{\lambda}) \geq 0$ e $\bar{c}_i(0) = \bar{c}_i^2 < 0$, pelo teorema do valor intermédio existe um $\lambda \leq \hat{\lambda}$ tal que $\bar{c}(\lambda)_i = 0$, isto é,

$$\bar{c}_i(\lambda) = 0 \Leftrightarrow \lambda(\bar{c}_i^1 - \bar{c}_i^2) + \bar{c}_i^2 = 0 \Leftrightarrow \lambda = \frac{-\bar{c}_i^2}{\bar{c}_i^1 - \bar{c}_i^2}, \quad \bar{c}_i^1 - \bar{c}_i^2 \neq 0.$$

De facto, se assim não fosse, não podia ter $\bar{c}_i(\hat{\lambda}) \geq 0$ e $\bar{c}_i(0) < 0$. Além disso, para garantir que $\lambda > 0$ precisamos que $\bar{c}_i^1 - \bar{c}_i^2 > 0$. Finalmente para garantir que $\lambda \leq 1$ temos que ter $\bar{c}_i^1 \geq 0$ juntando as 2 condições temos $\bar{c}_i^1 \geq 0$. É por isso que aparece esta restrição im I.

Consequentemente se definirmos $\mathcal{I} = \{i \in \mathcal{N} : \bar{c}_i^2 < 0, \bar{c}_i^1 \geq 0\}$ e

$$\lambda' = \max_{i \in \mathcal{I}} \frac{-\bar{c}_i^2}{\bar{c}_i^1 - \bar{c}_i^2}. \quad (3.22)$$

Definindo o λ' podemos concluir que $\hat{\mathcal{B}}$ é uma base ótima para o LP paramétrico (3.20) para todo o $\lambda \in [\lambda', \hat{\lambda}]$, pois $\bar{c}(\hat{\lambda}) \geq 0$. Porém, quando $\lambda < \lambda'$, $\bar{c}(\lambda)$ deixa de ser não negativo e consequentemente \mathcal{B} deixa de ser uma base ótima. Seja s o valor onde o máximo é atingido em (3.22), visto que $\bar{c}_s(\lambda)$ é uma função contínua estritamente crescente $\bar{c}_s(\lambda) < 0$ para $\lambda < \lambda'$. Repetindo o raciocínio do método monobjetivo conseguimos então encontrar uma base associada a uma solução com um valor objetivo inferior, bastando trocar a variável x_s por uma variável x_r , com r escolhido tal como na secção anterior.

No entanto, neste caso para encontrar uma base ótima apenas é necessário realizar uma troca de variável para obter uma base ótima \mathcal{B}' e uma solução ótima $x_{\mathcal{B}'}$. Relembremos que o conjunto de todas as soluções ótimas de $LP(\lambda')$, são soluções eficientes do nosso problema. Posto isto, como $\bar{c}_s(\lambda') = 0$, $x_{\mathcal{B}'}$ continua a ser uma solução ótima para $LP(\lambda')$ e consequentemente vai ser uma base eficiente do MOLP em estudo, isto é $x_{\mathcal{B}'} \in \mathcal{X}_E$.

Adicionalmente, relembremos que o conjunto das soluções eficientes, \mathcal{X}_E é um conjunto conexo, mais concretamente, uma reunião de faces adjacentes de \mathcal{X} . Geometricamente, podemos simplesmente descrever este fato como sendo uma passagem entre vértices adjacentes de \mathcal{X}_E .

Visto isto, para determinar todos os λ necessitamos de um inicial que podemos escolher sendo 1. Se a solução for única então a solução do programa linear biobjetivo (3.18) é eficiente. Caso não seja única, apenas podemos garantir que é uma solução fracamente eficiente do LP biobjetivo. Para encontrar a solução eficiente entre as soluções anteriores, resolve-se o seguinte LP lexicográfico:

$$\begin{aligned} \min \quad & (c^2)^T x \\ \text{sujeito a} \quad & (c^1)^T x = \hat{c} \\ & x \in \mathcal{X}, \end{aligned}$$

onde $\hat{c} = \min\{(c^1)^T x : Ax = b, x \geq 0\}$.

Iterativamente encontramos as variáveis de entrada e de saída e novos valores de λ que satisfaçam (3.22) até que o conjunto \mathcal{I} seja vazio. É importante realçar que inicialmente $\mathcal{I} \neq \emptyset$ pois nós assumimos que não existe nenhuma base ótima de LP(1) e LP(0). Desta forma conseguimos achar bases e consequentemente soluções ótimas para todos os $\lambda \in (0, 1)$, tal como pretendido. Este processo iterativo está representado no Algoritmo 3.

Em cada iteração do método determina-se uma nova base ótima escolhendo uma coluna pivot, que representa a variável de entrada na base, x_s e sendo o índice s , o índice no qual o valor λ' é atingido. A linha pivot, que representa a variável de saída da base, x_r , também é determinada. A variável x_s entra então na base e a troca é efetuada.

Após um número finito de iterações é gerada uma sequência de valores de λ , $1 = \lambda^1 > \lambda^2 \dots > \lambda^l > 0$ e bases ótimas $\mathcal{B}^1, \dots, \mathcal{B}^l$ que define BFSs ótimas de (3.20). Para simplificar a notação,

Algorithm 3: Algoritmo simplex para programação linear biobjetivo

Input : Dados $A, b, C = (c^1, c^2)$ do LP biobjetivo, com $b_i \geq 0$.

Fase I: Resolver o LP (3.17) usando o algoritmo simplex (2).

if o valor ótimo é positivo **then**

| STOP, $\mathcal{X} = \emptyset$.

else

| Seja \mathcal{B} a base ótima.

end

Fase II: Resolver o LP (3.20) para $\lambda = 1$ começando com a base \mathcal{B} encontrada na Fase I chegando à base ótima $\tilde{\mathcal{B}}$. Calcular \tilde{A}, \tilde{b} e \tilde{c} .

Fase III: **while** $\mathcal{J} = \{i \in \mathcal{N} : \tilde{c}_i^2 < 0, \tilde{c}_i^1 \geq 0\} \neq \emptyset$ **do**

$$\lambda := \max_{i \in \mathcal{J}} \frac{-\tilde{c}_i^2}{\tilde{c}_i^1 - \tilde{c}_i^2}.$$

$$s \in \operatorname{argmax} \left\{ i \in \mathcal{J} : \frac{-\tilde{c}_i^2}{\tilde{c}_i^1 - \tilde{c}_i^2} \right\}.$$

$$r \in \operatorname{argmin} \left\{ j \in \mathcal{B} : \frac{\tilde{b}_j}{\tilde{A}_{js}}, \tilde{A}_{js} > 0 \right\}.$$

Seja $\mathcal{B} := (\mathcal{B} \setminus \{r\}) \cup \{s\}$ e atualizar \tilde{A}, \tilde{b} e \tilde{c} .

end

Output : sequência dos valores de λ e sequência de BFSs ótimas.

iremos utilizar $\lambda^{l+1} = 0$. Desta forma obtém-se que todo o λ , \mathcal{B}^i é a base ótima de (3.20) para todo o $\lambda \in [\lambda^{i+1}, \lambda^i]$, $i = 1, \dots, l$. A solução básica admissível ótima x^i é a solução ótima de (3.20) para todo o $\lambda \in [\lambda^{i+1}, \lambda^i]$, $i = 1, \dots, l$.

Desta forma, conclui-se que para cada λ^i , $2 \leq i \leq l$ existem duas soluções básicas admissíveis ótimas, x^i e x^{i-1} .

Do ponto de vista geométrico, obtemos que, todas as soluções admissíveis nas faces que unem os pontos extremos são soluções ótimas de (3.20) com $\lambda = \lambda^i$.

Pelo teorema de Isermann 3.13 sabemos que uma solução eficiente do programa linear bicritério (3.18) se e só se for solução ótima do LP (λ), para algum $\lambda \in (0, 1)$. Contudo, não há garantia que todas as soluções eficientes possam ser construídas através das bases $\mathcal{B}^1, \dots, \mathcal{B}^l$ obtidos pelo algoritmo 3. De facto, este algoritmo encontra, para cada $y \in \mathcal{Y}_N$ um $x \in \mathcal{X}_E$ tal que $Cx = y$, mas se existirem várias soluções em \mathcal{X}_E com o mesmo valor no espaço dos critérios, o algoritmo só encontrará um deles. Assim, o algoritmo 3 encontra um caminho através de \mathcal{X}_E que trace \mathcal{Y}_N mas nem todas as bases definem soluções eficientes.

Apresenta-se no que se segue um exemplo ilustrativo do algoritmo simplex para programas lineares biobjetivo.

Exemplo 3.25. Relembremos o programa linear biobjetivo que temos vindo a acompanhar ao longo de todo o trabalho.

$$\begin{aligned} \min \quad & 20x_1 + 10x_2 \\ \min \quad & 60x_1 + 90x_2 \\ \text{s.a} \quad & 5x_1 + 2.5x_2 \geq 5 \\ & x_1 + x_2 \leq 5 \\ & 2x_1 + 2x_2 \geq 3 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Comecemos por representar o programa linear na forma (3.10):

$$\begin{aligned} \min \quad & 20x_1 + 10x_2 \\ \min \quad & 60x_1 + 90x_2 \\ \text{s.a} \quad & 5x_1 + 2.5x_2 - x_3 = 5 \\ & x_1 + x_2 + x_4 = 5 \\ & 2x_1 + 2x_2 - x_5 = 3 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{aligned}$$

Como $b_i > 0$ podemos então resolver o programa linear paramétrico

$$\begin{aligned} \min \quad & \lambda(20x_1 + 10x_2) + (1 - \lambda)(60x_1 + 90x_2) \\ \text{s.a} \quad & 5x_1 + 2.5x_2 - x_3 = 5 \\ & x_1 + x_2 + x_4 = 5 \\ & 2x_1 + 2x_2 - x_5 = 3 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0, \end{aligned}$$

isto é,

$$\begin{aligned} \min \quad & (-40\lambda + 60)x_1 + (-80\lambda + 90)x_2 \\ \text{s.a} \quad & 5x_1 + 2.5x_2 - x_3 = 5 \\ & x_1 + x_2 + x_4 = 5 \\ & 2x_1 + 2x_2 - x_5 = 3 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{aligned}$$

Para inicializar o método simplex biobjetivo necessitamos de ter uma solução básica admissível inicial. Essa solução básica pode ser encontrada resolvendo o seguinte PL monocritério o qual corresponde à fase I.

$$\begin{aligned} \min \quad & e^T z \\ \text{s.a} \quad & 5x_1 + 2.5x_2 - x_3 + z_1 = 5 \\ & x_1 + x_2 + x_4 + z_2 = 5 \\ & 2x_1 + 2x_2 - x_5 + z_3 = 3 \\ & x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3 \geq 0. \end{aligned}$$

Assim a solução básica admissível que vai inicializar o método simplex monobjetivo para resolver o programa linear auxiliar é $(0, 0, 0, 0, 0, 5, 5, 3)$. Mais uma vez, em cada iteração do método vão ser calculados os valores de s e r .

Iteração I:

\bar{c}	-8	-5.5	1	-1	1	0	0	0	-13
z_1	5	2.5	-1	0	0	1	0	0	5
z_2	1	1	0	1	0	0	1	0	5
z_3	2	2	0	0	-1	0	0	1	3

$s = 1, r = 6.$

Iteração II:

\bar{c}	0	-1.5	-0.6	-1	1	1.6	0	0	-5
x_1	1	0.5	-0.2	0	0	0.2	0	0	1
z_2	0	0.5	0.2	1	0	-0.2	1	0	4
z_3	0	1	0.4	0	-1	-0.4	0	1	1

$s = 2, r = 8.$

Iteração III:

\bar{c}	0	0	0	-1	-0.5	1	0	1.5	-3.5
x_1	1	0	-0.4	0	0.5	0.4	0	-0.5	0.5
z_2	0	0	0	1	0.5	0	1	-0.5	3.5
x_2	0	1	0.4	0	-1	-0.4	0	1	1

$s = 4, r = 7.$

Iteração IV:

\bar{c}	0	0	0	0	0	1	1	1	0
x_1	1	0	-0.4	0	0.5	0.4	0	-0.5	0.5
x_4	0	0	0	1	0.5	0	1	-0.5	3.5
x_2	0	1	0.4	0	-1	-0.4	0	1	1

O conjunto $\{i \in \mathcal{N} : \bar{c}_i < 0\} = \emptyset$ logo o algoritmo termina.

Seguimos então para a Fase II, onde queremos encontrar uma base ótima para $\lambda = 1$, isto é, pretende-se resolver o problema monocritério, em que a função objetivo corresponde a c^1 , para tal construímos a seguinte tabela:

\bar{c}^1	0	0	4	0	0	-20
x_1	1	0	-0.4	0	0.5	0.5
x_4	0	0	0	1	0.5	3.5
x_2	0	1	0.4	0	-1	1

Esta Fase termina ao fim, apenas, de uma iteração pois o conjunto $\{i \in \mathcal{N} : \bar{c}_i < 0\} = \emptyset$, obtendo a base ótima é $\mathcal{B} = \{1, 4, 2\}$ e a solução básica admissível ótima é $x = (0.5, 1, 0, 3.5, 0)$.

Assim, estamos em condições de avançar para a Fase III do método simplex para programação linear biobjetivo. Em todas iterações do método vão ser calculados: o conjunto \mathcal{J} , λ' , r , s e a tabela simplex.

Iteração I:

\bar{c}^1	0	0	4	0	0	-20
\bar{c}^2	0	0	-12	0	60	-120
x_1	1	0	-0.4	0	0.5	0.5
x_4	0	0	0	1	0.5	3.5
x_2	0	1	0.4	0	-1	1

$\lambda = 1$, $\bar{c}(\lambda) = (-40 + 60)x_1 + (-80 + 90)x_2 = 20x_1 + 10x_2 = (20, 10, 0, 0)$, $\mathcal{B}^1 = \{1, 4, 2\}$, $x^1 = (0.5, 1, 0, 3.5, 0)$, $\mathcal{J} = \{3\}$, $\lambda' = \max\{\frac{12}{4+12}\} = \frac{3}{4}$, $s = 1$, $r = 2$.

Iteração II:

\bar{c}^1	0	-10	0	0	10	-30
\bar{c}^2	0	30	0	0	30	-90
x_1	1	1	0	0	-0.5	1.5
x_4	0	0	0	1	0.5	3.5
x_3	0	2.5	1	0	-2.5	2.5

Na segunda iteração, fazendo $\lambda = \frac{3}{4}$ obtemos: $\bar{c}(\lambda) = (-40 \times \frac{3}{4} + 60)x_1 + (-80 \times \frac{3}{4} + 90)x_3 = (30, 0, 30, 0)$, $\mathcal{B}^2 = \{1, 4, 3\}$, $x^2 = (1.5, 0, 2.5, 3.5, 0)$, $\mathcal{J} = \emptyset$ (pelo que não é possível calcular λ').

O algoritmo termina e devolve os valores $\lambda^1 = 1$, $\lambda^2 = \frac{3}{4}$, $\lambda^3 = 0$ e as soluções básicas admissíveis x^1, x^2 .

Para sumarizar, temos os seguintes resultados:

- Base $\mathcal{B}^1 = \{1, 4, 2\}$ e BFS $x^1 = (0.5, 1, 0, 3.5, 0)$ é admissível para $\lambda \in [3/4, 1]$.
- Base $\mathcal{B}^2 = \{1, 4, 3\}$ e BFS $x^2 = (1.5, 0, 2.5, 3.5, 0)$ é admissível para $\lambda \in [0, 3/4]$.
- Os vetores objetivo das duas soluções básicas admissíveis são $Cx^1 = (2, 120)$ e $Cx^2 = (30, 90)$.

3.4.3 Método simplex para programas lineares multiobjetivo

Nesta secção pretende-se transmitir uma ideia muito resumida sobre a generalização do método simplex para programas lineares com $p > 2$ objetivos, este estudo pode ser visto com maior detalhe em [4]. Um MOLP com dois objetivos pode ser resolvido usando o algoritmo 3, apresentado na secção anterior. No entanto com três ou mais objetivos este método não resulta, pois lidamos com pelo menos dois parâmetros na função objetivo $c(\lambda)$.

Note-se que mesmo no caso monocritério, o algoritmo simplex, pode requerer um número elevado de pivots e portanto tal situação também pode ocorrer nos casos bi e multiobjetivo. Além disso, à medida que a dimensão aumenta, o número de pontos extremos eficientes pode crescer exponencialmente, o que faz com que o problema seja cada vez mais difícil a nível computacional.

Começemos por analisar o nosso problema de forma análoga ao caso biobjetivo, consideremos o MOLP geral

$$\begin{aligned} \min \quad & Cx \\ \text{sujeito a} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{3.23}$$

com $C \in \mathbb{R}^{p \times n}$, $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$.

Para $\lambda \in \mathbb{R}_{>}^p$ denotemos por $LP(\lambda)$ o programa linear de soma ponderada

$$\min\{\lambda^T Cx : Ax = b, x \geq 0\}. \tag{3.24}$$

Usemos ainda a notação $\bar{C} = C - C_{\mathcal{B}}A_{\mathcal{B}}^{-1}A$ para a matriz de custos reduzidos em relação à base \mathcal{B} e $R := \bar{C}_N$ a parte não básica da matriz de custos reduzidos.

Os resultados que irão ser apresentados são analogias para problemas multiobjetivo de resultados conhecidos da programação linear ou extensões necessárias para lidar com o aumento da complexidade da programação linear multiobjetivo em relação à monobjetivo.

Definição 3.26. *Uma base admissível \mathcal{B} é chamada eficiente se \mathcal{B} é uma base ótima do $LP(\lambda)$ para algum $\lambda \in \mathbb{R}_{>}^p$.*

Da definição anterior surge o seguinte lema que nos garante a existência de soluções básicas admissíveis eficientes.

Lema 3.27. *Se $\mathcal{X}_E \neq \emptyset$ então \mathcal{X} tem uma solução básica admissível eficiente.*

Denotemos o processo de mudança de base através do uso de pivots por pivotação. Consideremos ainda que um pivot é admissível se a solução obtida depois de um passo de pivot é admissível mesmo que o elemento pivot \tilde{A}_{rs} seja negativo. Desta forma podemos então definir o conceito de bases adjacentes.

Definição 3.28. *Duas bases \mathcal{B} e $\hat{\mathcal{B}}$ são chamadas adjacentes se uma poder ser obtida através da outra por um único passo de pivotação.*

Vejamos agora como pode ser feita a pivotação entre bases eficientes.

Definição 3.29. *Seja \mathcal{B} uma base eficiente.*

1. *Uma variável x_j , $j \in \mathcal{N}$, é designada por variável não-básica eficiente em \mathcal{B} se existir um $\lambda \in \mathbb{R}_>^p$ tal que $\lambda^T R \geq 0$ e $\lambda^T r^j = 0$, onde r^j é a coluna de R correspondente à variável x_j .*
2. *Seja x_j uma variável não-básica eficiente, então um pivot admissível de \mathcal{B} com x_j entrando na base é designado um pivot eficiente com respeito a \mathcal{B} e x_j .*

O sistema $\lambda^T R \geq 0$, $\lambda^T r^j = 0$ é a forma geral das equações usadas para calcular os valores críticos de λ na programação linear paramétrica que é usada na dedução de (3.22), é usada para escolher s tal que $\bar{c}(\lambda) \geq 0$, $\bar{c}_s(\lambda) = 0$. O resultado que se segue garante a existência de variáveis não-básicas eficientes.

Proposição 3.30. *Seja \mathcal{B} uma base eficiente. Existe uma variável não-básica eficiente em \mathcal{B} .*

A técnica por detrás do algoritmo simplex consiste em movermo-nos de bases eficientes em bases eficientes até que todas sejam encontradas.

Lema 3.31. *Seja \mathcal{B} uma base eficiente e x_j uma variável não-básica eficiente. Então qualquer pivot eficiente de \mathcal{B} leva a uma base eficiente adjacente $\hat{\mathcal{B}}$.*

Notemos que é possível restringirmos-nos apenas a bases adjacentes, isto é, bases eficientes são conexas. Definamos agora o conceito de bases eficientes conexas.

Definição 3.32. *Duas bases eficientes \mathcal{B} e $\hat{\mathcal{B}}$ são conexas se uma poder ser obtida a partir da outra utilizando apenas pivots eficientes.*

No teorema que se segue, garante-se que todas as bases eficientes são conexas, sendo este resultado a base do algoritmo simplex multiobjetivo.

Teorema 3.33. (Steuer(1985)) *Todas as bases eficientes são conexas.*

O teorema 3.33 não garante, por si só, a existência de variáveis não básicas eficientes, esta garantia é dada pela suposição não trivial feita no Capítulo 2, $\bigcap_{i=1}^p \mathcal{X}_i = \emptyset$.

Os resultados até agora enunciados permitem-nos mover de base eficiente em base eficiente. Para formular o algoritmo Simplex multiobjetivo necessitamos de uma base eficiente para inicializar o método.

Para o MOLP

$$\min\{Cx : Ax = b, x \geq 0\}$$

apenas um dos seguintes casos pode ocorrer:

- (i) O MOLP é não admissível, isto é, $\mathcal{X} = \emptyset$,
- (ii) é admissível ($\mathcal{X} \neq \emptyset$) mas não tem soluções eficientes ($\mathcal{X}_E = \emptyset$), ou
- (iii) é admissível e tem soluções eficientes, isto é $\mathcal{X}_E \neq \emptyset$.

O algoritmo Simplex multiobjetivo lida com estas situações nas seguintes fases:

- Fase I Determina uma solução básica admissível inicial ou para com a conclusão que $\mathcal{X} = \emptyset$.
Esta fase não envolve a matriz de função objetivo C e o LP (3.17) pode ser usado.
- Fase II Determina uma base eficiente inicial ou para com a conclusão que $\mathcal{X}_E = \emptyset$.
- Fase III Pivota entre bases eficientes até determinar todas as bases eficientes.

Capítulo 4

Comparação dos métodos

O estudo até agora feito sobre programação linear multiobjetivo tem sido sobretudo teórico, sendo esse o nosso principal objetivo. Até agora estudamos dois métodos para resolver problemas lineares biobjetivo, o método da soma ponderada e o método simplex.

Depois de um desenvolvimento teórico surgem naturalmente algumas perguntas: Qual o melhor método? Qual dos métodos encontra soluções mais rapidamente? O número de iterações dos métodos é o mesmo? Com intuito de dar resposta a algumas destas perguntas implementamos ambos os métodos, em *Matlab*, de forma a fazer um breve estudo computacional. Para fazer a comparação pretendida usou-se o seguinte problema linear biobjetivo: dado $n \in \mathbb{N}$

$$\begin{array}{ll} \min & x_1 \\ \min & x_2 \\ \text{sujeito a} & kx_1 + (n-k)x_2 \geq k(n-k), \quad k = 0, \dots, n \\ & x_1, x_2 \geq 0. \end{array}$$

Este problema foi escolhido de forma a poder ser controlado o número de soluções à priori e por ser simples de visualizar graficamente. Como se pode observar na figura 4.1, para o caso em que $n = 6$, tem-se 6 soluções e estas são relativamente fáceis de ser visualizadas. Para realizar um estudo computacional mais rigoroso, deveríamos considerar vários tipos de problemas, mas lembremos que não foi este o nosso principal objetivo.

Depois da implementação dos dois métodos em *Matlab* contabilizou-se o número de iterações e o tempo médio de 20 execuções de cada um dos métodos, para cada valor de n . Deste modo obtiveram-se os seguintes gráficos: 4.2, para o número de iterações, e 4.3 para o tempo de execução.

Como seria de esperar, no método simplex o número de iterações é sempre igual ao n . Relativamente ao método da soma ponderada o número de iterações é sempre inferior ao número de iterações do outro método, sendo que quando n é superior a 10 o número de iterações tende a ser entre 5 e 7.

Relativamente ao tempo de compilação de cada um dos métodos, os resultados computacionais mostram que o método da soma ponderada é muito mais lento a compilar que o método simplex.

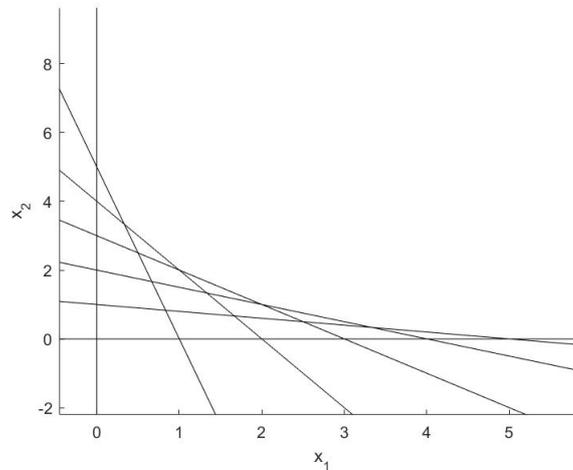


Figura 4.1 Resolução gráfica do PL para $n = 6$.

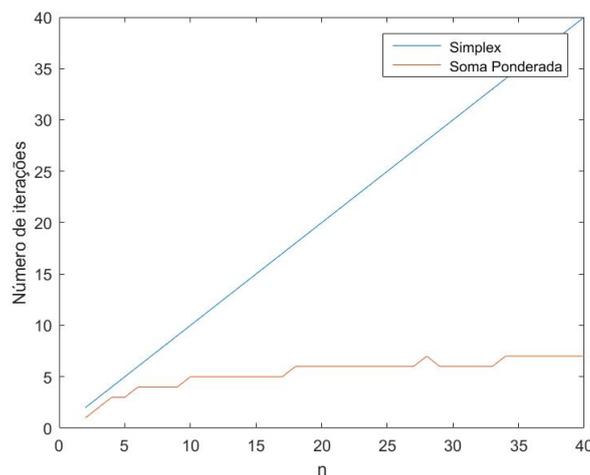


Figura 4.2 Número de iterações até $n = 40$.

Respondendo à primeira pergunta que enunciei no início desta secção, o método simplex é melhor que o método da soma ponderada pois o tempo de compilação é significativamente inferior. Embora o método simplex faça mais iterações que o método da soma ponderada, o parâmetro mais importante a nível prático para um programador no estudo computacional de métodos é o tempo. Este resultado era expectável, sendo esta uma das razões para o método simplex ser o mais utilizado.

Relembramos que estas conclusões são válidas apenas para o tipo de problemas que foi resolvido. Para tirar conclusões mais gerais, precisaríamos realizar um estudo mais aprofundado e fazer mais exemplos. Salienta-se que uma implementação mais eficiente do método da soma ponderada permitiria diminuir o tempo de execução. Contudo, não é de esperar que este seja mais rápido do que o método simplex. De facto, para obter uma nova solução eficiente usando o método da soma ponderada temos que resolver PLs desde o início enquanto que o método simplex efetua apenas uma pivotação.

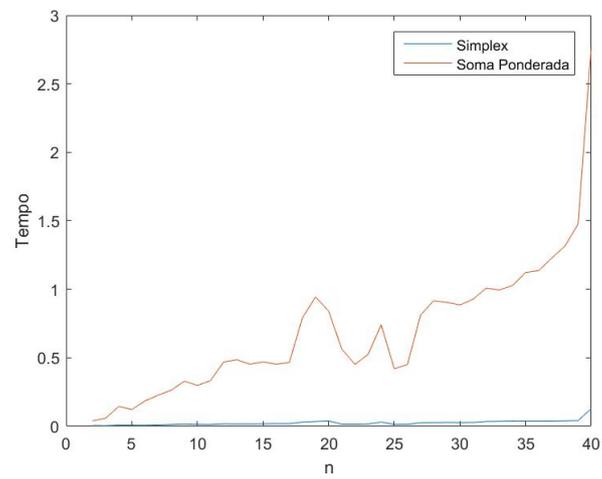


Figura 4.3 Tempo até $n = 40$.

Bibliografia

- [1] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.
- [2] João Namorado Clímaco, Carlos Henggeler Antunes, and Maria João Gomes Alves. *Programação linear multiobjetivo: do modelo de programação linear clássico à consideração explícita de várias funções objetivo*. Imprensa da Universidade de Coimbra/Coimbra University Press, 2003.
- [3] Matthias Ehrgott. *Multiple criteria optimization: classification and methodology*. Shaker, 1997.
- [4] Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2013.
- [5] Dinh The Luc. *Multiobjective Linear Programming: An Introduction*. Springer International Publishing, 2016.
- [6] Denis Trystram. *Multi-Criteria Optimization and its Application to Multi-Processor Embedded Systems*. PhD thesis, université de Provence, 2011.

Apêndice A

Códigos

Dados

```
function[Ad,bd,Ai,bi,C]= dados(opcao,n)
% função que recebe os dados o problema a minimizar (A,b,C)
% Ad=matriz das desigualdades, bd=vetor das desigualdades, Ai=matriz das
% igualdades, bi=vetor das igualdades

if (opcao==1)
Ad=[0,1; 3,-1];
bd=[3;6];
C=[3,1;-1,-2];
Ai=[];
bi=[];
elseif(opcao==2)
Ad=[1,2; 1,0];
bd=[10;5];
C=[-2,1;-4,-3];
Ai=[];
bi=[];
elseif(opcao==3)% programa linear usado no exemplo da secção "Método Simplex
%para programas lineares biobjetivo"
Ad=[-5,-2.5; 1,1;-2,-2];
bd=[-5;5;-3];
C=[20,10;60,90];
Ai=[];
bi=[];
elseif(opcao==4)
Ad=[-5,-2.5; 1,1;-2,-2];
bd=[-5;5;-3];
C=[20,10;60,90];
```

```

Ai=[2 4];
bi=[2];
elseif(opcao==5)% problema linear usado no Capítulo "Comparação dos métodos"
C=[1,0;0,1];
k=0:n;
Ad=-[k' (n-k')];
bd=-[k'.*(n-k')];
Ai=[];
bi=[];
end

```

```

% função que define N,A-tilde, cc=custos reduzidos, b-tilde. Estas notações
% vão ser usadas nos algoritmos monobjetivo e simplexbi,
function [N, AA, cc, bb] = notacoes( A,c,b,B)
n = size(A, 2);
N=setdiff(1:n,B);
A_B=A(:,B);
A_N=A(:,N);
c_B=c(:,B);
c_N=c(:,N);
in=A_B^-1;
AA=in*A_N;
cc=c_N-c_B*AA;
bb=in*b;
end

```

Método da Soma Ponderada

```

% função que resolve os problemas lexicográficos
function [x,valorND]=program (i,C,Ad,bd, Ai,bi, lb,ub)
[~,g]=linprog(C(i,:),Ad,bd,Ai,bi,lb,ub); % procura soluções que minimizem o objetivo i
A2=[Ad; C(i,:)];
bb=[ bd; g];
[x,f]= linprog(C(3-i,:),A2,bb,Ai,bi,lb,ub); % das soluções que minimizam o objetivo i
%procuramos as que minimizam também o outro objetivo
valorND = [g f]; % os 2 valores não dominados
end

```

```
% programa que nos dá uma solução do método paramétrico
function [x,f]= parametrico(lambda,C,A,b, Aeq,beq, lb,ub)
c=lambda*C;
[x,f]=linprog(c,A,b, Aeq,beq, lb,ub);
end

% função que faz a Fase II do método da soma ponderada
function [s,l] = paragem( s,l,C,A,b, Aeq,beq, lb,ub)
n=size(s,1)-1;
epsilon=10^-6;

% l é um vetor de 0 e 1, em que 0 diz que 2 vértices não foram analisados e 1
% diz que foram e que não é inserido mais nenhum ponto
i=1;
while i<=n % analisa cada dois pontos consecutivos e vê se existe um mínimo
% pelo método paramétrico
if l(i)==0

% calculo do vetor lambda
v=C*(s(i+1,:)-s(i,:))'; % vetor ortogonal a lambda
lambda(i,:)=[-v(2)' v(1)'];
lambda(i,:)=lambda(i,:)/norm(lambda(i,:));% normalizar, para não ficar
%um valor muito grande
p =(parametrico(lambda(i,:),C,A,b, Aeq,beq, lb,ub))'; % calcular o ponto do
% método paramétrico e passa-lo para o espaço dos objetivos
if abs(lambda(i,:)*C*(p-s(i+1,:))')< epsilon
l(i)=1;
else % se houver um mínimo adiciona ao vetor no meio dos dois pontos
s1=[s(1:i,:); p];
s2=[p; s(i+1:n+1,:)];
s=[s1; s2(2:end,:)];
l1=[l(1:i),0];
l2=[0, l(i+1:end)];
l=[l1,l2(2:end)];
i=i+2;
n=n+1;
end
```

```

else
i=i+1;
end
end

end

```

```

% função que resolve o algoritmo completo da soma ponderada
function [s, cont] = somaponderada(C,Ad,bd, Ai,bi)
lb=[0;0];
ub=[inf;inf];
% duas soluções e os respectivos valores não dominados
[x1,~]=program (1,C,Ad,bd, Ai,bi, lb,ub);
[x2,~]=program (2,C,Ad,bd, Ai,bi, lb,ub);

% ordenar os valores não dominados, por ordem crescente da primeira
% coordenada
cont=1;
if x1(1)<=x2(1)
s= [x1';x2'];
else
s= [x2';x1'];
end
l=[0];
% o novo vetor dos pontos não dominados
[s2,l]=paragem( s,l,C,Ad,bd, Ai,bi, lb,ub);
% repetição do processo enquanto os vetores forem diferentes
while size(s2,1)~=size(s,1)
s=s2;
[s2,l]=paragem( s,l,C,Ad,bd, Ai,bi, lb,ub);
cont=cont+1;
end

epsilon=10^-6;
i=2;
while i<(size(s,1)-1)
slo1=(s(i+1,2)-s(i-1,2))/(s(i+1,1)-s(i-1,1));
slo2=(s(i+1,2)-s(i,2))/(s(i+1,1)-s(i,1));
if slo2-slo1<epsilon
s=[s(1:i-1,:);s(i+1:end,:)];

```

```

else
i=i+1;
end
end

```

Método Simplex

```

% função que transforma as restrições todas em igualdades do tipo Ax=b
function [ A,b,C] = igualrest( Ad,bd,Ai,bi,C)

```

```

[n,m]=size(Ad);% n=número de linhas e m=número de colunas

```

```

for i=1:n
if bd(i)<0 % nos bi<0 transforma-os em positivos, multiplicando toda a
% restrição por (-1)
bd(i)=-bd(i);
Ad(i,:)=-Ad(i,:);
Ad=[Ad zeros(n,1)];% adiciona variáveis auxiliares para termos igualdades
Ad(i,m+i)=-1;
else
Ad=[Ad zeros(n,1)];
Ad(i,m+i)=1;
end
end

```

```

aux=find(bi<0);% encontra as posições dos elementos de bi<0
bi(aux)=-bi(aux); % multiplica cada restrição por (-1)
Ai(aux,:)=-Ai(aux,:);

```

```

if ~isempty(Ai)
b=[bd;bi];
Ai=[Ai zeros(size(Ai, 1), n)];
A=[Ad;Ai];
else
b=bd;
A=Ad;
end

```

```

C=[C zeros(size(C, 1), n)];
end

```

```
% função que devolve a solução básica admissível e a base admissível, B, associada
function [ B,BFS,fval ] =BFS( A,b)
```

```
[m,n]=size(A);% m o número das restrições e n o número das variáveis
B= (n+1:n+m);
A_Z=eye(m);
AUX=[A A_Z];% acrescenta as variáveis auxiliares z
c=[zeros(1,n) ones(1,m)];% c=e^T*z
[B,BFS]=simplexmono(B,AUX,b,c);
fval=c*BFS';
BFS=BFS(1:n);
end
```

```
% função que encontra o r, este vai ser usado nos dois algoritmos simplex
function r = escolherR(indicesBase, s, A_tilde, b_tilde)
indexJ = find(A_tilde(:,s)>0);% conjunto das posições de j em B tal que A_js>0
[~,indexMin]=min(b_tilde(indexJ)./A_tilde(indexJ,s));% encontra a posição de r em B
r=indicesBase(indexJ(indexMin));% encontra r
end
```

```
% algoritmo simplex monobjetivo
function [ OB,OBFS] = simplexmono( B,A,b,c )
% OB= base ótima, OBFS=solução básica admissível ótima
[N, AA, cc,bb] = notacoes( A,c,b,B);
s=find(cc<0,1);
while ~isempty(s)
aux=find(AA(:,s)>0,1);
if isempty(aux)
disp('O LP e ilimitado');
return;
else
r = escolherR(B, s, AA, bb);
i=find(B==r,1);% i é a linha do pivot
B(i)=N(s);% troca r por s
[N, AA, cc,bb] = notacoes( A,c,b,B);
```

```

s=find(cc<0,1);
end
end
OB=B;
OBFS=zeros(1,size(A,2));
OBFS(B)=bb;

% função que resolve o método simplex biobjetivo
function[lambda, CX,x_B,cont]=simplexbi(A,b,C)
epsilon = 10^-10;

[m,n]=size(A);% m o número das restrições e n o número das variáveis
[B,~,fval]=BFS(A,b);

if fval<=epsilon

lambda=1;
c = C(1,:); %c = [lambda (1-lambda)]*C
[B,OBFS] = simplexmono( B,A,b,c );
x_B=OBFS;

[N, AAN, CCN, bb] = notacoes( A,C,b,B);
C3=-C(:,B)*x_B(B)';
obj=C3';

CC=zeros(2,n);
CC(:,N)=CCN;
AA=zeros(length(B),n);
AA(:,N)=AAN;
for i=B
j=find(B==i,1);
AA(j,i)=1;
end

iv=find((CC(2,:)<0)&(CC(1,*)>=0));
cont=1;% conta o numero de iterações
while ~isempty(iv) %(se iv não for vazio)
[lambda1,s]=max(-CC(2,iv)./(CC(1,iv)-CC(2,iv)));% encontra os valores para lambda

```

```

e o argumento mínimo para s
s=iv(s(1));% s(1) dá-nos o argumento máximo, no caso de haverem dois máximos
    escolhe o primeiro
lambda=[lambda; lambda1];% guardar os lambdas
r = escolherR(B, s, AA, bb);
% actualização das variáveis
i=find(B==r,1);% i é a linha do pivot
B(i)=s;% troca r por s
bb(i)=bb(i)/ AA(i,s);
AA(i,:)=AA(i,)/AA(i,s);
aux1=(AA(i,s)*bb(i))/AA(i,s);
C3(1)=C3(1)-CC(1,s)*aux1;
C3(2)=C3(2)-CC(2,s)*aux1;
aux2=AA(i,)/AA(i,s);
CC(1,:)=CC(1,)-CC(1,s)*aux2;
CC(2,:)=CC(2,)-CC(2,s)*aux2;
obj=[obj;C3'];
aux=zeros(1,n);
for j=1:m
    if j~=i
        bb(j)=bb(j)-AA(j,s)*aux1;
        AA(j,:)=AA(j,)-AA(j,s)*aux2;
    end
    aux(B(j))=bb(j);
end
x_B=[x_B; aux];
N=setdiff(1:n,B);% Retirar tudo e deixar só as 2 linhas seguinte
iv=find((CC(2,)<-epsilon)&(CC(1,)>=0));
cont=cont+1;
end
CX=-obj;
lambda=[lambda; 0];

else
lambda=0;
CX=0;
x_B=0;
disp('Conjunto admissível vazio')
end

```

Resultados

% programa principal, vai buscar os dados à função dados e resolve a função
% simplexbi e a função somaponderada. Resolve a função para um n específico
e faz o gráfico de X

```
clear
n=6;
addpath('..//MetodoSomaPonderada')
[Ad,bd,Ai,bi,C]=dados(5,n);
hold on
for i=1:n
fplot(@(x) (bd(i)-x*Ad(i,1))/Ad(i,2),[0,6])
end
x=bd(n+1)/Ad(n+1,1)*ones(1,100);
y=0.05:0.05:5;
plot(x,y);
hold off
tic
[ A,b,Cs ] = igualrest( Ad,bd,Ai,bi,C);

[lambda, CX,x_B,niteracoes]=simplexbi(A,b,Cs);
%x_B
x_B(:,1:2)
temposimplex=toc% tempo CPU
nsolucoesSimplex=size(x_B,1)% número de soluções
niteracoes
tic
[s,cont]=somaponderada(C,Ad,bd, Ai,bi);
s
temposomaponderada=toc
nsolucoesSomaponderada=size(s,1)
cont
```

% programa que produz os gráficos que compara o número de iterações e tempo
% de compilação

```
clear
```

```
k=20;
```

```

n=40;
temposomaponderada=zeros(n-1,1);
temposimplex=zeros(n-1,1);
nitS=zeros(n-1,1);
nitP=zeros(n-1,1);
for i=1:n-1
    [Ad,bd,Ai,bi,C]=dados(5,i+1);
    for j=1:k
        i+1;
        tic
        [ A,b,Cs ] = igualrest( Ad,bd,Ai,bi,C);

        [lambda, CX,x_B,niteracoes]=simplexbi(A,b,Cs);
        temposimplex(i)=toc+temposimplex(i);% tempo CPU
        nitS(i)=niteracoes+nitS(i);
        %nsolucoesSimplex=size(x_B,1)% número de soluções

        tic
        [s,cont]=somaponderada(C,Ad,bd, Ai,bi);
        temposomaponderada(i)=toc+ temposomaponderada(i);
        nitP(i)=cont+nitP(i);
        %nsolucoesSomaponderada=size(s,1)

    end
    nitS(i)=nitS(i)/k;
    temposimplex(i)=temposimplex(i)/k;

    nitP(i)=nitP(i)/k;
    temposomaponderada(i)=temposomaponderada(i)/k;
end

x=2:n;

% plot(x,temposimplex,x,temposomaponderada)
% xlabel('n') % x-axis label
% ylabel('Tempo') % y-axis label
% legend('Simplex','Soma Ponderada')
% title('Tempo')
plot(x,nitS,x,nitP);
legend(' Simplex', ' Soma Ponderada')
xlabel('n') % x-axis label
ylabel('Número de iterações') % y-axis label

```

```
%plot(x,nitS)  
%plot(x,nitP)
```