



# Rank-two programs involving linear fractional functions

Riccardo Cambini<sup>1</sup> · Giovanna D'Inverno<sup>1</sup>

Received: 12 June 2023 / Accepted: 9 March 2024  
© The Author(s) 2024

## Abstract

The aim of this paper is to deepen the study of solution methods for rank-two nonconvex problems with polyhedral feasible region, expressed by means of equality, inequality and box constraints, and objective function in the form of  $\phi\left(c^T x + c_0, \frac{d^T x + d_0}{b^T x + b_0}\right)$  or  $\bar{\phi}\left(\frac{\bar{c}^T y + \bar{c}_0}{a^T y + a_0}, \frac{d^T y + d_0}{b^T y + b_0}\right)$ . These problems arise in bicriteria programs, quantitative management science, data envelopment analysis, efficiency analysis and performance measurement. Theoretical results are proved and applied to propose a solution algorithm. Computational results are provided, comparing various splitting criteria.

**Keywords** Nonlinear programming · Low-rank structures · Linear fractional programs · Optimal level solutions · Global optimization

**Mathematics Subject Classification** 90C30 · 90C26 · 90C32

**JEL codes** C02 · C61 · C63

## 1 Introduction

Rank-two problems are low-rank problems where a linear transformation of the variables provides an objective function which depends on two variables. In other words, the objective function of rank-two problems is something like  $\phi(g_1(x), g_2(x))$ , which can be rewritten in a two-variable form  $\phi(\theta, \xi)$ , by assuming  $\theta = g_1(x)$  and  $\xi = g_2(x)$ . In this paper we will consider rank-two problems, where  $g_1(x)$  and  $g_2(x)$  are linear fractional functions. This class of problems is very used in applied models. In light of

---

✉ Riccardo Cambini  
riccardo.cambini@unipi.it

Giovanna D'Inverno  
giovanna.dinverno@unipi.it

<sup>1</sup> Department of Economics and Management, University of Pisa, Via Cosimo Ridolfi 10, 56124 Pisa, Italy

this, it is worth emphasizing chapter 3.5 in Bajalinov (2003), pp.62–72, and the books (Barros 1998; Cooper et al. 2004; Mjelde 1983) describing applications of linear fractional programs to transportation problems, product planning, location problems, quantitative management science and data envelopment analysis (see also (Bomze et al. 1997; Cambini and Martein 2009; Cambini and Sodini 2002, 2003, 2008, 2010, 2014; Frenk and Schaible 2005; Horst and Pardalos 1995; Horst and Tuy 1996; Horst et al. 2001; Konno et al. 1997, 1991; Konno and Kuno 1995; Kuno 1996; Ryoo and Sahinidis 2003; Schaible 1995; Tuy 1998)). From a theoretical and algorithmic point of view, particular rank-two problems involving linear fractional functions have been recently studied in Shen and Lu (2018), Shen et al. (2019). The aim of this paper is to study rank-two problems involving linear fractional functions from both a theoretical and an algorithmic point of view. Specifically speaking, the theoretical results stated in Cambini and Sodini (2010), Cambini (2020), Cambini and Venturi (2021) will be generalized, a solution algorithm based on a partitioning approach will be proposed, and a detailed computational experience will be provided, with new splitting criteria proposed and compared with the classical ones. In Sect. 2, a pair of rank-two problems is defined: a general one, denoted with  $\bar{P}$ , and a more particular one, named  $P$ . It will be shown that  $\bar{P}$  can be equivalently expressed in the form of  $P$ , so that just  $P$  needs to be studied from a theoretical and a computational point of view. In Sect. 3, various theoretical results are stated and compared with the ones in Cambini and Sodini (2010), Cambini (2020), Cambini and Venturi (2021). These results concerning optimal level solutions, optimal value functions and underestimation functions allow to propose in Sect. 4 a solution algorithm based on a partitioning approach. The solution method is described in detail, and a deep computational experience is discussed in Sect. 5, where new splitting criteria are proposed and compared with the classical ones.

## 2 Preliminary definitions

From now on, we will denote with  $\mathbb{R}$  the set of real numbers and with  $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$  the affinely extended real number system.

### 2.1 Main problem

**Definition 1** Let us consider a general polyhedron  $X \subset \mathbb{R}^n$  defined as:

$$X = \{x \in \mathbb{R}^n : A_{in}x \leq b_{in}, A_{eq}x = b_{eq}, l \leq x \leq u\}$$

where  $A_{in} \in \mathbb{R}^{m \times n}$ ,  $b_{in} \in \mathbb{R}^m$ ,  $A_{eq} \in \mathbb{R}^{p \times n}$ ,  $b_{eq} \in \mathbb{R}^p$ ,  $l, u \in \bar{\mathbb{R}}^n$ . Given vectors  $b, c, d \in \mathbb{R}^n$  and scalar values  $b_0, c_0, d_0 \in \mathbb{R}$ , the following class of nonconvex problems can be defined as:

$$P : \inf_{x \in X} \phi \left( c^T x + c_0, \frac{d^T x + d_0}{b^T x + b_0} \right)$$

where  $b^T x + b_0 > 0$  for all  $x \in X$ . The scalar function  $\phi(\theta, \xi)$  is defined on the following set:

$$\Omega = \left\{ (\theta, \xi) \in \mathbb{R}^2 : \theta = c^T x + c_0, \xi = \frac{d^T x + d_0}{b^T x + b_0}, x \in X \right\}$$

Notice that  $\Omega \subseteq \{\theta \in \mathbb{R} : \theta = c^T x + c_0, x \in X\} \times \left\{ \xi \in \mathbb{R} : \xi = \frac{d^T x + d_0}{b^T x + b_0}, x \in X \right\}$ . Moreover,  $\phi(\theta, \xi)$  is assumed to be continuous for all the values in the set  $\Omega$  and strictly increasing with respect to its first variable  $\theta$ .

Notice that, if  $b = 0$ , problem  $P$  reduces to the ones already studied in Cambini and Sodini (2010), Cambini (2020), Cambini and Venturi (2021) and involving linear functions  $c^T x + c_0$  and  $d^T x + d_0$  only. Notice also that these classes of problems cover several multiplicative, fractional and d.c. problems which are very used in applications (see for all Bajalinov 2003; Barros 1998; Cooper et al. 2004; Frenk and Schaible 2005; Konno et al. 1991; Kuno 1996; Mjelde 1983; Schaible 1995).

### 2.2 A more general problem

The aim of this section is to show that a class of problems involving two linear fractional functions (hence problems more general than the ones described in  $P$ ) can be converted in the form of  $P$  and hence be solved.

**Definition 2** Let us consider a general polyhedron  $Y \subset \mathbb{R}^n$  defined as:

$$Y = \{y \in \mathbb{R}^n : A_{in} y \leq b_{in}, A_{eq} y = b_{eq}, l \leq y \leq u\}$$

where  $A_{in} \in \mathbb{R}^{m \times n}$ ,  $b_{in} \in \mathbb{R}^m$ ,  $A_{eq} \in \mathbb{R}^{p \times n}$ ,  $b_{eq} \in \mathbb{R}^p$ ,  $l, u \in \overline{\mathbb{R}}^n$ . Given vectors  $a, b, \bar{c}, d \in \mathbb{R}^n$  and values  $a_0, b_0, \bar{c}_0, d_0 \in \mathbb{R}$ , the following class of nonconvex problems can be defined as:

$$\bar{P} : \inf_{y \in Y} \bar{\phi} \left( \frac{\bar{c}^T y + \bar{c}_0}{a^T y + a_0}, \frac{d^T y + d_0}{b^T y + b_0} \right)$$

where  $a^T y + a_0 > 0$  and  $b^T y + b_0 > 0$  for all  $y \in Y$ . The scalar function  $\bar{\phi}(\theta, \xi)$  is defined on the following set:

$$\bar{\Omega} = \left\{ (\theta, \xi) \in \mathbb{R}^2 : \theta = \frac{c^T y + c_0}{a^T y + a_0}, \xi = \frac{d^T y + d_0}{b^T y + b_0}, y \in Y \right\}$$

Moreover, the function  $\bar{\phi}(\theta, \xi)$  is assumed to be continuous for all the values in the set  $\bar{\Omega}$  and strictly monotone with respect to its first variable  $\theta$ .

Let us point out that particular cases of problem  $\bar{P}$  have been recently studied in Shen and Lu (2018), Shen et al. (2019) (where  $\bar{\phi}(\theta, \xi) = \theta + \xi$  or  $\bar{\phi}(\theta, \xi) = \theta \cdot \xi$ ).

First, notice that the objective function of problem  $\bar{P}$  can be equivalently rewritten as a strictly increasing function with respect to its first variable. In light of this, let us introduce the following notations:

- if  $\bar{\phi}(\theta, \xi)$  is strictly decreasing with respect to its first variable  $\theta$ , then let  $c = -\bar{c}$ ,  $c_0 = -\bar{c}_0$  and  $\phi(\theta, \xi) = \bar{\phi}(-\theta, \xi)$ ;
- if  $\bar{\phi}(\theta, \xi)$  is strictly increasing with respect to its first variable  $\theta$ , then let  $c = \bar{c}$ ,  $c_0 = \bar{c}_0$  and  $\phi(\theta, \xi) = \bar{\phi}(\theta, \xi)$ .

It results

$$\bar{\phi} \left( \frac{\bar{c}^T y + \bar{c}_0}{a^T y + a_0}, \frac{d^T y + d_0}{b^T y + b_0} \right) = \phi \left( \frac{c^T y + c_0}{a^T y + a_0}, \frac{d^T y + d_0}{b^T y + b_0} \right)$$

where  $\phi(\theta, \xi)$  is strictly increasing with respect to variable  $\theta$ .

By means of a Charnes-Cooper transformation (Charnes and Cooper 1962), it is then possible to convert problem  $\bar{P}$  in the form of  $P$ . In light of this, let us define the following scalar variable  $x_0$  and vector variable  $x_n$ :

$$x_0 = \frac{1}{a^T y + a_0} \in \mathbb{R} \text{ and } x_n = x_0 \cdot y \in \mathbb{R}^n$$

Being  $a^T y + a_0 > 0 \forall y \in Y$ , it results  $x_0 > 0$  so that:

$$\begin{aligned} \frac{c^T y + c_0}{a^T y + a_0} &= c^T x_n + c_0 x_0 \\ \frac{d^T y + d_0}{b^T y + b_0} &= \frac{d^T x_n + d_0 x_0}{b^T x_n + b_0 x_0} \end{aligned}$$

while the constraints defining  $Y$  result to be:

$$A_{in} x_n - b_{in} x_0 \leq 0 \quad A_{eq} x_n - b_{eq} x_0 = 0 \tag{1}$$

$$x_n - u x_0 \leq 0 \quad -x_n + l x_0 \leq 0 \tag{2}$$

$$a^T x_n + a_0 x_0 = 1 \quad x_0 \geq 0 \tag{3}$$

Notice that the constraint  $x_0 > 0$  has been equivalently written as  $x_0 \geq 0$  in (3), with the aim to guarantee the closure of the feasible region. In light of this, observe that the value  $x_0 = 0$  is not feasible (if  $x_0 = 0$  then (2) implies  $x_n = 0$  and hence in (3) it yields  $1 = a^T x_n + a_0 x_0 = 0$ ).

As a consequence, problem  $\bar{P}$  can be rewritten in the form of  $P$  as:

$$\bar{P} : \inf_{(x_n, x_0) \in X} \phi \left( c^T x_n + c_0 x_0, \frac{d^T x_n + d_0 x_0}{b^T x_n + b_0 x_0} \right)$$

where the polyhedron  $X \subset \mathbb{R}^{n+1}$  is given by the vectors  $(x_n, x_0) \in \mathbb{R}^n \times \mathbb{R}$  verifying the conditions (1), (2) and (3). Once the optimal solution of  $(x_n^*, x_0^*) \in X$  is found, then the corresponding optimal solution of  $\bar{P}$  is given by  $y^* = \frac{1}{x_0^*} x_n^*$ .

### 2.3 Generalized convexity properties

It is worth recalling (Avriel et al. 2010; Cambini and Martein 2009; Frenk and Schaible 2005; Schaible 1995) that linear fractional functions like  $\frac{a^T x + d_0}{b^T x + b_0}$  are both pseudoconcave and pseudoconvex.

**Definition 3** A differentiable function  $g : X \rightarrow \mathbb{R}$  with  $X \subseteq \mathbb{R}^n$  open convex set is said to be *pseudoconvex* on  $X$  if the following logical implication holds for all  $x_1, x_2 \in X$ :

$$g(x_1) > g(x_2) \Rightarrow \nabla g(x_1)^T (x_2 - x_1) < 0$$

Function  $g$  is also said to be *pseudoconcave* on  $X$  if  $-g$  is pseudoconvex on  $X$ . Moreover,  $g$  is said to be *pseudolinear* on  $X$  if it is both pseudoconvex and pseudoconcave on  $X$ .

The following useful properties hold (see for all Cambini and Martein 2009 subsection 3.3.2).

**Theorem 1** Let  $g : X \rightarrow \mathbb{R}$ , with  $X \subseteq \mathbb{R}^n$  open convex set, be a differentiable function. If  $g$  is pseudolinear on  $X$ , then either  $\nabla g(x) \neq 0$  for all  $x \in X$  or  $g$  is a constant function.

**Theorem 2** Let  $g : X \rightarrow \mathbb{R}$ , with  $X \subseteq \mathbb{R}^n$  open convex set, be a differentiable function. Then,  $g$  is pseudolinear on  $X$  if and only if the following double implication holds for all  $x_1, x_2 \in X$ :

$$g(x_1) = g(x_2) \Leftrightarrow \nabla g(x_1)^T (x_2 - x_1) = 0$$

Theorem 1 shows that differentiable pseudolinear functions are either constant or with no critical points, while Theorem 2 allows to verify that linear fractional functions are pseudolinear.

Finally, let us recall the following further results involving the concept of semistrict quasilinearity (see for all Cambini and Martein 2009, subsection 3.3.1 and subsection 3.3.2).

**Theorem 3** Let  $g : X \rightarrow \mathbb{R}$ , with  $X \subseteq \mathbb{R}^n$  open convex set, be a differentiable function. If  $g$  is pseudolinear on  $X$ , then  $g$  is also semistrictly quasilinear on  $X$ .

**Theorem 4** Let  $g : X \rightarrow \mathbb{R}$ , with  $X \subseteq \mathbb{R}^n$  convex set, be a function. Then,  $g$  is semistrictly quasilinear on  $X$  if and only if the following condition holds:

- any restriction of  $g$  on a line segment is an increasing function or a decreasing function or a constant function.

Theorems 3 and 4 show that linear fractional functions restricted on feasible segments are either constant or strictly monotone. Notice also that this property guarantees that non-constant linear fractional functions over a compact feasible region attain the minimum and the maximum at the boundaries of the feasible region.

### 3 Theoretical results

The aim of this section is to state various theoretical results concerning problem  $P$  which will allow to propose a solution algorithm. In light of this, the results stated in Cambini and Sodini (2010), Cambini (2020), Cambini and Venturi (2021) will be generalized or compared.

#### 3.1 Image of feasible intervals over $\Omega$

**Theorem 5** Consider problem  $P$  and let  $x_1, x_2 \in X$ , with  $x_1 \neq x_2$ , be extrema of the feasible segment:

$$[x_1, x_2] = \{x \in X : x = x_1 + \lambda(x_2 - x_1), \lambda \in [0, 1]\}$$

For the sake of convenience, define also the following auxiliary values:

$$\xi_1 = \frac{d^T x_1 + d_0}{b^T x_1 + b_0}, \quad \xi_2 = \frac{d^T x_2 + d_0}{b^T x_2 + b_0}$$

$\beta_1 = b^T x_1 + b_0 > 0$ ,  $\beta_2 = b^T x_2 + b_0 > 0$ ,  $\theta_1 = c^T x_1 + c_0$  and  $\theta_2 = c^T x_2 + c_0$ . We can also assume, without loss of generality, that  $\xi_1 \leq \xi_2$ .

i) If  $\xi_1 = \xi_2$ , then, denoting  $x(\lambda) = x_1 + \lambda(x_2 - x_1)$  and  $\theta(\lambda) = c^T x(\lambda) + c_0$ , it results:

$$\frac{d^T x(\lambda) + d_0}{b^T x(\lambda) + b_0} = \xi_1, \quad \theta(\lambda) = \theta_1 + \lambda(\theta_2 - \theta_1), \quad \forall \lambda \in [0, 1]$$

ii) If  $\xi_1 < \xi_2$ , then for all  $\xi \in [\xi_1, \xi_2]$  there exists one and only one  $x(\xi) \in [x_1, x_2]$  such that:

$$\xi = \frac{d^T x(\xi) + d_0}{b^T x(\xi) + b_0}$$

Moreover, denoting  $\theta(\xi) = c^T x(\xi) + c_0$ , for all  $\xi \in [\xi_1, \xi_2]$  it results:

$$x(\xi) = x_1 + \frac{(\xi - \xi_1)\beta_1}{(\xi - \xi_1)\beta_1 + (\xi_2 - \xi)\beta_2}(x_2 - x_1)$$

$$\theta(\xi) = \theta_1 + \frac{(\xi - \xi_1)\beta_1}{(\xi - \xi_1)\beta_1 + (\xi_2 - \xi)\beta_2}(\theta_2 - \theta_1)$$

**Proof** i) Just notice that:

$$\frac{d^T x(\lambda) + d_0}{b^T x(\lambda) + b_0} = \frac{d^T (x_1 + \lambda(x_2 - x_1)) + d_0}{b^T (x_1 + \lambda(x_2 - x_1)) + b_0}$$

$$\begin{aligned} &= \frac{(d^T x_1 + d_0) + \lambda ((d^T x_2 + d_0) - (d^T x_1 + d_0))}{(b^T x_1 + b_0) + \lambda ((b^T x_2 + b_0) - (b^T x_1 + b_0))} \\ &= \frac{\xi_1 \beta_1 + \lambda(\xi_2 \beta_2 - \xi_1 \beta_1)}{\beta_1 + \lambda(\beta_2 - \beta_1)} \end{aligned}$$

so that, if  $\xi_1 = \xi_2$  then  $\frac{d^T x(\lambda)+d_0}{b^T x(\lambda)+b_0} = \xi_1$ . Finally, it results:

$$c^T x(\lambda) + c_0 = (c^T x_1 + c_0) + \lambda ((c^T x_2 + c_0) - (c^T x_1 + c_0)) = \theta_1 + \lambda(\theta_2 - \theta_1)$$

ii) Since linear fractional functions are both pseudoconcave and pseudoconvex, then restricted to any segment they results to be either constant or strictly monotone. As a consequence, condition  $\xi_1 < \xi_2$  implies that function  $\frac{d^T x+d_0}{b^T x+b_0}$  is continuous and strictly monotone over the segment  $[x_1, x_2] \subset X$  so that:

$$[\xi_1, \xi_2] = \left\{ \xi : \xi = \frac{d^T x + d_0}{b^T x + b_0}, x \in [x_1, x_2] \right\}$$

This yields that for all  $\xi \in [\xi_1, \xi_2]$  there is one and only one  $x(\xi) \in [x_1, x_2]$  such that  $\xi = \frac{d^T x(\xi)+d_0}{b^T x(\xi)+b_0}$ . In light of this, it results:

$$\xi = \frac{d^T (x_1 + \lambda(x_2 - x_1)) + d_0}{b^T (x_1 + \lambda(x_2 - x_1)) + b_0} = \frac{\xi_1 \beta_1 + \lambda(\xi_2 \beta_2 - \xi_1 \beta_1)}{\beta_1 + \lambda(\beta_2 - \beta_1)}$$

which yields:

$$\xi \beta_1 + \lambda \xi (\beta_2 - \beta_1) = \xi_1 \beta_1 + \lambda (\xi_2 \beta_2 - \xi_1 \beta_1)$$

and hence:

$$\lambda = \frac{(\xi - \xi_1) \beta_1}{(\xi - \xi_1) \beta_1 + (\xi_2 - \xi) \beta_2}$$

As a consequence, it results:

$$x(\xi) = x_1 + \frac{(\xi - \xi_1) \beta_1}{(\xi - \xi_1) \beta_1 + (\xi_2 - \xi) \beta_2} (x_2 - x_1)$$

and

$$\theta(\xi) = c^T x(\xi) + c_0 = \theta_1 + \frac{(\xi - \xi_1) \beta_1}{(\xi - \xi_1) \beta_1 + (\xi_2 - \xi) \beta_2} (\theta_2 - \theta_1)$$

□

**Remark 1** Notice that when  $\xi_1 < \xi_2$ :

$$\theta'(\xi) = \frac{\beta_1\beta_2(\theta_2 - \theta_1)(\xi_2 - \xi_1)}{[(\xi - \xi_1)\beta_1 + (\xi_2 - \xi)\beta_2]^2}$$

$$\theta''(\xi) = \frac{2\beta_1\beta_2(\beta_2 - \beta_1)(\theta_2 - \theta_1)(\xi_2 - \xi_1)}{[(\xi - \xi_1)\beta_1 + (\xi_2 - \xi)\beta_2]^3}$$

Being  $\beta_1 > 0, \beta_2 > 0, \xi_1 < \xi_2$ , it results  $(\xi - \xi_1)\beta_1 + (\xi_2 - \xi)\beta_2 > 0$  for all  $\xi \in [\xi_1, \xi_2]$  so that function  $\theta(\xi)$  is:

- constant if  $\theta_1 = \theta_2$
- strictly increasing if  $\theta_1 < \theta_2$
- strictly decreasing if  $\theta_1 > \theta_2$
- linear if either  $\beta_1 = \beta_2$  or  $\theta_1 = \theta_2$
- strictly convex if  $(\beta_2 - \beta_1)(\theta_2 - \theta_1) > 0$
- strictly concave if  $(\beta_2 - \beta_1)(\theta_2 - \theta_1) < 0$

**Remark 2** It is useful to determine the “distance” between  $\theta(\xi)$  and its linear counterpart  $\theta_1 + \frac{\xi - \xi_1}{\xi_2 - \xi_1}(\theta_2 - \theta_1)$ :

$$\Delta(\xi) = \theta(\xi) - \left( \theta_1 + \frac{\xi - \xi_1}{\xi_2 - \xi_1}(\theta_2 - \theta_1) \right)$$

By means of simple calculations we get:

$$\Delta(\xi) = \frac{(\theta_2 - \theta_1)(\beta_2 - \beta_1)}{(\xi_2 - \xi_1)} \cdot \frac{-(\xi - \xi_1)(\xi_2 - \xi)}{(\xi - \xi_1)\beta_1 + (\xi_2 - \xi)\beta_2}$$

Clearly, if either  $\beta_1 = \beta_2$  or  $\theta_1 = \theta_2$ , then  $\Delta(\xi)$  is constantly null. If  $(\beta_2 - \beta_1)(\theta_2 - \theta_1) \neq 0$ , it is interesting to determine the “maximum distance”  $\Delta(\xi)$ . In light of this, notice that:

$$\Delta'(\xi) = \frac{(\theta_2 - \theta_1)(\beta_2 - \beta_1)}{(\xi_2 - \xi_1)} \cdot \frac{(\xi - \xi_1)^2\beta_1 - (\xi_2 - \xi)^2\beta_2}{[(\xi - \xi_1)\beta_1 + (\xi_2 - \xi)\beta_2]^2}$$

which vanishes when  $(\xi - \xi_1)^2\beta_1 = (\xi_2 - \xi)^2\beta_2$  that is  $(\xi - \xi_1)\sqrt{\beta_1} = (\xi_2 - \xi)\sqrt{\beta_2}$  (all factors are positive). The “maximum distance” is then reached at:

$$\xi^* = \frac{\xi_1\sqrt{\beta_1} + \xi_2\sqrt{\beta_2}}{\sqrt{\beta_1} + \sqrt{\beta_2}} = \xi_1 + (\xi_2 - \xi_1)\frac{\sqrt{\beta_2}}{\sqrt{\beta_1} + \sqrt{\beta_2}}$$

so that:

$$\Delta(\xi^*) = \frac{(\theta_2 - \theta_1)(\beta_2 - \beta_1)}{(\xi_2 - \xi_1)} \cdot \frac{-(\xi_2 - \xi_1)}{(\sqrt{\beta_1} + \sqrt{\beta_2})^2} = \frac{-(\theta_2 - \theta_1)(\beta_2 - \beta_1)}{(\sqrt{\beta_1} + \sqrt{\beta_2})^2}$$



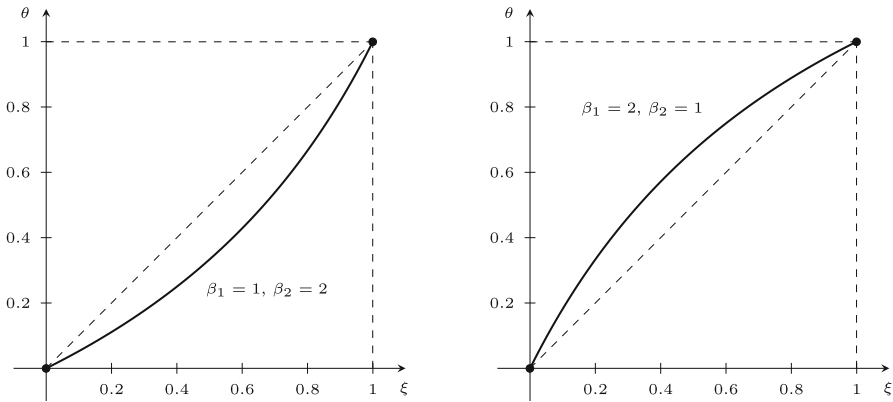


Fig. 1 Plots of  $\theta(\xi)$  with  $\theta_1 = \xi_1 = 0$  and  $\theta_2 = \xi_2 = 1$

As expected, if  $\theta(\xi)$  is strictly concave, then  $(\theta_2 - \theta_1)(\beta_2 - \beta_1) < 0$  and the “distance” is positive (right-hand side of Fig. 1), while if  $\theta(\xi)$  is strictly convex, then  $(\theta_2 - \theta_1)(\beta_2 - \beta_1) > 0$  and the “distance” is negative (left-hand side of Fig. 1).

### 3.2 Feasible levels

The *feasible levels set* is defined as:

$$\Lambda = \left\{ \xi \in \mathbb{R} : \xi = \frac{d^T x + d_0}{b^T x + b_0}, x \in X \right\} \tag{4}$$

Scalar values in  $\Lambda$  are said *feasible levels*. In the following, specific values of  $\Lambda$  will be denoted with  $\xi'$ ,  $\xi_1$ ,  $\xi_2$ , and so on.

**Theorem 6** *The following properties hold:*

- i)  $\Lambda$  is convex;
- ii) if  $X$  is bounded, then  $\Lambda$  is compact.

**Proof** i) The set  $X \subset \mathbb{R}^n$  is a polyhedron, hence it is convex. All convex sets are connected, and hence  $X$  is connected too. Given a continuous function, the image of a connected set is connected too. As a consequence, the continuity of  $\frac{d^T x + d_0}{b^T x + b_0}$  implies that  $\Lambda$  is connected. Subsets of  $\mathbb{R}$  are convex if and only if they are connected. As a consequence, the set  $\Lambda \subseteq \mathbb{R}$  is convex.

ii) The set  $X \subset \mathbb{R}^n$  is a closed polyhedron and is bounded, hence it is compact. Given a continuous function, the image of a compact set is compact too. As a consequence, the continuity of  $\frac{d^T x + d_0}{b^T x + b_0}$  implies that  $\Lambda$  is compact. □

**Remark 3** Notice that the closedness of polyhedron  $X$  does not guarantee the closedness of  $\Lambda$ . For example, consider the closed unbounded polyhedral set  $X = \{x \in \mathbb{R} :$

$x \geq 0$ ) and values  $d = b = 1, d_0 = 0, b_0 = 1$ . Then, it is:

$$\Lambda = \left\{ \xi \in \mathbb{R} : \xi = \frac{x}{x + 1}, x \geq 0 \right\} = [0, 1)$$

which is a bounded interval neither open nor closed.

For the sake of convenience, the following values can be determined:

$$\xi_{min} = \inf_{x \in X} \frac{d^T x + d_0}{b^T x + b_0}, \quad \xi_{max} = \sup_{x \in X} \frac{d^T x + d_0}{b^T x + b_0}$$

Notice that  $\xi_{min}$  and  $\xi_{max}$  can be computed by means of a Charnes-Cooper transformation (Charnes and Cooper 1962) or by means of the method proposed in Cambini and Sodini (2010), Cambini (2020), Cambini and Venturi (2021).

### 3.3 Optimal level solutions

Given a specific feasible level  $\xi' \in \Lambda$ , the corresponding subset of  $X$  can be defined as:

$$X_{\xi'} = \left\{ x \in X : \xi' = \frac{d^T x + d_0}{b^T x + b_0} \right\} = \left\{ x \in X : d^T x + d_0 - \xi'(b^T x + b_0) = 0 \right\}$$

Notice that the set  $X_{\xi'}$  is a polyhedron too. Moreover, the following parametric subproblems can be introduced for all  $\xi \in \Lambda$ :

$$P_{\xi} : \inf_{x \in X_{\xi}} \phi(c^T x + c_0, \xi) \quad ; \quad \hat{P}_{\xi} : \inf_{x \in X_{\xi}} c^T x + c_0$$

Notice that being  $\phi(\theta, \xi)$  strictly increasing with respect to its first variable  $\theta$ , then subproblem  $P_{\xi}$  can be solved by means of the linear programming subproblem  $\hat{P}_{\xi}$ . Given a specific feasible level  $\xi' \in \Lambda$ , optimal solutions of  $P_{\xi'}$  and  $\hat{P}_{\xi'}$ , if they exist, are said to be *optimal level solutions*. The set  $S_{\xi'} \subset X_{\xi'}$  collects all of these optimal level solutions. It is worth recalling that the optimal level solutions have been the key tool in stating the so called “optimal level solutions method”, see for all (Cambini and Martein 2009; Cambini and Sodini 2002, 2003, 2008, 2010, 2014).

As preliminary results, let us now provide some conditions related to the existence of optimal level solutions. These conditions extend some of the results proved in Cambini and Sodini (2010), Cambini and Venturi (2021). Let us also recall that for Theorem 6, if  $X$  is bounded, then  $\Lambda$  is convex and compact.

**Theorem 7** Consider problem  $P$ . The following properties hold:

- i) if there exists  $\xi' \in \Lambda$  such that  $S_{\xi'} = \emptyset$ , then  $S_{\xi} = \emptyset \forall \xi \in \Lambda$ ;
- ii) if there exists  $\xi' \in \Lambda$  such that  $S_{\xi'} \neq \emptyset$ , then  $S_{\xi} \neq \emptyset \forall \xi \in \Lambda$ ;
- iii) if there exists  $\xi' \in \Lambda$  such that  $S_{\xi'} \neq \emptyset$  and  $\Lambda$  is compact, then problem  $P$  admits at least one optimal solution;

iv) every local minimum of  $P$  is also an optimal level solution.

**Proof** i) Since  $X$  is a closed polyhedron and  $S_{\xi'} = \emptyset$ , then  $\inf_{x \in X_{\xi'}} c^T x = -\infty$  and hence there exists a half-line  $x = x' + tu, t \geq 0$ , contained in  $X_{\xi'}$  such that for all  $t \geq 0$ :

$$\xi' = \frac{d^T x + d_0}{b^T x + b_0} = \frac{d^T x' + d_0 + td^T u}{b^T x' + b_0 + tb^T u}$$

and  $\lim_{t \rightarrow +\infty} c^T x' + tc^T u = -\infty$ . As a consequence,  $c^T u < 0$ . Moreover,  $t(\xi' b - d)^T u = (d^T x' + d_0) - \xi'(b^T x' + b_0) = 0$  for all  $t \geq 0$  which implies  $(\xi' b - d)^T u = 0$ . Given any feasible level  $\xi \in \Lambda$ , we can consider a half-line  $x = x_{\xi} + tu, t \geq 0$ , where  $x_{\xi} \in X_{\xi}$  and  $u$  is the direction of the half-line contained in  $X_{\xi}$ . Since  $(\xi' b - d)^T u = 0$ , this half-line belongs to  $X_{\xi}$  and from  $c^T u < 0$  it follows that  $\lim_{t \rightarrow +\infty} c^T (x_{\xi} + tu) = \lim_{t \rightarrow +\infty} (c^T x_{\xi} + tc^T u) = -\infty$  and hence  $S_{\xi} = \emptyset$ .

ii) Assume by contradiction that there exists  $\xi' \in \Lambda$  such that  $S_{\xi'} = \emptyset$ . Then, i) yields  $S_{\xi} = \emptyset \forall \xi \in \Lambda$ , which contradicts the assumption.

iii) Assume by contradiction that  $P$  has no optimal solutions. Then, there exists a half-line  $x = x' + tu, t \geq 0$ , such that:

$$\inf_{x \in X} \phi \left( c^T x + c_0, \frac{d^T x + d_0}{b^T x + b_0} \right) = \lim_{t \rightarrow +\infty} \phi \left( c^T (x' + tu) + c_0, \frac{d^T (x' + tu) + d_0}{b^T (x' + tu) + b_0} \right)$$

Notice that  $\lim_{t \rightarrow +\infty} \frac{d^T x' + d_0 + td^T u}{b^T x' + b_0 + tb^T u} = \frac{d^T u}{b^T u}$ . Since  $\frac{d^T x' + d_0 + td^T u}{b^T x' + b_0 + tb^T u} \in \Lambda$  and  $\Lambda$  is compact, then  $\hat{\xi} = \frac{d^T u}{b^T u} \in \Lambda$ . Moreover, since  $P$  has no optimal solutions, then  $\lim_{t \rightarrow +\infty} c^T x' + c_0 + tc^T u = -\infty$  and hence  $c^T u < 0$ . Let  $\hat{x} \in X_{\hat{\xi}}$  so that  $\hat{\xi} = \frac{d^T \hat{x} + d_0}{b^T \hat{x} + b_0}$  and consider the half-line  $x = \hat{x} + tu, t \geq 0$ , where  $u$  is the direction of the previous half-line. Taking into account this new half-line, it results:

$$\hat{\xi} = \frac{d^T \hat{x} + d_0}{b^T \hat{x} + b_0} \quad \text{and} \quad \hat{\xi} = \frac{d^T u}{b^T u} = \lim_{t \rightarrow +\infty} \frac{d^T \hat{x} + d_0 + td^T u}{b^T \hat{x} + b_0 + tb^T u}$$

By Theorem 4, the linear fractional function  $\frac{d^T x + d_0}{b^T x + b_0}$  along the half-line  $x = \hat{x} + tu, t \geq 0$ , must be constant, so that  $\hat{x} + tu \in X_{\hat{\xi}}$  for all  $t \geq 0$ . Being  $c^T u < 0$ , it results  $\lim_{t \rightarrow +\infty} c^T \hat{x} + c_0 + tc^T u = -\infty$  so that there are no optimal level solutions for  $\hat{\xi} \in \Lambda$ . For i), it follows that  $S_{\xi} = \emptyset$  for all  $\xi \in \Lambda$  and this contradicts the assumptions.

iv) Let  $x' \in X$  be a local minimum of  $P$ , let  $\xi' = \frac{d^T x' + d_0}{b^T x' + b_0}$  the corresponding feasible level, and suppose by contradiction that it is not an optimal level solution. Then, there exists  $y \in X_{\xi'}$  such that  $c^T y < c^T x'$  with  $\xi' = \frac{d^T y + d_0}{b^T y + b_0}$ . As a consequence, for all  $\lambda \in (0, 1)$  it is

$$\frac{d^T (x' + \lambda(y - x')) + d_0}{b^T (x' + \lambda(y - x')) + b_0} = \frac{d^T x' + d_0 + \lambda(d^T y + d_0 - d^T x' - d_0)}{b^T (x' + \lambda(y - x')) + b_0}$$

$$= \frac{\xi'(b^T x' + b_0) + \xi' \lambda(b^T y + b_0 - b^T x' - b_0)}{b^T x' + b_0 + \lambda(b^T y + b_0 - b^T x' - b_0)} = \xi'$$

and  $c^T(x' + \lambda(y - x')) < c^T x'$ . Since  $\phi(\cdot, \cdot)$  is strictly increasing with respect to its first variable, it holds:

$$\phi(c^T(x' + \lambda(y - x')) + c_0, \xi') < \phi(c^T x' + c_0, \xi')$$

and this contradicts the local optimality of  $x'$  with respect to problem  $P$ . □

The following results follow straightforward from the previous theorem:

- if  $P$  admits at least one local minimum, then  $S_\xi \neq \emptyset \forall \xi \in \Lambda$ .
- if  $P$  admits at least one local minimum and  $\Lambda$  is compact, then problem  $P$  admits at least one optimal solution.

Moreover, the results stated in the previous theorem can be improved by means of following value:

$$\theta_{min} = \inf_{x \in X} c^T x + c_0$$

**Theorem 8** Consider problem  $P$ . The following properties hold:

i) if  $\theta_{min} > -\infty$  then  $\exists x_{\theta_{min}} \in X$  such that:

$$x_{\theta_{min}} = \arg \min_{x \in X} c^T x + c_0 \quad \text{and} \quad x_{\theta_{min}} \in S_{\xi_{\theta_{min}}} \neq \emptyset \quad \xi_{\theta_{min}} = \frac{d^T x_{\theta_{min}} + d_0}{b^T x_{\theta_{min}} + b_0}$$

ii) if  $\theta_{min} > -\infty$  then  $S_\xi \neq \emptyset \forall \xi \in \Lambda$ ;

iii) if  $\theta_{min} > -\infty$  and the set of feasible levels  $\Lambda$  is compact, then problem  $P$  admits at least one optimal solution.

**Proof** i) If  $\theta_{min} > -\infty$  then  $\inf_{x \in X} c^T x + c_0$  is reached as a minimum, so that there exists  $x_{\theta_{min}} = \arg \min_{x \in X} c^T x + c_0$ . Defined  $\xi_{\theta_{min}} \in \Lambda$ , then  $x_{\theta_{min}}$  results to be an optimal level solution corresponding to the feasible level  $\xi_{\theta_{min}}$  since it is  $c^T x + c_0 \geq \theta_{min}$  for all  $x \in X \supset X_{\xi_{\theta_{min}}}$ .

ii) Follows from i) and from ii) of Theorem 7.

iii) Follows from i) and from iii) of Theorem 7. □

**Remark 4** Regarding ii) of Theorem 8, it is worth noticing that it is possible to have  $\theta_{min} = -\infty$  and  $S_\xi \neq \emptyset \forall \xi \in \Lambda$ . For example, consider the closed unbounded polyhedral set  $X = \{(x_1, x_2) \in \mathbb{R}^2 : x_2 \geq x_1\}$ , vectors  $c = (1, 1)^T$ ,  $d = (1, 0)^T$ ,  $b = (0, 0)^T$  and values  $c_0 = d_0 = 0$ ,  $b_0 = 1$ , so that the problem is:

$$\begin{cases} \inf_{(x_1, x_2) \in \mathbb{R}^2} \phi(x_1 + x_2, x_1) \\ x_2 \geq x_1 \end{cases}$$

It results  $\theta_{min} = \inf_{(x_1, x_2) \in X} x_1 + x_2 = -\infty$  and  $\Lambda = \mathbb{R}$ . Moreover, for all  $\xi \in \Lambda$  it is  $X_\xi = \{(x_1, x_2) \in \mathbb{R}^2 : x_2 \geq x_1, x_1 = \xi\}$  so that:

$$\arg \min_{(x_1, x_2) \in X_\xi} (x_1 + x_2) = (\xi, \xi) \in S_\xi \neq \emptyset .$$

### 3.4 Optimal value functions

For the sake of convenience, the following optimal value functions can be introduced:

$$\sigma(\xi) = \inf_{x \in X_\xi} \phi(c^T x + c_0, \xi) \quad , \quad \hat{\theta}(\xi) = \inf_{x \in X_\xi} c^T x + c_0 \quad , \quad \xi \in \Lambda$$

Notice that the fact that  $\phi(\theta, \xi)$  is strictly increasing with respect to its first variable  $\theta$  implies that  $\sigma(\xi) = \phi(\hat{\theta}(\xi), \xi)$ .

The properties stated in the previous subsection are very useful since they suggest a simple way to check the existence of optimal level solutions. In light of this, let us first define the following starting feasible level  $\xi_0$ :

$$\xi_0 = \begin{cases} 0 & \text{if } \xi_{min} \xi_{max} \leq 0 \\ \xi_{min} & \text{if } \xi_{min} > 0 \text{ (and hence } \xi_{max} > 0) \\ \xi_{max} & \text{if } \xi_{max} < 0 \text{ (and hence } \xi_{min} < 0) \end{cases} \tag{5}$$

The following two cases occur:

- if  $S_{\xi_0} = \emptyset$ , then  $S_\xi = \emptyset \forall \xi \in \Lambda$  so that:

$$\inf_{x \in X} \phi\left(c^T x + c_0, \frac{d^T x + d_0}{b^T x + b_0}\right) = \inf_{\xi \in \Lambda} \inf_{x \in X_\xi} \phi\left(c^T x + c_0, \xi\right) = \inf_{\xi \in \Lambda} \phi(-\infty, \xi)$$

- if  $S_{\xi_0} \neq \emptyset$ , then  $S_\xi \neq \emptyset \forall \xi \in \Lambda$  so that:

$$\inf_{x \in X} \phi\left(c^T x + c_0, \frac{d^T x + d_0}{b^T x + b_0}\right) = \inf_{\xi \in \Lambda} \min_{x \in X_\xi} \phi\left(c^T x + c_0, \xi\right) = \inf_{\xi \in \Lambda} \phi\left(\hat{\theta}(\xi), \xi\right)$$

It is now possible to prove the following fundamental result, which will allow to determine an underestimation function for  $\sigma(\xi)$  and hence to develop a solution algorithm based on a partitioning approach.

**Theorem 9** Assume  $S_\xi \neq \emptyset \forall \xi \in \Lambda$ . Then, the optimal value function  $\hat{\theta}(\xi), \xi \in \Lambda$ , is a semistrictly quasiconvex function. Moreover, if  $\theta_{min} > -\infty$ , then define:

$$\xi_{\theta_{min}}^L = \inf_{x \in X, c^T x + c_0 = \theta_{min}} \frac{d^T x + d_0}{b^T x + b_0} \quad , \quad \xi_{\theta_{min}}^R = \sup_{x \in X, c^T x + c_0 = \theta_{min}} \frac{d^T x + d_0}{b^T x + b_0}$$

so that it results:

- $\hat{\theta}(\xi)$  is strictly decreasing for  $\xi \in \Lambda$ ,  $\xi \leq \xi_{\theta_{min}}^L$
- $\hat{\theta}(\xi) = \theta_{min}$  for all  $\xi \in [\xi_{\theta_{min}}^L, \xi_{\theta_{min}}^R]$
- $\hat{\theta}(\xi)$  is strictly increasing for  $\xi \in \Lambda$ ,  $\xi \geq \xi_{\theta_{min}}^R$

Finally, if  $\theta_{min} = -\infty$ , then  $\hat{\theta}(\xi)$  is strictly monotone for  $\xi \in \Lambda$ .

**Proof** We have to prove that if  $\hat{\theta}(\xi_1) < \hat{\theta}(\xi_2)$ , then  $\hat{\theta}(\xi_1 + \lambda(\xi_2 - \xi_1)) < \hat{\theta}(\xi_2)$  for all  $\lambda \in [0, 1]$ . Assume by contradiction that there exists  $\bar{\lambda} \in [0, 1]$  such that  $\hat{\theta}(\bar{\xi}) \geq \hat{\theta}(\xi_2)$ ,  $\bar{\xi} = \xi_1 + \bar{\lambda}(\xi_2 - \xi_1)$ , and let  $x_1 = \arg \min_{x \in X_{\xi_1}} c^T x + c_0$  and  $x_2 = \arg \min_{x \in X_{\xi_2}} c^T x + c_0$ ,

so that  $\hat{\theta}(\xi_1) = c^T x_1 + c_0$  and  $\hat{\theta}(\xi_2) = c^T x_2 + c_0$ . Consider the feasible segment  $[x_1, x_2]$ . By Theorem 5 and Remark 1, being  $\xi_1 \neq \xi_2$ , there exists one and only one  $\bar{x} \in [x_1, x_2] \cap X_{\bar{\xi}}$ . Moreover, being  $c^T x_1 + c_0 < c^T x_2 + c_0$ , then  $c^T x + c_0$  is strictly monotone on the segment  $[x_1, x_2]$  so that:

$$c^T \bar{x} + c_0 < c^T x_2 + c_0 = \hat{\theta}(\xi_2) \leq \hat{\theta}(\bar{\xi})$$

which is a contradiction since  $\bar{x} \in X_{\bar{\xi}}$  and  $\hat{\theta}(\bar{\xi}) = \min_{x \in X_{\bar{\xi}}} c^T x + c_0$ .

Being  $\hat{\theta}(\xi)$  semistrictly quasiconvex, then every local minimum results to be global minimum too, and the set of global minima is convex. As a consequence,  $\hat{\theta}(\xi)$  can be constant over an interval  $[\xi_1, \xi_2]$  only if  $\hat{\theta}(\xi) = \theta_{min}$  for all  $\xi \in [\xi_1, \xi_2]$ , while  $\hat{\theta}(\xi)$  is strictly monotone in intervals  $[\xi_1, \xi_2]$  such that  $[\xi_{\theta_{min}}^L, \xi_{\theta_{min}}^R] \cap [\xi_1, \xi_2] = \emptyset$  or when  $\theta_{min} = -\infty$ . □

### 3.5 Underestimation functions

In order to propose a partitioning solution method, an underestimation function for  $\sigma(\xi)$  is needed.

**Definition 4** Assume  $S_{\xi} \neq \emptyset \forall \xi \in \Lambda$ . A function  $\psi(\xi)$  such that  $\psi(\xi) \leq \sigma(\xi) \forall \xi \in [\xi_1, \xi_2] \subseteq \Lambda$  is said to be an *underestimation function* for  $\sigma(\xi)$  in  $[\xi_1, \xi_2]$ . Moreover, a function  $\gamma(\xi)$  such that  $\gamma(\xi) \leq \hat{\theta}(\xi) \forall \xi \in [\xi_1, \xi_2] \subseteq \Lambda$  is said to be an *underestimation function* for  $\hat{\theta}(\xi)$  in  $[\xi_1, \xi_2]$ .

A particular underestimation function  $\psi(\xi)$  for  $\sigma(\xi)$  in  $[\xi_1, \xi_2]$  can be built by considering the composition  $\phi(\gamma(\xi), \xi)$ , where  $\gamma(\xi)$  is an underestimation function for  $\hat{\theta}(\xi)$  in  $[\xi_1, \xi_2]$ .

**Theorem 10** Let  $[\xi_1, \xi_2] \subseteq \Lambda$  and let  $\gamma(\xi)$  be an underestimation function for  $\hat{\theta}(\xi)$  in  $[\xi_1, \xi_2]$ . Then,  $\psi(\xi) = \phi(\gamma(\xi), \xi)$  is an underestimation function for  $\sigma(\xi)$  in  $[\xi_1, \xi_2]$ .

**Proof** Being function  $\phi(\cdot, \cdot)$  strictly increasing with respect to its first variable, it results:

$$\psi(\xi) = \phi(\gamma(\xi), \xi) \leq \phi(\hat{\theta}(\xi), \xi) = \sigma(\xi) \quad \forall \xi \in [\xi_1, \xi_2] \tag{6}$$

and hence  $\psi(\xi)$  is an underestimation function for  $\sigma(\xi)$  in  $[\xi_1, \xi_2]$ . □

The following results will suggest some stopping criteria for the solution algorithm, thus allowing to improve its performance.

**Theorem 11** *Let  $[\xi_1, \xi_2] \subseteq \Lambda$ , let  $\psi(\xi)$  be an underestimation function for  $\sigma(\xi)$  in  $[\xi_1, \xi_2]$  and assume that:*

$$\xi_m = \arg \min_{\xi \in [\xi_1, \xi_2]} \psi(\xi) \quad \text{and} \quad \psi(\xi_m) = \sigma(\xi_m)$$

*Then,  $\xi_m = \arg \min_{\xi \in [\xi_1, \xi_2]} \sigma(\xi)$ .*

**Proof** Just notice that:

$$\sigma(\xi_m) = \psi(\xi_m) \leq \psi(\xi) \leq \sigma(\xi) \quad \forall \xi \in [\xi_1, \xi_2]$$

so that  $\xi_m = \arg \min_{\xi \in [\xi_1, \xi_2]} \sigma(\xi)$ . □

The following corollary comes straightforward from the previous theorem.

**Corollary 1** *Let  $[\xi_1, \xi_2] \subseteq \Lambda$ , let  $\gamma(\xi)$  be an underestimation function for  $\hat{\theta}(\xi)$  in  $[\xi_1, \xi_2]$ , let  $\psi(\xi) = \phi(\gamma(\xi), \xi)$  and assume that:*

$$\xi_m = \arg \min_{\xi \in [\xi_1, \xi_2]} \psi(\xi) \quad \text{and} \quad \gamma(\xi_m) = \hat{\theta}(\xi_m)$$

*Then,  $\xi_m = \arg \min_{\xi \in [\xi_1, \xi_2]} \sigma(\xi)$ .*

**Proof** Just notice that  $\psi(\xi_m) = \phi(\gamma(\xi_m), \xi) = \phi(\hat{\theta}(\xi_m), \xi) = \sigma(\xi_m)$ . □

The results stated in Subsection 3.4 allow to determine the following properties.

**Theorem 12** *Let  $[\xi_1, \xi_2] \subseteq \Lambda$  and assume that one of the following conditions holds:*

- $\theta_{min} = -\infty$ ;
- $\theta_{min} > -\infty$  with either  $\xi_{\theta_{min}}^R \leq \xi_1$  or  $\xi_2 \leq \xi_{\theta_{min}}^L$ .

*Then, the following functions  $\gamma_1(\xi)$  and  $\psi_1(\xi)$  are underestimation functions in  $[\xi_1, \xi_2]$  for  $\hat{\theta}(\xi)$  and  $\sigma(\xi)$ , respectively:*

$$\gamma_1(\xi) = \min\{\hat{\theta}(\xi_1), \hat{\theta}(\xi_2)\} \quad \text{and} \quad \psi_1(\xi) = \phi(\gamma_1(\xi), \xi)$$

*Moreover, the following results hold:*

- i) if  $\xi_1 = \arg \min_{\xi \in [\xi_1, \xi_2]} \psi_1(\xi)$  and  $\hat{\theta}(\xi_1) \leq \hat{\theta}(\xi_2)$ , then  $\xi_1 = \arg \min_{\xi \in [\xi_1, \xi_2]} \sigma(\xi)$ ;*
- ii) if  $\xi_2 = \arg \min_{\xi \in [\xi_1, \xi_2]} \psi_1(\xi)$  and  $\hat{\theta}(\xi_2) \leq \hat{\theta}(\xi_1)$ , then  $\xi_2 = \arg \min_{\xi \in [\xi_1, \xi_2]} \sigma(\xi)$ .*

**Proof** By Theorem 9 it follows that  $\hat{\theta}(\xi)$  is strictly monotone in  $[\xi_1, \xi_2]$  which yields:

$$\gamma_1(\xi) = \min\{\hat{\theta}(\xi_1), \hat{\theta}(\xi_2)\} \leq \hat{\theta}(\xi) \quad \forall \xi \in [\xi_1, \xi_2]$$

so that  $\gamma_1(\xi)$  is an underestimation functions for  $\hat{\theta}(\xi)$  in  $[\xi_1, \xi_2]$ . By Theorem 10 this yields that  $\psi_1(\xi)$  is an underestimation functions for  $\sigma(\xi)$  in  $[\xi_1, \xi_2]$ . Finally, to prove *i)* assume  $\xi_1 = \arg \min_{\xi \in [\xi_1, \xi_2]} \psi_1(\xi)$  and  $\hat{\theta}(\xi_1) \leq \hat{\theta}(\xi_2)$ . Then,  $\gamma_1(\xi_1) = \min\{\hat{\theta}(\xi_1), \hat{\theta}(\xi_2)\} = \hat{\theta}(\xi_1)$  so that the result follows from Corollary 1. The proof of *ii)* is analogous. □

## 4 Partitioning solution method for P

The results stated in Sect. 3 allow us to state an iterative partitioning solution method for problem  $P$ . Specifically, in each iteration of the proposed algorithm:

- the best optimal level solution, denoted as  $Sol$ , found so far is maintained, as well as its value  $UB$ ;
- the partition of the feasible region corresponding to an interval  $[\xi_1, \xi_2] \subset \Lambda$  of the feasible levels set is analysed and eventually splitted;

The strategy of such a solution methods is:

- to store the partitions and their lower bound in a priority queue;
- to analyse the partitions stored in the queue following a priority order, that is to say that the partition with the smallest lower bound is visited first;
- to postpone as much as possible the analysis of the partitions having larger lower bounds;
- to avoid the visit of the partitions whose lower bound is bigger than the incumbent optimal value (these partitions cannot improve the incumbent optimal solution).

The following auxiliary subprocedure “ $OptLevSol()$ ” will be very used in the solution algorithm. Its aim is twofold, that is, to determine the optimal level solution corresponding to the feasible level  $\xi' \in \Lambda$  and to improve (if possible) the values of  $UB$  and  $Sol$ .

**Subprocedure  $OptLevSol$**  (inputs:  $\xi'$ ; outputs:  $\hat{\theta}(\xi')$ )

Let  $x' := \arg \min_{x \in X_{\xi'}} c^T x + c_0$ ,  $\hat{\theta}(\xi') = c^T x' + c_0$ ,  $val' = \phi(\hat{\theta}(\xi'), \xi')$ ;

if  $val' < UB$  then  $UB := val'$ ,  $Sol := x'$  end if;

**end subproc.**

### 4.1 Initializations and unboundedness

As it has been described in the introduction to this section, the partitioning method maintains, along iterations, the current best optimal level solution  $Sol$  as well as its value  $UB$ . The value of  $UB$  is fundamental to improve the performance of the algorithm since it is used to discard the not useful partitions. For this very reason, the algorithm should be initialized with  $Sol$  having a small  $UB$  value. Moreover, it is worth noticing that the feasible region may be unbounded (components of  $l_y$  and  $u_y$  may assume  $-\infty$  and  $+\infty$  values) so that there is the need to manage the case of unbounded  $\xi_{min}$  and  $\xi_{max}$ . These preliminary steps are described in the following subprocedures “ $InitializeAlgorithm()$ ” and “ $UnboundImproveUB()$ ”.

**Subprocedure  $InitializeAlgorithm()$**

Compute  $\xi_{min} := \inf_{x \in X} \frac{d^T x + d_0}{b^T x + b_0}$ ,  $\xi_{max} := \sup_{x \in X} \frac{d^T x + d_0}{b^T x + b_0}$ ;

if  $\xi_{min} \xi_{max} \leq 0$  then  $\xi_0 := 0$  elseif  $\xi_{min} > 0$  then  $\xi_0 := \xi_{min}$  else  $\xi_0 := \xi_{max}$  end if;



```

if  $\inf_{x \in X_{\xi_0}} c^T x + c_0 = -\infty$  then
     $Opt^* := [], Val^* := \inf_{\xi \in [\xi_{min}, \xi_{max}]} \phi(-\infty, \xi)$ , QuitProcedure;
end if;
Let  $UB := +\infty$  and  $\hat{\theta}(\xi_0) := OptLevSol(\xi_0)$ ;

```

**end subproc.**

The aim of subprocedure “*InitializeAlgorithm()*” is to determine the set  $\Lambda$  of feasible levels and to use the results of Subsection 4.4 in order to determine whether  $S_\xi = \emptyset \forall \xi \in \Lambda$  or  $S_\xi \neq \emptyset \forall \xi \in \Lambda$ . In the former case, just  $\phi(-\infty, \xi)$  have to be analyzed over  $\Lambda$ , while in the latter case  $Sol$  and  $UB$  are initialized.

**Subprocedure UnboundImproveUB()**

```

Let  $\xi_{big} \gg 0$ ;
if  $\xi_{min} > -inf$  then  $\xi_L := \xi_{min}$  else  $\xi_L := -\xi_{big}$  end if;
if  $\xi_{max} < +inf$  then  $\xi_R := \xi_{max}$  else  $\xi_R := \xi_{big}$  end if;
Let  $\hat{\theta}(\xi_L) := OptLevSol(\xi_L)$ ;
Let  $\hat{\theta}(\xi_R) := OptLevSol(\xi_R)$ ;
if  $\xi_{min} < \xi_L$  then
    [ $opt, val$ ] := SolveUnbound( $\phi, X, \xi_{min}, \xi_L$ );
    if  $val < UB$  then  $UB := val, Sol := opt$  end if;
end if;
if  $\xi_{max} > \xi_R$  then
    [ $opt, val$ ] := SolveUnbound( $\phi, X, \xi_R, \xi_{max}$ );
    if  $val < UB$  then  $UB := val, Sol := opt$  end if;
end if;

```

**end subproc.**

The aim of subprocedure “*UnboundImproveUB()*” is twofold: to manage the unbounded sides of the feasible region (if any) and to improve variable  $UB$  with a smaller value. Specifically, the unbounded cases are recognized by means of the value  $\xi_{big} \gg 0$ . The feasible levels can then be first partitioned into three subsets, that are the two possibly unbounded ones  $[\xi_{min}, \xi_L]$  and  $[\xi_R, \xi_{max}]$  and the compact one  $[\xi_L, \xi_R]$ . The unbounded subproblems are solved by means of procedure “*SolveUnbound()*” which just need to visit a recession line. Variable  $UB$  is improved as much as possible by means of the visit of the unbounded subregions, if there are any, and by means of the feasible levels  $\xi_L$  and  $\xi_R$ . The compact interval  $[\xi_L, \xi_R]$  will be studied later in the algorithm.

### 4.2 Partitions priority queue

The proposed solution algorithm needs to analyze several partitions with different feasible levels intervals  $[\xi_1, \xi_2]$ . The partitions are stored in a priority queue,  $PQ$ , and analyzed following a priority order, that is to say that the partition with the smallest lower bound is visited first. The priority queue is managed by means of the following commands:

- $AddToPQ(LB, \xi_a, \xi_b, \xi_p, \hat{\theta}(\xi_a), \hat{\theta}(\xi_b), \hat{\theta}(\xi_p));$
- $(LB, \xi_1, \xi_2, \xi', \hat{\theta}(\xi_1), \hat{\theta}(\xi_2), \hat{\theta}(\xi')) := ExtractFromPQ().$

where  $[\xi_a, \xi_b]$  denotes the generic interval added to the queue in the various iterations, while  $[\xi_1, \xi_2]$  is the specific interval extracted from the queue in the single iteration.

The first command adds to the priority queue  $PQ$  the interval  $[\xi_a, \xi_b]$ , the level  $\xi_p \in [\xi_a, \xi_b]$  where the underestimation function  $\psi(\xi)$  reaches the minimum value, and the corresponding lower bound  $LB = \psi(\xi_p)$ . For the sake of convenience,  $\hat{\theta}(\xi_a)$ ,  $\hat{\theta}(\xi_b)$  and  $\hat{\theta}(\xi_p)$  are stored too. Clearly,  $PQ$  is a priority queue and hence the smaller is the value of the lower bound  $LB$ , the higher the priority of the partition in the queue  $PQ$  will be.

The second command removes from the priority queue  $PQ$  the interval  $[\xi_1, \xi_2]$  having the smaller lower bound  $LB$  and provides such an interval as the output, as well as the values of  $\xi'$ ,  $\hat{\theta}(\xi_1)$ ,  $\hat{\theta}(\xi_2)$  and  $\hat{\theta}(\xi')$ .

The following auxiliary subprocedure “*EvalLevels()*” is devoted to the analysis of a segment of feasible levels  $[\xi_a, \xi_b]$ .

**Subprocedure EvalLevels**(inputs:  $\xi_a, \xi_b, \hat{\theta}(\xi_a), \hat{\theta}(\xi_b)$ )

Let  $\psi(\xi) = \phi(\min\{\hat{\theta}(\xi_a), \hat{\theta}(\xi_b)\}, \xi);$

Let  $\xi_p := \arg \min_{\xi \in [\xi_a, \xi_b]} \psi(\xi); LB := \psi(\xi_p);$

if  $LB < UB$  then

    Let  $\hat{\theta}(\xi_p) := OptLevSol(\xi_p);$

    if  $(\xi_b - \xi_a) > \epsilon$  then

        if  $(\xi_p - \xi_a) \leq \epsilon$  and  $\hat{\theta}(\xi_a) \leq \hat{\theta}(\xi_b)$  then exit EvalLevels end if;

        if  $(\xi_b - \xi_p) \leq \epsilon$  and  $\hat{\theta}(\xi_b) \leq \hat{\theta}(\xi_a)$  then exit EvalLevels end if;

        AddToPQ  $(LB, \xi_a, \xi_b, \xi_p, \hat{\theta}(\xi_a), \hat{\theta}(\xi_b), \hat{\theta}(\xi_p));$

    end if;

end if;

**end subproc.**

Specifically, the underestimation function proposed in Subsection 4.5 is used to compute a lower bound  $LB$  corresponding to the interval  $[\xi_a, \xi_b]$  and an optimal level solution is stated with respect to the feasible level  $\xi_p$  giving the lower bound  $LB$  (eventually improving the value  $UB$  too). If  $LB \geq UB$ , then the interval  $[\xi_a, \xi_b]$  cannot improve the incumbent optimal solution and hence can be discarded. If  $(\xi_b - \xi_a) \leq \epsilon$ , with  $\epsilon > 0$  small enough ( $\epsilon = 2^{-30}$  is used in the computational test of the forthcoming section), the interval  $[\xi_a, \xi_b]$  can be discarded too. By *i*) and *ii*) of Theorem 12, if one of the following conditions holds:

- $\xi_p = \xi_a$  and  $\hat{\theta}(\xi_a) \leq \hat{\theta}(\xi_b)$
- $\xi_p = \xi_b$  and  $\hat{\theta}(\xi_b) \leq \hat{\theta}(\xi_a)$

then the optimal solution in  $[\xi_a, \xi_b]$  has been found and there is no need to further visit such an interval (notice the use of the tolerance  $\epsilon$  to manage numerical issues). Finally, if no stopping condition held then the interval is added to the priority queue  $PQ$  as well as the other useful values. Notice that subprocedure “*EvalLevels()*” is the only one which adds records to the priority queue  $PQ$ .

The initialization process of the priority queue  $PQ$  is described in subprocedure “*InitializePriorityQueue()*”.

**Subprocedure InitializePriorityQueue()**

```

Let  $PQ := \emptyset$ , compute  $\theta_{min} := \inf_{x \in X} c^T x + c_0$ ;
if  $\theta_{min} = -\infty$  then
    EvalLevels( $\xi_L, \xi_R, \hat{\theta}(\xi_L), \hat{\theta}(\xi_R)$ );
else
    Let  $\xi_{\theta_{min}}^L = \inf_{x \in X, c^T x + c_0 = \theta_{min}} \frac{d^T x + d_0}{b^T x + b_0}, \xi_{\theta_{min}}^R = \sup_{x \in X, c^T x + c_0 = \theta_{min}} \frac{d^T x + d_0}{b^T x + b_0}$ ;
    Let  $\xi_{\theta_{min}}^* = \arg \min_{\xi \in [\xi_{\theta_{min}}^L, \xi_{\theta_{min}}^R]} \phi(\theta_{min}, \xi)$  and  $\hat{\theta}(\xi_{\theta_{min}}^*)$  :
    = OptLevSol( $\xi_{\theta_{min}}^*$ );
    if  $\xi_L < \xi_{\theta_{min}}^L$  then EvalLevels( $\xi_L, \xi_{\theta_{min}}^L, \hat{\theta}(\xi_L), \theta_{min}$ ) end if;
    if  $\xi_{\theta_{min}}^R < \xi_R$  then EvalLevels( $\xi_{\theta_{min}}^R, \xi_R, \theta_{min}, \hat{\theta}(\xi_R)$ ) end if;
end if;
    
```

**end subproc.**

Recall that if  $S_\xi \neq \emptyset \forall \xi \in \Lambda$ , function  $\hat{\theta}(\xi), \xi \in \Lambda$ , is semistrictly quasiconvex (see Subsection 4.4). As a consequence, if  $\theta_{min} = -\infty$ , then  $\hat{\theta}(\xi)$  is strictly monotone in  $[\xi_L, \xi_R]$ , while if  $\theta_{min} > -\infty$ , then it is worth to divide interval  $[\xi_L, \xi_R]$  as follows:

$$[\xi_L, \xi_R] = [\xi_L, \xi_{\theta_{min}}^L] \cup [\xi_{\theta_{min}}^L, \xi_{\theta_{min}}^R] \cup [\xi_{\theta_{min}}^R, \xi_R]$$

so that  $\hat{\theta}(\xi)$  results to be strictly monotone in  $[\xi_L, \xi_{\theta_{min}}^L]$  and  $[\xi_{\theta_{min}}^R, \xi_R]$  (and an under-estimation function to compute  $LB$  is available) and constant in  $[\xi_{\theta_{min}}^L, \xi_{\theta_{min}}^R]$  (and just  $\sigma(\xi)$  needs to be analyzed).

**4.3 The main procedure**

The main solution procedure receives, as inputs, the parameters of problem  $P$  and provides, as outputs, a global optimal solution,  $Opt^*$ , and its value,  $Val^*$ .

**Procedure SolveP(inputs:  $P, \alpha$ ; outputs:  $Opt^*, Val^*$ )**

```

InitializeAlgorithm();
UnboundImproveUB();
InitializePriorityQueue();
while not (isEmpty(PQ)) do
    ( $LB, \xi_1, \xi_2, \xi', \hat{\theta}(\xi_1), \hat{\theta}(\xi_2), \hat{\theta}(\xi')$ ) := ExtractFromPQ();
    if  $LB \geq UB$  then  $PQ := \emptyset$ 
        else  $\xi_s := DetermineSplitLevel(\xi_1, \xi_2, \xi')$ ;
            SplitInterval( $\xi_1, \xi_2, \xi', \xi_s, \hat{\theta}(\xi_1), \hat{\theta}(\xi_2), \hat{\theta}(\xi')$ );
        end if
    end while;
    
```

Let  $Opt^* := Sol$ ,  $Val^* := UB$ ;

**end proc.**

In a preliminary phase, the algorithm is initialized by means of the subprocedures described in Subsections 4.1 and 4.2. In subprocedure “*InitializeAlgorithm()*” the feasible levels  $\xi_{min}$ ,  $\xi_{max}$  and  $\xi_0$  are computed. If there are no optimal level solutions ( $\inf_{x \in X_{\xi_0}} c^T x + c_0 = -\infty$ ), then an infimum is found and the whole procedure is closed.

Otherwise, variables  $Sol$  and  $UB$  are initialized by means of the optimal level solution corresponding to the feasible level  $\xi_0$ . Then, in subprocedure “*UnboundImproveUB()*” unboundedness is managed and the value of  $UB$  improved, finally in subprocedure “*InitializePriorityQueue()*” the priority queue  $PQ$  is initialized.

Various iterations are then executed, while the priority queue  $PQ$  is nonempty. The partition  $[\xi_1, \xi_2]$  having the smallest lower bound  $LB$  is extracted. If  $LB \geq UB$ , then all the partitions in the queue  $PQ$  result not to be able to improve the incumbent optimal solution and hence the queue itself can be emptied. If  $LB < UB$ , the interval is split in subprocedure “*SplitInterval()*”, where the new intervals are evaluated by means of subprocedure “*EvalLevels()*”, with respect to the level  $\xi_s$  given by subprocedure “*DetermineSplitLevel()*”. In the next section, different splitting criteria will be described and compared. At the end of the procedure, the final results are set. Notice that the correctness of this proposed method is given by the explicit or implicit visit of all the optimal level solutions. Moreover, the algorithm convergence will be guaranteed by requiring the new intervals determined in subprocedure “*SplitInterval()*” to have a length smaller than 90% the length of  $[\xi_1, \xi_2]$ .

## 5 Splitting criteria and computational results

The proposed algorithm has been fully implemented in a macOS 13.2.1 environment with an M1 Pro 10-core processor, MATLAB 2022b for coding and Gurobi 10.0.1 as solver for LP problems. Three different objective functions have been considered in the computational test:

- i)  $\phi_1(y_1, y_2) = y_1 y_2^4 - y_2^2$ ,
- ii)  $\phi_2(y_1, y_2) = y_1(1.5 + \sin(5 * y_2))$ ,
- iii)  $\phi_3(y_1, y_2) = \log(y_1)(1.5 + \cos(4 * y_2)) - \sqrt{y_2}$ .

Notice that these functions are not semistrictly quasiconvex and hence allow the presence of many local minima.

Problems with  $n = 50$  and  $n = 100$  variables have been considered, with  $2n$  corresponding box constraints and with a number  $m = 2n$  of further inequality constraints. Various instances (200 for  $n = 50$  and 200 for  $n = 100$  for each of the three considered objective functions) have been randomly generated and solved with all the considered splitting criteria, with a grand total of 19200 problems solved. Specifically, matrices and vectors  $A_{in} \in \mathbb{R}^{m \times n}$ ,  $b_{in} \in \mathbb{R}^m$ ,  $A_{eq} \in \mathbb{R}^{p \times n}$ ,  $b_{eq} \in \mathbb{R}^p$ ,  $l, u, a, b, \bar{c}, d \in \mathbb{R}^n$ , have been randomly generated with components in the interval  $[-10, 10]$  by using the “*randi()*” MATLAB function (integer numbers generated with uniform distribution). Values  $a_0, b_0, \bar{c}_0, d_0 \in \mathbb{R}$  are generated in order to guarantee that  $a^T y + a_0 > 0$  and

$b^T y + b_0 > 0$  for all  $y \in Y$ . The generated instances have used to solve Problem  $\bar{P}$  with each of the three objective functions  $\phi_1(y_1, y_2)$ ,  $\phi_2(y_1, y_2)$  and  $\phi_3(y_1, y_2)$ . The average number of visited partitions, the average number of computed optimal level solutions and the average time spent to solve the instances (obtained with the “tic” and “toc” MATLAB commands) are given as the result of the computational test. Values in bold emphasize the best performances obtained in the test.

### 5.1 The classical approach

The following splitting criteria for  $[\xi_1, \xi_2]$  are commonly used in the literature:

- split the interval with respect to the level  $\xi'$  which gave the lower bound  $LB$ , that is  $[\xi_1, \xi_2] = [\xi_1, \xi'] \cup [\xi', \xi_2]$ ; notice that this criteria may provide infinite loops if  $\xi' \approx \xi_2$  with  $\hat{\theta}(\xi_1) \leq \hat{\theta}(\xi_2)$  or  $\xi' \approx \xi_1$  with  $\hat{\theta}(\xi_2) \leq \hat{\theta}(\xi_1)$  (one of the new intervals would be almost equal to the old one), hence it will not be used in the forthcoming computational test;
- split the interval with respect to the middle level  $\xi_m = \frac{\xi_1 + \xi_2}{2}$ , that is  $[\xi_1, \xi_2] = [\xi_1, \xi_m] \cup [\xi_m, \xi_2]$ ; this criterion may be slow since does not take into account the info given by  $LB$ ;
- split the interval with respect to the level  $\xi_3 := \alpha \xi' + (1 - \alpha)\xi_m$ , with  $\alpha \in [0, 1]$ , that is  $[\xi_1, \xi_2] = [\xi_1, \xi_3] \cup [\xi_3, \xi_2]$ ; this linear combination of  $\xi'$  and  $\xi_m$  is aimed to improve the use of  $\xi_m$  by using the information implicitly given by  $\xi'$ ; to guarantee the convergence of the method,  $\alpha \in [0, 0.8]$  will be used (notice that  $\alpha = 1$  means that the interval is split with respect to  $\xi'$ ).

The following subprocedures “*DetermineSplitLevel<sub>1</sub>()*” and “*SplitInterval<sub>1</sub>()*” describe these criteria.

**Procedure DetermineSplitLevel<sub>1</sub>**(inputs:  $\xi_1, \xi_2, \xi'$ ; outputs:  $\xi_s$ )

Let  $\xi_m := \frac{\xi_1 + \xi_2}{2}$ ;  $\xi_s := \alpha \xi' + (1 - \alpha)\xi_m$ ;

**end proc.**

**Procedure SplitInterval<sub>1</sub>**(inputs:  $\xi_1, \xi_2, \xi', \xi_s, \hat{\theta}(\xi_1), \hat{\theta}(\xi_2), \hat{\theta}(\xi')$ )

if  $\xi' = \xi_s$  then  
     *EvalLevels*( $\xi_1, \xi', \hat{\theta}(\xi_1), \hat{\theta}(\xi')$ );  
     *EvalLevels*( $\xi', \xi_2, \hat{\theta}(\xi'), \hat{\theta}(\xi_2)$ );  
 else  $\hat{\theta}(\xi_s) := \text{OptLevSol}(\xi_s)$ ;  
     *EvalLevels*( $\xi_1, \xi_s, \hat{\theta}(\xi_1), \hat{\theta}(\xi_s)$ );  
     *EvalLevels*( $\xi_s, \xi_2, \hat{\theta}(\xi_s), \hat{\theta}(\xi_2)$ );  
 end if

**end proc.**

The results obtained in the computational test are summarized in Table 1.

**Table 1** “SplitInterval<sub>1</sub>”,  $0 \leq \alpha \leq 0.8$ 

n	$\alpha$	Num visited partitions			Num optimal level solutions			Elapsed times (secs)		
		$\phi_1$	$\phi_2$	$\phi_3$	$\phi_1$	$\phi_2$	$\phi_3$	$\phi_1$	$\phi_2$	$\phi_3$
50	0	<b>2175.1</b>	<b>862.13</b>	25.276	<b>5226.3</b>	<b>2048.3</b>	57.128	<b>43.200</b>	<b>16.508</b>	0.54059
50	0.2	2261.2	887.05	23.113	5414.1	2100.3	52.990	44.632	16.881	0.50162
50	0.4	2487.7	977.91	<b>22.291</b>	5902.5	2294.2	<b>51.512</b>	48.611	18.403	<b>0.48844</b>
50	0.6	3036.9	1198.7	23.631	7075.0	2764.3	54.350	58.291	22.159	0.51298
50	0.8	4668.1	1850.8	32.951	10531	4137.7	73.685	86.904	33.138	0.68248
100	0	<b>2284.8</b>	<b>848.23</b>	23.471	<b>5486.9</b>	<b>1973.0</b>	53.150	<b>121.51</b>	<b>42.877</b>	1.4423
100	0.2	2318.5	859.18	21.214	5549.0	1993.8	48.981	122.64	43.310	1.3369
100	0.4	2549.9	943.69	<b>19.519</b>	6039.5	2167.3	<b>45.602</b>	133.53	47.088	<b>1.2500</b>
100	0.6	3108.9	1154.3	20.044	7234.2	2610.1	47.019	159.97	56.707	1.2873
100	0.8	4800.3	1764.1	25.757	10816	3870.6	58.694	239.47	83.892	1.5807

Bold values emphasize the best performance for each function  $\phi$  and for a number of variables given by  $n = 50$  or  $n = 100$

It is worth noticing that:

- the number of visited partitions and the number of computed optimal level solutions do not vary so much with respect to the number of variables  $n$ . On the other hand, the number of variables strongly affects the time needed to solve the problem, due to the fact that the higher is the value of  $n$ , the higher is the time needed to compute an optimal level solution;
- performances strongly depend on the objective function;
- performances vary with respect to both the value of  $\alpha$  and the used objective function  $\phi$ . Specifically,  $\alpha = 0$  seems to be the choice for  $\phi_1$  and  $\phi_2$ , while  $\alpha = 0.4$  should be preferred for  $\phi_3$ ;
- performances clearly depend on the used underestimation function  $\psi(\xi)$ .

## 5.2 A costless improvement

In the previous subsection, the interval  $[\xi_1, \xi_2]$  is split with respect to the level  $\xi_s = \alpha\xi' + (1 - \alpha)\xi_m$ . Such a decision implies that  $\hat{\theta}(\xi')$  is not used, even if it is known. A first improvement of the previous splitting criteria could be the use of  $\xi'$  and  $\hat{\theta}(\xi')$  too, taking into account that they are known and hence their use is costless. Such an approach is described in the following subprocedure “SplitInterval<sub>2</sub>()”.

**Table 2** “SplitInterval<sub>2</sub>”,  $0 \leq \alpha \leq 0.8$

n	$\alpha$	Num visited partitions			Num optimal level solutions			Elapsed times (secs)		
		$\phi_1$	$\phi_2$	$\phi_3$	$\phi_1$	$\phi_2$	$\phi_3$	$\phi_1$	$\phi_2$	$\phi_3$
50	0	<b>2199.6</b>	<b>847.68</b>	20.468	<b>5280.2</b>	<b>2012.5</b>	48.099	<b>43.308</b>	<b>16.017</b>	0.46072
50	0.2	2249.6	879.18	18.453	5389.3	2082.9	44.232	44.124	16.507	0.42752
50	0.4	2478.7	973.07	<b>18.005</b>	5880.9	2279.9	<b>43.571</b>	48.126	17.964	<b>0.42141</b>
50	0.6	3037.4	1195.5	20.212	7079.9	2757.7	48.079	57.902	21.651	0.46301
50	0.8	4667.9	1859.8	30.399	10527	4159.1	69.473	85.999	32.307	0.65455
100	0	<b>2232.2</b>	<b>831.64</b>	18.500	<b>5352.0</b>	<b>1935.2</b>	44.039	<b>118.07</b>	<b>41.861</b>	1.2131
100	0.2	2308.9	856.59	16.029	5528.1	1987.0	39.223	122.07	42.958	1.0933
100	0.4	2557.2	942.34	<b>15.078</b>	6067.5	2165.0	<b>37.427</b>	133.95	46.516	<b>1.0488</b>
100	0.6	3121.7	1160.3	16.209	7261.7	2624.6	39.913	160.24	56.182	1.1110
100	0.8	4777.0	1775.8	22.791	10742	3903.3	53.816	236.83	83.312	1.4633

Bold values emphasize the best performance for each function  $\phi$  and for a number of variables given by  $n = 50$  or  $n = 100$

**Procedure SplitInterval<sub>2</sub>**(inputs:  $\xi_1, \xi_2, \xi', \xi_s, \hat{\theta}(\xi_1), \hat{\theta}(\xi_2), \hat{\theta}(\xi')$ )

```

if  $\xi' = \xi_s$  then
    EvalLevels( $\xi_1, \xi', \hat{\theta}(\xi_1), \hat{\theta}(\xi')$ );
    EvalLevels( $\xi', \xi_2, \hat{\theta}(\xi'), \hat{\theta}(\xi_2)$ );
else  $\hat{\theta}(\xi_s) := OptLevSol(\xi_s)$ ;
    if  $\xi' < \xi_s$  then
        EvalLevels( $\xi_1, \xi', \hat{\theta}(\xi_1), \hat{\theta}(\xi')$ );
        EvalLevels( $\xi', \xi_s, \hat{\theta}(\xi'), \hat{\theta}(\xi_s)$ );
        EvalLevels( $\xi_s, \xi_2, \hat{\theta}(\xi_s), \hat{\theta}(\xi_2)$ );
    else
        EvalLevels( $\xi_1, \xi_s, \hat{\theta}(\xi_1), \hat{\theta}(\xi_s)$ );
        EvalLevels( $\xi_s, \xi', \hat{\theta}(\xi_s), \hat{\theta}(\xi')$ );
        EvalLevels( $\xi', \xi_2, \hat{\theta}(\xi'), \hat{\theta}(\xi_2)$ );
    end if
end if
    
```

**end proc.**

The results obtained in the computational test are summarized in Table 2.

It is worth noticing that performances are slightly better than the ones in Table 1, since partitions may be smaller with no additional optimal level solutions needed to be computed.

**5.3 Some new splitting criteria**

Some more splitting criteria can be found by changing the approach in determining the split level  $\xi_s$ . So far,  $\xi_s = \alpha \xi' + (1 - \alpha) \xi_m$  has been used. Such splitting level does not use the position of  $\xi'$  with respect to the interval  $[\xi_1, \xi_2]$ , that is to say whether  $\xi'$  is “close” to either  $\xi_1, \xi_2$  or  $\xi_m$ . The underestimation function  $\psi(\xi) =$

$\phi(\min\{\hat{\theta}(\xi_a), \hat{\theta}(\xi_b)\}, \xi)$  is asymmetric, in the sense that it is tight on one side of the interval  $[\xi_1, \xi_2]$  and it does not fit on the other one. In light of this, the computational results given so far point out that it is better to split the interval almost in the middle to reduce the interval/error as fast as possible. On the other hand, the computational results emphasize that the fewer optimal level solutions are computed, the faster the solution method is. For this very reason, a possible idea is to split the interval with respect to  $\xi'$  whenever  $\xi'$  is “close” to  $\xi_m$ , so that no further optimal level solution has to be computed. If  $\xi'$  is not “close” to  $\xi_m$ , then we can split the interval  $[\xi_1, \xi_2]$  with respect to either  $\xi_m$  (in the middle),  $\frac{\xi_1 + \xi_m}{2}$  (on one side but not “close” to  $\xi_1$  to guarantee the convergence) or  $\frac{\xi_m + \xi_2}{2}$  (on the other side but not “close” to  $\xi_2$  to guarantee the convergence). Three different criteria  $C1$ ,  $C2$  and  $C3$  are described in the following subprocedure “*DetermineSplitLevel<sub>2</sub>*()”.

**Procedure DetermineSplitLevel<sub>2</sub>**(inputs:  $\xi_1, \xi_2, \xi'$ ; outputs:  $\xi_s$ )

```

Let position :=  $\frac{\xi' - \xi_1}{\xi_2 - \xi_1}$ ;
if position  $\in [0.25, 0.75]$  then  $\xi_s := \xi'$ 
  else case C1
    Let  $\xi_s := \frac{\xi_1 + \xi_2}{2}$ ;
  case C2
    if position < 0.25 then  $\xi_s := \xi_1 + 0.25(\xi_2 - \xi_1)$ 
    else  $\xi_s := \xi_1 + 0.75(\xi_2 - \xi_1)$ 
  end if;
  case C3
    if position < 0.25 then  $\xi_s := \xi_1 + 0.25(\xi_2 - \xi_1)$  else  $\xi_s := \frac{\xi_1 + \xi_2}{2}$  end if;
  end case;
end if

```

**end proc.**

The results obtained in the computational test are summarized in Table 3, where  $C1$ ,  $C2$  and  $C3$  refer to the use of subprocedures “*DetermineSplitLevel<sub>2</sub>*()” and “*SplitInterval<sub>1</sub>*()”, while  $C1bis$ ,  $C2bis$  and  $C3bis$  refer to the use of subprocedures “*DetermineSplitLevel<sub>2</sub>*()” and “*SplitInterval<sub>2</sub>*()”.

It is worth noticing that:

- the number of visited partitions and the number of computed optimal level solutions are a little bit decreased with respect to the best performances in Tables 1 and 2;
- the elapsed solution time is decreased too, since some optimal level solutions are not needed to be computed (when  $\xi'$  is “close” to  $\xi_m$ );
- $C1$  with subprocedure “*DetermineSplitLevel<sub>2</sub>*()” seems to be the best for the objective functions  $\phi_1$  and  $\phi_2$ , while  $C2$  with subprocedure “*DetermineSplitLevel<sub>2</sub>*()” seems to be used for  $\phi_3$  ( $C2$  allows the use of a splitting level  $\xi_s$  not close to  $\xi_m$  just like  $\alpha = 0.4$  does in Tables 1 and 2).



**Table 3** C1, C2, C3, “SplitInterval<sub>1</sub>” and “SplitInterval<sub>2</sub>”

n	Criteria	Num visited partitions			Num optimal level solutions			Elapsed times (secs)		
		$\phi_1$	$\phi_2$	$\phi_3$	$\bar{\phi}_1$	$\bar{\phi}_2$	$\bar{\phi}_3$	$\bar{\phi}_1$	$\bar{\phi}_2$	$\bar{\phi}_3$
50	C1	2185.0	854.88	25.606	5242.4	2025.9	56.911	42.438	15.510	0.52921
50	C2	2704.0	1058.6	21.039	6366.1	2463.2	48.103	51.437	18.759	0.45115
50	C3	2417.7	932.55	25.606	5746.8	2191.7	56.911	46.225	16.558	0.52579
50	C1bis	<b>2182.7</b>	<b>847.06</b>	20.941	<b>5238.9</b>	<b>2007.5</b>	48.798	<b>42.091</b>	<b>15.066</b>	0.45991
50	C2bis	2709.2	1061.6	<b>18.833</b>	6378.2	2470.1	<b>44.966</b>	51.193	18.534	<b>0.42832</b>
50	C3bis	2428.3	928.19	20.941	5768.5	2181.3	48.803	46.164	16.287	0.45736
100	C1	2197.4	835.06	24.248	5272.2	1937.0	53.937	115.63	41.045	1.4499
100	C2	2777.2	1034.5	18.131	6528.3	2355.5	41.845	143.24	49.658	1.1468
100	C3	2512.1	910.94	24.228	5956.1	2100.1	53.917	129.84	43.823	1.4419
100	C1bis	<b>2195.2</b>	<b>831.73</b>	18.947	<b>5269.3</b>	<b>1931.9</b>	44.680	<b>114.72</b>	<b>40.284</b>	1.2157
100	C2bis	2779.6	1032.4	<b>15.393</b>	6533.7	2352.3	<b>37.699</b>	142.48	49.055	<b>1.0447</b>
100	C3bis	2512.3	908.02	18.947	5956.9	2095.7	44.689	129.37	43.588	1.2126

Bold values emphasize the best performance for each function  $\phi$  and for a number of variables given by  $n = 50$  or  $n = 100$

### 5.4 Final remarks

The best performances are given by the new splitting criteria, that is, by the use of subprocedures “*DetermineSplitLevel<sub>2</sub>()*” and “*SplitInterval<sub>2</sub>()*”. Specifically speaking, C1 or C2 seem to be chosen depending on the objective function.

The obtained results are strictly related to the chosen underestimation function  $\psi(\xi)$ . In light of this, the study of underestimation functions tighter than the one used in this paper deserve to be further explored.

This paper extends the existing literature in several ways. First, problems  $P$  and  $\bar{P}$  are more general than the ones studied in Cambini (2020), Cambini and Venturi (2021), which can be seen as the particular case of  $\bar{P}$  where  $a = b = 0$ . In other words, the solution algorithm presented in this paper generalizes the one proposed in Cambini (2020), Cambini and Venturi (2021). From an application point of view, these problems extend and overcome the limitations of the ones presented in Cambini (2020), Cambini and Venturi (2021). For example,  $\bar{P}$  allows to approach in a very general way bicriteria efficiency models (Khalili et al. 2010), which need linear fractional functions to be managed in the objectives. It is worth pointing out that such a generalization is paid with a not necessarily convex region  $\Omega$ , which does not allow to determine tight underestimations.

Second, this paper covers and extends the class of functions presented in Shen and Lu (2018), Shen et al. (2019), though limited to the rank-two case. In  $\bar{P}$ , the objective function has two linear fractional functions scalarized by means of a generic two-variable function assumed to be strictly monotone with respect to the first variable. On the contrary, Shen and Lu (2018) consider the sum of linear fractional functions, while Shen et al. (2019) study the sum or product of linear fractional functions. In a future work, it could be interesting to computationally compare the general method proposed in this paper for two linear fractional functions with the ones proposed by

Shen and Lu (2018), Shen et al. (2019) for the particular cases of the sum or of the product of linear fractional functions.

## 6 Conclusions

In this paper a class of rank-two nonconvex problems involving linear fractional functions has been studied. Various theoretical results have been stated, generalizing recent studies concerning rank-two problems involving linear functions. These theoretical results allowed to propose a solution algorithm based on a partitioning approach. A complete computational experience has been provided, and new splitting criteria have been proposed and compared with the classical ones.

**Acknowledgements** Riccardo Cambini and Giovanna D'Inverno gratefully acknowledge financial support from the University of Pisa (grant number PRA\_2022–2023\_21). The authors also gratefully acknowledge the careful reviews by the anonymous referees.

**Funding** Open access funding provided by Università di Pisa within the CRUI-CARE Agreement.

## Declarations

**Conflict of interest** Riccardo Cambini and Giovanna D'Inverno declare that they have had no potential conflicts of interest in developing the paper and that the research did not involve human participants and/or animals.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Avriel, M., Diewert, W.E., Schaible, S., Zang, I.: Generalized concavity. *SIAM Class. Appl. Math.* (2010). <https://doi.org/10.1137/1.9780898719437>
- Bajalinov, E.B.: Linear-fractional programming theory, methods, applications and software. *Applied optimization*, vol. 84. Springer, Berlin (2003)
- Barros, A.I.: Discrete and fractional programming techniques for location models. *Combinatorial Optimization*, vol. 3. Kluwer Academic Publishers, Dordrecht (1998)
- Bomze, I.M., Csendes, T., Horst, R., (eds): *Developments in Global Optimization. Nonconvex Optimization and its Applications*, vol.18. Kluwer Academic Publishers, Dordrecht (1997)
- Cambini, A., Martein, L.: Generalized convexity and optimization: theory and applications. In: *Lecture notes in economics and mathematical systems*, vol. 616. Springer, Berlin (2009)
- Cambini, R., Sodini, C.: A finite algorithm for a particular d.c. quadratic programming problem. *Ann. Oper. Res.* **117**, 33–49 (2002). <https://doi.org/10.1023/A:1021509220392>
- Cambini, R., Sodini, C.: A finite algorithm for a class of nonlinear multiplicative programs. *J. Global Optim.* **26**, 279–296 (2003). <https://doi.org/10.1023/A:1023279306921>
- Cambini, R., Sodini, C.: A sequential method for a class of box constrained quadratic programming problems. *Math. Methods Oper. Res.* **67**(2), 223–243 (2008). <https://doi.org/10.1007/s00186-007-0173-x>

- Cambini, R., Sodini, C.: Global optimization of a rank-two nonconvex program. *Math. Methods Oper. Res.* **71**(1), 165–180 (2010). <https://doi.org/10.1007/s00186-009-0289-2>
- Cambini, R., Sodini, C.: On the minimization of a class of generalized linear functions on a flow polytope. *Optimization* **63**(10), 1449–1464 (2014). <https://doi.org/10.1080/02331934.2013.852548>
- Cambini, R., Carosi, L., Martein, L.: Generating the efficient frontier of a class of bicriteria generalized fractional programming. *Decis. Econ. Finan.* **40**(1–2), 81–101 (2017). <https://doi.org/10.1007/s10203-017-0196-6>
- Cambini, R., Carosi, L., Martein, L., Valipour, E.: Simplex-like sequential methods for a class of generalized fractional programs. *Math. Method Oper. Res.* **85**(1), 77–96 (2017). <https://doi.org/10.1007/s00186-016-0556-y>
- Cambini, R.: Underestimation functions for a rank-two partitioning method. *Decis. Econ. Finan.* **43**(2), 465–489 (2020). <https://doi.org/10.1007/s10203-020-00288-6>
- Cambini, R., Venturi, I.: A new solution method for a class of large dimension rank-two nonconvex programs. *IMA J. Manag. Math.* **32**(2), 115–137 (2021). <https://doi.org/10.1093/imaman/dpaa001>
- Charnes, A., Cooper, W.W.: Programming with linear fractional functionals. *Naval Res. Logist. Q.* **9**(3–4), 181–186 (1962). <https://doi.org/10.1002/nav.3800090303>
- Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. *Eur. J. Oper. Res.* **2**(6), 429–444 (1978). [https://doi.org/10.1016/0377-2217\(78\)90138-8](https://doi.org/10.1016/0377-2217(78)90138-8)
- Cooper, W.W., Seiford, L.M., Zhu, J., (eds.): *Handbook on Data Envelopment Analysis*. International Series in Operations Research & Management Science, vol.71, Kluwer Academic Publishers, Boston (2004)
- Frenk, J.B.G., Schaible, S.: Fractional programming. *Nonconvex Optimization and Its Applications*. In: *Handbook of generalized convexity and generalized monotonicity*. Springer, New York (2005)
- Horst, R., Pardalos, P.M. (eds.): *Handbook of Global Optimization*. *Nonconvex Optimization and its Applications*, vol.2, Kluwer Academic Publishers, Dordrecht (1995)
- Horst, R., Tuy, H.: *Global optimization: deterministic approaches*. Springer, Berlin (1996)
- Horst, R., Pardalos, P.M., Thoai, N.V.: *Introduction to Global Optimization*. In: *Nonconvex optimization and its applications*, vol. 48, 2nd edn. Kluwer Academic Publishers, Dordrecht (2001)
- Konno, H., Thach, P.T., Tuy, H.: *Optimization on low rank nonconvex structures, nonconvex optimization and its applications*, vol. 15. Kluwer Academic Publishers, Dordrecht (1997)
- Konno, H., Yajima, Y., Matsui, T.: Parametric simplex algorithms for solving a special class of nonconvex minimization problems. *J. Global Optim.* **1**, 65–82 (1991)
- Konno, H., Kuno, T.: Multiplicative programming problems. *Handbook of Global Optimization*. In: Horst, R., Pardalos, P.M. (eds.) *Nonconvex optimization and its applications*, vol. 2. Kluwer Academic Publishers, Dordrecht (1995). <https://doi.org/10.1007/978-1-4615-2025-2>
- Kuno, T.: A practical algorithm for minimizing a rank-two saddle function on a polytope. *J. Oper. Res. Soc. Jpn.* **39**, 63–76 (1996)
- Mjelde, K.M.: *Methods of the allocation of limited resources*. Wiley, New York (1983)
- Ryoo, H.-S., Sahinidis, N.V.: Global optimization of multiplicative programs. *J. Global Optim.* **26**, 387–418 (2003)
- Schaible, S.: Fractional programming. *Handbook of Global Optimization*. In: Horst, R., Pardalos, P.M. (eds.) *Nonconvex optimization and its applications*, vol. 2, pp. 495–608. Kluwer Academic Publishers, Dordrecht (1995)
- Shen, P., Lu, T.: Regional division and reduction algorithm for minimizing the sum of linear fractional functions. *J. Inequal. Appl.* (2018). <https://doi.org/10.1186/s13660-018-1651-9>
- Shen, P., Huang, B., Wang, L.: Range division and linearization algorithm for a class of linear ratios optimization problems. *J. Comput. Appl. Math.* **350**, 324–342 (2019). <https://doi.org/10.1016/j.cam.2018.10.038>
- Tuy, H.: *Convex Analysis and Global Optimization*. In: *Nonconvex optimization and its applications*. Kluwer Academic Publishers, Dordrecht (1998)
- Khalili, M., Camanho, A.S., Portela, M.C.A.S., Alirezaee, M.R.: The measurement of relative efficiency using data envelopment analysis with assurance regions that link inputs and outputs. *Eur. J. Oper. Res.* **203**(3), 761–770 (2010)