

# An Approach to Fractional Programming via D.C. Constraints Problem: Local Search

Tatiana Gruzdeva<sup>(✉)</sup> and Alexander Strekalovsky<sup>(✉)</sup>

Matrosov Institute for System Dynamics and Control Theory of SB RAS,  
Lermontov Str., 134, 664033 Irkutsk, Russia  
gruzdeva@icc.ru, strekal@icc.ru  
<http://nonconvex.isc.irk.ru>

**Abstract.** We consider the problem of optimizing the sum of several rational functions via reduction to a problem with d.c. constraints. We propose a method of finding a local solution to the fractional program which can be subsequently used in the global search method based on the global optimality conditions for a problem with nonconvex (d.c.) constraints [21–23]. According to the theory, we construct explicit representations of the constraints in the form of differences of two convex functions and perform a local search method that takes into account the structure of the problem in question. This algorithm was verified on a set of low-dimensional test problems taken from literature as well as on randomly generated problems with up to 200 variables and 200 terms in the sum.

**Keywords:** Nonconvex optimization · Rational optimization · d.c. representation · Local search · Linearized problems

## 1 Introduction

The fractional optimization is quite challenging and arises in various on a both economic and non-economic applications, whenever one or several ratios are required to be optimized. Let us mention a few examples mainly following the surveys by Schaible [8, 19, 20], where numerous other applications can be found. Numerators and denominators in ratios may represent cost, capital, profit, risk or time, etc. Fractional programs is closely related to the associated multiple-objective optimization problem, where a number of ratios are to be maximized simultaneously. Thus, the objective function in a fractional program can be considered as a utility function expressing a compromise between the different objective functions of the multiple-objective problem. Other applications include a multistage stochastic shipping problem [1, 7], profit maximization under fixed cost [4], various models in cluster analysis [17], multi-objective bond portfolio [13], and queuing location problems [5].

As known, without assumptions, the sum-of-ratios program is NP-complete [9]. Surveys on methods for solving this problem can be found in [8, 20, 28, 29]. According to the surveys, the majority of the methods make restrictive assumptions either on the concavity or linearity of the ratios. When the

ratios are nonlinear, the most popular techniques are based on the Branch and Bound approach, see e.g. [2, 6].

We propose reducing a fractional problem to the optimization problem with nonconvex constraints [10, 23], as it was mentioned in [6], with a subsequent application of the Global search theory for solving this class of nonconvex problems [21–23, 25, 26].

The outline of the paper is as follows. In Sect. 2 we substantiate the reduction of the sum-of-ratios fractional problem to the optimization problem with nonconvex constraints. Then in Sect. 3 we recall the local search method from [21], which implies linearization of the functions defining the basic non-convexity of the problem with d.c. constraints in the current point. In Sect. 4 we show how to explicitly represent the nonconvex functions, describing the constraints of the problem in question, as differences of two convex functions (the d.c. representation). The final section offers computational testing of the local search method on fractional program instances with a small number of variables and terms in the sum. We use the examples found in the literature as well as randomly generated problems of higher dimension.

## 2 Reduction to the Problem with Nonconvex Constraints

Now consider the following problem of the fractional optimization [3, 20]

$$f(x) := \sum_{i=1}^m \frac{\psi_i(x)}{\varphi_i(x)} \downarrow \min_x, \quad x \in S, \tag{P_0}$$

where  $\psi_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\varphi_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\varphi_i(x) > 0, \forall x \in S, i = 1, \dots, m$ , and  $S \subset \mathbb{R}^n$  is a convex set.

**Proposition 1.** (i) Let the pair  $(x_*, \alpha_*) \in \mathbb{R}^n \times \mathbb{R}^m$  be a solution to the following problem:

$$\sum_{i=1}^m \alpha_i \downarrow \min_{(x, \alpha)} \frac{\psi_i(x)}{\varphi_i(x)} = \alpha_i, \quad i = 1, \dots, m. \tag{P_1}$$

Then  $x_*$  is the solution to Problem (P<sub>0</sub>) and  $f(x_*) = \sum_{i=1}^m \alpha_{*i}$ .

(ii) Conversely, if  $x_*$  is the solution to Problem (P<sub>0</sub>) ( $x_* \in \text{Sol}(P_0)$ ), then the vector  $\alpha_* = (\alpha_{*1}, \dots, \alpha_{*m})^T \in \mathbb{R}^m$  defined as  $\alpha_{*i} = \frac{\psi_i(x_*)}{\varphi_i(x_*)}$ ,  $i = 1, \dots, m$ , is part of the solution  $(x_*, \alpha_*)$  to Problem (P<sub>1</sub>).

*Proof.* (i) Let  $(x_*, \alpha_*) \in \text{Sol}(P_1)$ , i.e.  $\alpha_{*i} = \frac{\psi_i(x_*)}{\varphi_i(x_*)}$ ,  $i = 1, \dots, m$ ,  $x_* \in S$ , and

$$\sum_{i=1}^m \alpha_{*i} \leq \sum_{i=1}^m \alpha_i \text{ for all } \alpha_i : \exists x \in S, \frac{\psi_i(x)}{\varphi_i(x)} = \alpha_i, i = 1, \dots, m,$$

$$f(x_*) = \sum_{i=1}^m \frac{\psi_i(x_*)}{\varphi_i(x_*)} = \sum_{i=1}^m \alpha_{*i} \leq \sum_{i=1}^m \alpha_i = \sum_{i=1}^m \frac{\psi_i(x)}{\varphi_i(x)} = f(x) \quad \forall x \in S.$$

Therefore,  $x_*$  is the solution to Problem  $(\mathcal{P}_0)$ .

(ii) Now let  $x_* \in \text{Sol}(\mathcal{P}_0)$ . Then

$$f(x_*) = \sum_{i=1}^m \frac{\psi_i(x_*)}{\varphi_i(x_*)} \leq \sum_{i=1}^m \frac{\psi_i(x)}{\varphi_i(x)} = f(x) \quad \forall x \in S. \tag{1}$$

Define  $\alpha_{*i} := \frac{\psi_i(x_*)}{\varphi_i(x_*)}$ ,  $i = 1, \dots, m$ , and consider the set

$$D_\alpha = \left\{ \alpha \in \mathbb{R}^m : \exists x \in S, \alpha_i = \frac{\psi_i(x)}{\varphi_i(x)} \right\}.$$

Then (1) implies  $\sum_{i=1}^m \alpha_{*i} = f(x_*) \leq f(x) = \sum_{i=1}^m \alpha_i \quad \forall \alpha \in D_\alpha, \forall x \in S$ . Therefore  $x_* \in \text{Sol}(\mathcal{P}_0)$ . □

**Proposition 2.** *Let the pair  $(x_*, \alpha_*) \in \mathbb{R}^n \times \mathbb{R}^m$  be a solution to the following problem:*

$$\sum_{i=1}^m \alpha_i \downarrow \min_{(x, \alpha)} \quad x \in S, \quad \frac{\psi_i(x)}{\varphi_i(x)} \leq \alpha_i, \quad i = 1, \dots, m. \tag{\mathcal{P}_2}$$

Then

$$\frac{\psi_i(x_*)}{\varphi_i(x_*)} = \alpha_{*i}, \quad i = 1, \dots, m. \tag{2}$$

*Proof.* Let  $(x_*, \alpha_*) \in \text{Sol}(\mathcal{P}_2)$ . Suppose that

$$\exists j \in \{1, 2, \dots, m\} : \frac{\psi_j(x_*)}{\varphi_j(x_*)} < \alpha_{*j}, \tag{3}$$

and construct  $\hat{\alpha} : \hat{\alpha}_i = \alpha_{*i} \quad \forall i \neq j, \hat{\alpha}_j = \frac{\psi_j(x_*)}{\varphi_j(x_*)}$ . It can be readily seen that  $(x_*, \hat{\alpha})$  is a feasible pair to Problem  $(\mathcal{P}_2)$ . The assumption (3) implies

$$\sum_{i=1}^m \alpha_{*i} = \sum_{i \neq j} \alpha_{*i} + \alpha_{*j} > \sum_{i=1}^m \hat{\alpha}_i.$$

It means that in Problem  $(\mathcal{P}_2)$  the pair  $(x_*, \hat{\alpha})$  is better than the pair  $(x_*, \alpha_*)$ , so,  $(x_*, \alpha_*) \notin \text{Sol}(\mathcal{P}_2)$ . This contradiction proves the equality (2). □

**Corollary 1.** *Any solution  $(x_*, \alpha_*) \in \mathbb{R}^n \times \mathbb{R}^m$  to Problem  $(\mathcal{P}_2)$  will be a solution to Problem  $(\mathcal{P}_1)$ , and, therefore, will be a solution to Problem  $(\mathcal{P}_0)$ .*

*Remark 1.* The inequality constraints in Problem  $(\mathcal{P}_2)$  can be replaced by equivalent constraints  $\psi_i(x) - \alpha_i \varphi_i(x) \leq 0, \quad i = 1, \dots, m$ , since  $\varphi_i(x) > 0 \quad \forall x \in S$ . It leads us to the following problem with  $m$  nonconvex constraints:

$$\sum_{i=1}^m \alpha_i \downarrow \min_{(x, \alpha)} \quad x \in S, \quad \psi_i(x) - \alpha_i \varphi_i(x) \leq 0, \quad i = 1, \dots, m. \tag{\mathcal{P}}$$

It is easy to see that Problem  $(\mathcal{P})$  is a global optimization problem with the nonconvex feasible set (see, e.g., [10, 23]), and we can apply the Global Search Theory for solving this class of nonconvex problems [21–23, 25, 26]. The Theory allows one to construct an algorithm for solving problems with nonconvex constraints. The algorithm contains two principal stages: (a) local search, which provides an approximately critical point; (b) procedures of escaping from critical points.

In the next section, we shall consider a local search method.

### 3 Local Search for Problem with D.C. Constraints

The local search methods (LSMs) play an important role in searching for the global solution to nonconvex problems, since it provides the so-called critical (stationary) points that might be considerably be better than a simple feasible point. Moreover, if a starting point occurs rather close to the global solution, then the LSMs can provide the global solution.

In order to find a local solution to Problem  $(\mathcal{P})$ , we apply a special LSM [21]. Let us consider the following problem with d.c. constraints:

$$\left. \begin{aligned} f_0(x) \downarrow \min_x, \quad x \in S, \\ f_i(x) := g_i(x) - h_i(x) \leq 0, \quad i \in I \triangleq \{1, \dots, m\}, \end{aligned} \right\} \quad (4)$$

where the functions  $f_0$  and  $g_i, h_i, i \in I$ , as well as the set  $S \subset \mathbb{R}^n$ , are convex. Further, suppose that the feasible set  $D := \{x \in S \mid f_i(x) \leq 0, \quad i \in I\}$  of the problem (4) is not empty and the optimal value  $\mathcal{V}(4) := \inf_x \{f_0(x) \mid x \in D\}$  of the problem (4) is finite:  $\mathcal{V}(4) > -\infty$ .

Furthermore, assume that a feasible starting point  $x^0 \in D$  is given and, in addition, after several iterations it has derived the current iterate  $x^s \in D, s \in \mathbb{Z}_+ = \{0, 1, 2, \dots\}$ .

In order to propose a local search method for the problem (4), apply a classical idea of linearization with respect to the basic nonconvexity of the problem (i.e. with respect to  $h_i(\cdot), i \in I$ ) at the point  $x^s$  [21]. Thus, we obtain the following linearized problem:

$$\left. \begin{aligned} f_0(x) \downarrow \min_x, \quad x \in S, \\ \varphi_{is}(x) := g_i(x) - \langle \nabla h_i(x^s), x - x^s \rangle - h_i(x^s) \leq 0, \quad i \in I. \end{aligned} \right\} \quad (\mathcal{P}\mathcal{L}_s)$$

Suppose the point  $x^{s+1}$  is provided by solving Problem  $(\mathcal{P}\mathcal{L}_s)$ , so that

$$x^{s+1} \in D_s = \{x \in S \mid g_i(x) - \langle \nabla h_i(x^s), x - x^s \rangle - h_i(x^s) \leq 0, \quad i \in I\}$$

and inequality  $f_0(x^{s+1}) \leq \mathcal{V}(\mathcal{P}\mathcal{L}_s) + \delta_s$  holds. Here  $\mathcal{V}(\mathcal{P}\mathcal{L}_s)$  is the optimal value to Problem  $(\mathcal{P}\mathcal{L}_s)$ :

$$\mathcal{V}_s := \mathcal{V}(\mathcal{P}\mathcal{L}_s) \triangleq \inf_x \{f_0(x) \mid x \in S, \varphi_{is}(x) \leq 0, \quad i \in I\},$$

and the sequence  $\{\delta_s\}$  satisfies the following condition functions  $\sum_{s=0}^{\infty} \delta_s < +\infty$ .

It can be easily seen that  $D_s \subset D$ , so  $x^{s+1}$  turns out to be feasible in the problem (4). Actually, since the functions  $h_i(\cdot)$ ,  $i \in I$  are convex, the following inequalities hold

$$\begin{aligned} 0 &\geq g_i(x^{s+1}) - \langle h'_i(x^s), x^{s+1} - x^s \rangle - h_i(x^s) = \varphi_{is}(x^{s+1}) \geq \\ &\geq g_i(x^{s+1}) - h_i(x^{s+1}) = f_i(x^{s+1}), \quad i \in I. \end{aligned}$$

Therefore, the LSM generates the sequence  $\{x^s\}$ ,  $x^s \in D_s$ ,  $s = 0, 1, 2, \dots$ , of solutions to Problems  $(\mathcal{P}\mathcal{L}_s)$ . As it was proven in [21], the cluster point  $x_* \in D_*$  of the sequence  $\{x^s\}$  is a solution to the linearized Problem  $(\mathcal{P}\mathcal{L}_*)$  (which is Problem  $(\mathcal{P}\mathcal{L}_s)$  with  $x^s$  instead of  $x_*$ ), and  $x_*$  can be called the critical point with respect to the LSM. Thus, the algorithm constructed in this way provides critical points by employing suitable convex optimization methods [15] for any given accuracy  $\tau$ . The following inequality:

$$f_0(x^s) - f_0(x^{s+1}) \leq \frac{\tau}{2}, \quad \delta_s \leq \frac{\tau}{2},$$

can be chosen as a stopping criterion for the LSM [21].

In order to implement the LSM, we need an explicit d.c. representation of  $f_i(\cdot)$ , i.e.  $f_i(\cdot) = g_i(\cdot) - h_i(\cdot)$ ,  $i \in I$ .

## 4 D.C. Representation of the Constraints

The first stage of any algorithm developed according the Global search theory is the decomposition of a nonconvex function as a difference of two convex functions. Such decomposition is constructing in several different ways depending on the functions  $\psi_i(\cdot)$  and  $\varphi_i(\cdot)$ .

### 4.1 Affine Functions

Let the functions  $\psi_i(\cdot)$  and  $\varphi_i(\cdot)$  be constructed by means of the vectors  $a^i, c^i \in \mathbb{R}^n$ , and numbers  $b_i, d_i \in \mathbb{R}$ ,

$$\psi_i(x) = \langle a^i, x \rangle + b_i, \quad \varphi_i(x) = \langle c^i, x \rangle + d_i > 0, \quad i \in I.$$

In this case, the basic nonconvexity of Problem  $(\mathcal{P})$  is the bilinear term  $\alpha_i \varphi_i(x) = \langle \alpha_i c^i, x \rangle + \alpha_i d_i$  in each constraint ( $i \in I$ ). Then, the bilinear function can be represented as a difference of two convex functions in the following way [27]:

$$\langle \alpha_i c^i, x \rangle = \frac{1}{4} \| \alpha_i c^i + x \|^2 - \frac{1}{4} \| \alpha_i c^i - x \|^2, \quad i \in I. \tag{5}$$

Hence, the functions  $f_i(\cdot)$  have the form  $f_i(x, \alpha_i) = g_i(x, \alpha_i) - h_i(x, \alpha_i)$ , where

$$\begin{aligned} g_i(x, \alpha_i) &= \frac{1}{4} \| \alpha_i c^i - x \|^2 - \alpha_i d_i + \langle a^i, x \rangle + b_i, \\ h_i(x, \alpha_i) &= \frac{1}{4} \| \alpha_i c^i + x \|^2. \end{aligned} \tag{6}$$

Taking into account the d.c. representation (6), the linearized Problem  $(\mathcal{PL}_s)$  has the following form

$$\left. \begin{aligned} & \sum_{i=1}^m \alpha_i \downarrow \min_{(x, \alpha)}, \quad x \in S, \\ & \frac{1}{4} \|\alpha_i c^i - x\|^2 + \langle a^i, x \rangle - \alpha_i d_i \langle \nabla h_i(x^s, \alpha_i^s), (x, \alpha_i) \rangle + \mathcal{C}_{is} \leq 0, \quad i \in I, \end{aligned} \right\} \quad (7)$$

where  $\mathcal{C}_{is} = b_i + \langle \nabla h_i(x^s, \alpha_i^s), (x^s, \alpha_i^s) \rangle - \frac{1}{4} \|\alpha_i^s c^i + x^s\|^2$ ,

$$\begin{aligned} \nabla h_{ix}(x^s, \alpha_i^s) &= (\nabla h_{ix}, \nabla h_{i\alpha})^T, \\ \nabla h_{ix} &= \frac{1}{2}(\alpha_i^s c^i + x^s), \quad \nabla h_{i\alpha} = \frac{1}{2}(\alpha_i^s \|c^i\|^2 + \langle c^i, x^s \rangle). \end{aligned} \quad (8)$$

The problem (7) is a convex optimization problem and it can be solved by an appropriate convex optimization method [15] at a given accuracy:  $\delta_s > 0$ ,  $s = 0, 1, \dots$

Further, we will consider Problem  $(\mathcal{P})$  where  $\psi_i(\cdot)$  are convex quadratic functions and  $\varphi_i(\cdot)$  are affine functions,  $i \in I$ .

## 4.2 Quadratic/Affine Functions

Suppose we are given symmetric positive definite matrices  $A^i$  ( $n \times n$ ), vectors  $p^i, c^i \in \mathbb{R}^n$ , and scalars  $q_i, d_i \in \mathbb{R}$ ,

$$\psi_i(x) = \langle x, A^i x \rangle + \langle p^i, x \rangle + q_i, \quad \varphi_i(x) = \langle c^i, x \rangle + d_i > 0, \quad i \in I.$$

As has been done in Subject. 4.1, we represent the bilinear term  $\alpha_i \varphi_i(x)$  as the difference of two convex functions, which yields us the d.c. representations  $f_i(x, \alpha_i) = g_i(x, \alpha_i) - h_i(x, \alpha_i)$ ,  $i \in I$ , where

$$\begin{aligned} g_i(x, \alpha_i) &= \langle x, A^i x \rangle + \langle p^i, x \rangle + q_i + \frac{1}{4} \|\alpha_i c^i - x\|^2 - \alpha_i d_i, \\ h_i(x, \alpha_i) &= \frac{1}{4} \|\alpha_i c^i + x\|^2. \end{aligned} \quad (9)$$

Taking into account the d.c. representation (9), the linearized Problem  $(\mathcal{PL}_s)$  takes the following form

$$\left. \begin{aligned} & \sum_{i=1}^m \alpha_i \downarrow \min_{(x, \alpha)}, \quad x \in S, \\ & \langle x, A^i x \rangle + \langle p^i, x \rangle + \frac{1}{4} \|\alpha_i c^i - x\|^2 - \alpha_i d_i \\ & - \langle \nabla h_i(x^s, \alpha_i^s), (x, \alpha_i) \rangle + \mathcal{C}_{is} \leq 0, \quad i \in I, \end{aligned} \right\} \quad (10)$$

where the gradient  $\nabla h_i(x^s, \alpha_i^s)$  is calculated by the formula (8), and  $\mathcal{C}_{is} = q_i + \langle \nabla h_i(x^s, \alpha_i^s), (x^s, \alpha_i^s) \rangle - \frac{1}{4} \|\alpha_i^s c^i + x^s\|^2$ .

The problem (10), as well as (7), can be solved by a suitable convex optimization method [15].

*Remark 2.* If the symmetric matrices  $A^i$  in the quadratic functions  $\psi_i(\cdot)$  are indefinite, then one can represent  $A^i$  as the difference of two symmetric positive definite matrices  $A^i = A_1^i - A_2^i$ ,  $A_1^i, A_2^i > 0$ , using, for example, a simple method from [24]. Afterwards, it is possible to construct functions  $g_i(\cdot)$  and  $h_i(\cdot)$  as follows: for all  $i \in I$  add the convex part with the matrix  $A_1^i$  to the function  $g_i(\cdot)$  and the nonconvex part with the matrix  $A_2^i$  to  $h_i(\cdot)$ .

In what follows, we will examine the case where  $\psi_i(\cdot)$  and  $\varphi_i(\cdot)$  are convex quadratic functions,  $i \in I$ .

### 4.3 Quadratic Functions

Now let us consider the following functions:

$$\psi_i(x) = \langle x, A^i x \rangle + \langle p^i, x \rangle + q_i, \quad \varphi_i(x) = \langle x, B^i x \rangle + \langle c^i, x \rangle + d_i > 0,$$

$A^i$  and  $B^i$  are positive definite ( $n \times n$ ) matrices,  $p^i, c^i \in \mathbb{R}^n$ ,  $q_i, d_i \in \mathbb{R}$ ,  $i \in I$ . Therefore, Problem (P) has the following term

$$\alpha_i \varphi_i(x) = \alpha_i \langle x, B^i x \rangle + \alpha_i \langle c^i, x \rangle + \alpha_i d_i, \tag{11}$$

which generate nonconvexity in every constraint ( $i \in I$ ).

The term  $\alpha_i \langle c^i, x \rangle$  in (11) can be presented in the d.c. form by the formula (5).

Further, let us denote  $r_i := \langle x, B^i x \rangle$ . Then, the product  $\alpha_i u_i$  can be expressed by formula (5) as follows

$$\begin{aligned} \alpha_i r_i &= \frac{1}{4}(\alpha_i + r_i)^2 - \frac{1}{4}(\alpha_i - r_i)^2 \\ &= \frac{1}{4}(\alpha_i + \langle x, B^i x \rangle)^2 - \frac{1}{4}(\alpha_i - \langle x, B^i x \rangle)^2, \quad i \in I. \end{aligned}$$

If  $B^i$ ,  $i \in I$ , are positive definite matrices and the following conditions hold

$$\alpha_i + \langle x, B^i x \rangle \geq 0, \quad \alpha_i - \langle x, B^i x \rangle \geq 0 \quad \forall x \in S, \quad i \in I, \tag{12}$$

then

$$\begin{aligned} g_i(x, \alpha_i) &= \frac{1}{4}(\alpha_i - \langle x, B^i x \rangle)^2 + \frac{1}{4} \|\alpha_i c^i - x\|^2 - \alpha_i d_i + \psi_i(x), \\ h_i(x, \alpha_i) &= \frac{1}{4}(\alpha_i + \langle x, B^i x \rangle)^2 + \frac{1}{4} \|\alpha_i c^i + x\|^2 \end{aligned}$$

are convex functions. Hence, we obtain the following d.c. representation:

$$f_i(x, \alpha_i) = g_i(x, \alpha_i) - h_i(x, \alpha_i), \quad i \in I, \tag{13}$$

and the following linearized Problem ( $\mathcal{PL}_s$ )

$$\left. \begin{aligned} &\sum_{i=1}^m \alpha_i \downarrow \min_{(x, \alpha)} \\ &\langle x, A^i x \rangle + \langle p^i, x \rangle + \frac{1}{4}(\alpha_i - \langle x, B^i x \rangle)^2 + \frac{1}{4} \|\alpha_i c^i - x\|^2 - \alpha_i d_i \\ &-\langle \nabla h_i(x^s, \alpha_i^s), (x, \alpha_i) \rangle + \mathcal{C}_{is} \leq 0, \quad i \in I, \end{aligned} \right\} \tag{14}$$

where  $\mathcal{C}_{is} = q_i + \langle \nabla h_i(x^s, \alpha_i^s), (x^s, \alpha_i^s) \rangle - \frac{1}{4} (\alpha_i^s + \langle x^s, B^i x^s \rangle)^2 - \frac{1}{4} \|\alpha_i^s c^i + x^s\|^2$ ,  $\nabla h_i(x^s, \alpha_i^s) = (\nabla h_{ix}, \nabla h_{i\alpha})^T$  is the gradient of the function  $h(\cdot)$ :

$$\begin{aligned} \nabla h_{ix} &= (\alpha_i^s + \langle x^s, B^i x^s \rangle) B^i x^s + \frac{1}{2} (\alpha_i^s c^i + x^s), \\ \nabla h_{i\alpha} &= \frac{1}{2} (\alpha_i^s + \langle x^s, B^i x^s \rangle) + \frac{1}{2} (\alpha_i^s \|c^i\|^2 + \langle c^i, x^s \rangle). \end{aligned}$$

If the conditions (12) are not satisfied, one can construct the d.c. representation (13) by decomposition of the trilinear term  $\alpha_i \langle x, B^i x \rangle = \sum_{l=1}^n \sum_{j=1}^n b_{lj}^i x_l x_j \alpha_i$  using the following equality holding for the product of three variables (for example,  $u, v, w$ ):

$$\begin{aligned} uvw &= \frac{1}{8} \hat{g}(u, v, w) - \frac{1}{8} \hat{h}(u, v, w), \\ \hat{g}(u, v, w) &= ((u + v)^2 + (1 + w)^2)^2 + (1 + w)^4 + (u^2 + w^2)^2 \\ &\quad + (v^2 + w^2)^2 + 2(u^2 + v^2), \\ \hat{h}(u, v, w) &= w^4 + ((u + v)^2 + w^2)^2 + (u^2 + (1 + w)^2)^2 \\ &\quad + (v^2 + (1 + w)^2)^2 + 2(u + v)^2. \end{aligned}$$

Therefore, we get (13), where

$$\begin{aligned} g_i(x, \alpha_i) &= \psi_i(x) + \frac{1}{4} \|\alpha_i c^i - x\|^2 - \alpha_i d_i + \frac{1}{8} \sum_{l=1}^n \sum_{j=1}^n b_{lj}^i \hat{g}(x_l, x_j, \alpha_i), \\ h_i(x, \alpha_i) &= \frac{1}{4} \|\alpha_i c^i + x\|^2 + \frac{1}{8} \sum_{l=1}^n \sum_{j=1}^n b_{lj}^i \hat{h}(x_l, x_j, \alpha_i). \end{aligned}$$

Obviously, in this case the linearized Problem ( $\mathcal{P}_{\mathcal{L}_s}$ ) is the problem of minimization of the linear function over the convex feasible set given by more complicated nonlinear functions  $\varphi_{ik}(x, \alpha_i)$  in comparison with the problems (7), (10) or even (14). At the same time, the linearized problems are convex, and therefore can be solved by a suitable convex optimization method [15].

*Remark 3.* If the symmetric matrices  $A^i$  and  $B^i$  in the functions  $\psi_i(\cdot)$  and  $\varphi_i(\cdot)$ , respectively, are indefinite, then this case is already described above in Remark 2.

## 5 Computational Simulations

The algorithm of the local search method (LSM) from Sect. 3 was coded in C++ language and was tested with various starting points. All computational experiments were performed on the Intel Core i7-4790K CPU 4.0 GHz.



At each iteration of the LSM, the convex Problem  $(\mathcal{P}\mathcal{L}_s)$  was solved by the software package IBM ILOG CPLEX 12.6.2 [11]. The accuracy of the LSM was  $\tau = 10^{-6}$ . The accuracy of the solution to the linearized problems  $(\mathcal{P}\mathcal{L}_s)$  increased during the LSM. Thus, we solved  $(\mathcal{P}\mathcal{L}_s)$  at a low accuracy at the first steps; further, the accuracy  $\delta_s$  was gradually improved ( $\delta_s \downarrow 0$ ), i.e.,  $\delta_0 = 0.1$ ,  $\delta_{s+1} = 0.5\delta_s$ , until the condition  $\delta_s \leq \frac{\tau}{2}$  was fulfilled with a given accuracy  $\tau > 0$ .

At the first stage, we numerically solved several instances of fractional programming problems from [2, 3, 7, 14, 16, 18] with a small number of variables.

### 5.1 Low-Dimensional Fractional Program with Affine Functions

Tables 1 and 2 represent the results of the computational testing of the LSM and employ the following designations:

- name* is the name of the test example;
- n* is the number of variables (problem’s dimension);
- m* is the number of terms in the sum;

**Table 1.** Low-dimensional fractional program. Minimization.

| name       | n | m | $f_0(x_0)$ | $f_0(z)$ | it | Time | $x_0$                 | $z$                   |
|------------|---|---|------------|----------|----|------|-----------------------|-----------------------|
| Prob3 [18] | 2 | 1 | 0.400      | 0.333    | 6  | 0.01 | (1.000; 0.000)        | (0.000; 0.000)        |
|            |   |   | 0.750      | 0.333    | 7  | 0.01 | (1.000; 1.000)        | (0.000; 0.000)        |
|            |   |   | 1.000      | 0.333    | 6  | 0.01 | (0.000; 1.000)        | (0.000; 0.000)        |
|            |   |   | 0.333      | 0.333    | 1  | 0.00 | (0.000; 0.000)        | (0.000; 0.000)        |
|            |   |   | 4.500      | 4.500    | 1  | 0.00 | (2.000; 1.000)        | (2.000; 1.000)        |
| Prob5 [18] | 2 | 2 | 4.156      | 4.500    | 5  | 0.01 | (2.250; 1.250)        | (2.000; 1.000)        |
|            |   |   | 6.500      | 4.500    | 7  | 0.01 | (1.000; 4.000)        | (2.000; 1.000)        |
|            |   |   | 5.000      | 4.500    | 5  | 0.01 | (1.000; 1.000)        | (2.000; 1.000)        |
|            |   |   | 1.733      | 1.623    | 12 | 0.02 | (0.000; 0.000)        | (0.000; 0.284)        |
| Prob3 [7]  | 2 | 2 | 2.758      | 1.623    | 11 | 0.02 | (0.750; 0.750)        | (0.000; 0.284)        |
|            |   |   | 2.400      | 1.623    | 14 | 0.02 | (0.500; 1.000)        | (0.000; 0.284)        |
|            |   |   | 4.250      | 1.623    | 18 | 0.02 | (0.000; 1.000)        | (0.000; 0.284)        |
|            |   |   | 2.830      | 2.830    | 1  | 0.00 | (1.500; 1.500)        | (1.500; 1.500)        |
| Prob3 [14] | 2 | 2 | 3.524      | 2.830    | 6  | 0.01 | (3.000; 4.000)        | (1.500; 1.500)        |
|            |   |   | 3.129      | 2.830    | 5  | 0.01 | (2.000; 2.333)        | (1.500; 1.500)        |
|            |   |   | 3.070      | 3.000    | 4  | 0.01 | (0.314; 0.842; 0.427) | (0.437; 0.000; 0.000) |
| Prob6 [14] | 3 | 3 | 3.035      | 3.000    | 3  | 0.01 | (0.900; 0.000; 0.633) | (0.952; 0.000; 0.000) |
|            |   |   | 3.000      | 3.000    | 1  | 0.00 | (1.100; 0.000; 0.000) | (1.098; 0.000; 0.000) |
|            |   |   | 2.895      | 2.889    | 4  | 0.01 | (0.000; 0.000; 2.000) | (0.513; 0.000; 1.795) |
| Prob7 [14] | 3 | 3 | 2.890      | 2.889    | 3  | 0.01 | (0.431; 0.000; 1.828) | (0.513; 0.000; 1.795) |
|            |   |   | 3.000      | 2.889    | 4  | 0.01 | (0.000; 1.111; 0.000) | (0.513; 0.000; 1.795) |

**Table 2.** Low-dimensional fractional program. Maximization.

| name       | n | m | $f_0(x_0)$ | $f_0(z)$     | it | Time | $x_0$                 | $z$                          |
|------------|---|---|------------|--------------|----|------|-----------------------|------------------------------|
| Prob1 [18] | 2 | 1 | 1.500      | 3.714        | 31 | 0.04 | (0.000; 2.500)        | (30.000; 0.000)              |
|            |   |   | 2.000      | 3.714        | 31 | 0.04 | (0.000; 0.000)        | (30.000; 0.000)              |
|            |   |   | 2.143      | 3.714        | 25 | 0.03 | (9.000; 7.000)        | (30.000; 0.000)              |
| Prob6 [2]  | 2 | 2 | 4.500      | <b>5.000</b> | 6  | 0.01 | (2.000; 1.000)        | <b>(1.000; 1.000)</b>        |
|            |   |   | 4.156      | <b>5.000</b> | 7  | 0.01 | (2.250; 1.250)        | <b>(1.000; 1.000)</b>        |
|            |   |   | 6.500      | 6.500        | 1  | 0.00 | (1.000; 4.000)        | (1.000; 4.000)               |
|            |   |   | 5.000      | <b>5.000</b> | 1  | 0.00 | (1.000; 1.000)        | <b>(1.000; 1.000)</b>        |
| Prob1 [7]  | 2 | 2 | 4.913      | 5.000        | 30 | 0.04 | (1.500; 1.500)        | (3.000; 4.000)               |
|            |   |   | 5.000      | 5.000        | 1  | 0.00 | (3.000; 4.000)        | (3.000; 4.000)               |
|            |   |   | 4.946      | 5.000        | 11 | 0.02 | (2.000; 2.333)        | (3.000; 4.000)               |
| Prob2 [7]  | 3 | 2 | 1.348      | 2.471        | 6  | 0.01 | (1.003; 0.731; 1.184) | (1.000; 0.000; 0.000)        |
|            |   |   | 1.879      | 2.471        | 6  | 0.01 | (1.500; 0.000; 0.500) | (1.000; 0.000; 0.000)        |
|            |   |   | 2.471      | 2.471        | 1  | 0.00 | (1.000; 0.000; 0.000) | (1.000; 0.000; 0.000)        |
|            |   |   | 2.107      | 2.471        | 5  | 0.01 | (0.750; 0.000; 0.250) | (1.000; 0.000; 0.000)        |
| Prob1 [16] | 3 | 3 | 2.988      | 3.003        | 30 | 0.05 | (0.414; 1.954; 0.000) | (0.000; 3.333; 0.000)        |
|            |   |   | 3.003      | 3.003        | 1  | 0.00 | (0.000; 3.333; 0.000) | (0.000; 3.333; 0.000)        |
|            |   |   | 2.947      | <b>3.000</b> | 4  | 0.01 | (0.512; 0.000; 0.610) | <b>(0.000; 0.082; 0.000)</b> |
|            |   |   | 2.963      | <b>3.000</b> | 4  | 0.01 | (1.000; 0.000; 0.000) | <b>(0.000; 0.114; 0.000)</b> |
| Prob2 [16] | 3 | 4 | 3.967      | 4.091        | 5  | 0.01 | (0.000; 0.000; 2.000) | (1.111; 0.000; 0.000)        |
|            |   |   | 4.000      | 4.091        | 4  | 0.01 | (0.000; 0.000; 0.000) | (1.111; 0.000; 0.000)        |
|            |   |   | 4.091      | 4.091        | 1  | 0.00 | (1.111; 0.000; 0.000) | (1.111; 0.000; 0.000)        |
|            |   |   | 3.868      | 4.091        | 5  | 0.01 | (0.000; 0.625; 1.875) | (1.111; 0.000; 0.000)        |

$f_0(x^0)$  is the value of the goal function to Problem ( $\mathcal{P}$ ) at the starting point;  
 $f_0(z)$  is the value of the function at the critical point provided by the LSM;  
 $it$  is the number of linearized problems solved (iterations of the LSM);  
 $Time$  stands for the CPU time of computing (seconds);  
 $x^0$  stands for the starting point chosen in the test problem;  
 $z$  is the critical point provided by the LSM.

Note that in the problems “Prob6 [2]” and “Prob1 [16]” in Table 2, local solutions derived by the LSM are not global (shown in bold).

Known global solutions to all problem instances were found just by the local search that confirms the computational effectiveness of the LSM. All test problems were successfully solved.

Further, we study if the LSM performance is affected by the increase in dimension of the variable  $x$  and the number of terms in the sum.

### 5.2 Randomly Generated Problems with Affine and Quadratic Functions

In this subsection, we will report computational results of testing the LSM on randomly generated problems of the form

$$f_0(x) := \sum_{i=1}^m \frac{\langle a^i, x \rangle + b_i}{\langle c^i, x \rangle + d_i} \uparrow \max_x, \quad \langle \bar{A}, x \rangle \leq \bar{b}, \quad x \geq 0. \tag{15}$$

Data  $a_j^i, c_j^i, \bar{A}_{lj} \in [0, 10]$  were uniformly random numbers,  $b_i = d_i = 10, \bar{b}_l = 10, i = 1, \dots, m, j = 1, \dots, n, l = 1, \dots, L$ .

Results of the computational testing of the LSM on fractional problems (15) up to 100 variables and 100 terms in the sum are listed in Table 3. The denotations in Table 3 are the same as in Tables 1 and 2.

**Table 3.** Randomly generated problems (15) with affine functions

| n   | m   | $f_0(x_0)$ | $f_0(z)$   | it  | Time   |
|-----|-----|------------|------------|-----|--------|
| 5   | 5   | 5.000000   | 5.659817   | 14  | 0.02   |
| 5   | 10  | 10.000000  | 11.399243  | 15  | 0.03   |
| 5   | 50  | 50.000000  | 56.107594  | 16  | 0.08   |
| 5   | 100 | 100.000000 | 106.644654 | 19  | 0.16   |
| 10  | 5   | 5.000000   | 5.560987   | 27  | 0.06   |
| 10  | 10  | 10.000000  | 12.368279  | 18  | 0.05   |
| 10  | 50  | 50.000000  | 57.873668  | 27  | 0.20   |
| 10  | 100 | 100.000000 | 106.665004 | 89  | 1.27   |
| 50  | 5   | 5.000000   | 7.286323   | 85  | 1.83   |
| 50  | 10  | 10.000000  | 12.572450  | 282 | 13.63  |
| 50  | 50  | 50.000000  | 58.460209  | 158 | 22.65  |
| 50  | 100 | 100.000000 | 109.059418 | 224 | 24.12  |
| 100 | 5   | 5.000000   | 6.809288   | 265 | 17.05  |
| 100 | 10  | 10.000000  | 13.774653  | 251 | 42.11  |
| 100 | 50  | 50.000000  | 56.692829  | 340 | 64.94  |
| 100 | 100 | 100.000000 | 109.858345 | 589 | 209.59 |

Moreover, we have carried out testing of the LSM on fractional problems with quadratic functions in the numerators of ratios. We generated the problems from [12] up to 200 variables and 200 terms in the sum:

$$f_0(x) := \sum_{i=1}^m \frac{\frac{1}{2}\langle x, A^i x \rangle + \langle p^i, x \rangle}{\langle c^i, x \rangle} \downarrow \min_x, \quad \langle \bar{A}, x \rangle \leq \bar{b}, \quad x \in [1, 5]^n, \tag{16}$$

**Table 4.** Randomly generated problems (16) with quadratic functions

| n   | m   | $f_0(x_0)$ | $f_0(z)$   | it  | Time   |
|-----|-----|------------|------------|-----|--------|
| 10  | 5   | 17.747630  | 15.607047  | 14  | 0.04   |
| 10  | 10  | 37.690672  | 35.242307  | 16  | 0.08   |
| 10  | 50  | 165.797927 | 155.938010 | 44  | 0.62   |
| 10  | 100 | 312.867828 | 296.561267 | 130 | 3.35   |
| 10  | 200 | 616.334973 | 601.908457 | 202 | 10.80  |
| 50  | 5   | 17.940091  | 15.460443  | 15  | 0.46   |
| 50  | 10  | 35.338013  | 30.663274  | 15  | 0.91   |
| 50  | 50  | 158.226948 | 151.224157 | 35  | 5.28   |
| 50  | 100 | 306.427872 | 297.535557 | 50  | 15.13  |
| 50  | 200 | 607.096322 | 589.882128 | 141 | 92.93  |
| 100 | 5   | 18.943321  | 15.771841  | 16  | 1.57   |
| 100 | 10  | 33.902949  | 29.500002  | 17  | 4.46   |
| 100 | 50  | 156.222645 | 148.923647 | 25  | 15.40  |
| 100 | 100 | 305.948925 | 296.692149 | 49  | 61.85  |
| 100 | 200 | 608.664712 | 591.959402 | 104 | 263.58 |
| 200 | 5   | 19.018280  | 15.512586  | 17  | 8.04   |
| 200 | 10  | 34.139152  | 29.172948  | 16  | 22.70  |
| 200 | 50  | 155.645907 | 146.144178 | 19  | 44.37  |
| 200 | 100 | 306.462441 | 295.558108 | 39  | 188.26 |
| 200 | 200 | 603.988798 | 587.865367 | 69  | 701.85 |

where  $A^i = U_i D^i U_i^T$ ,  $U_i = Q_1 Q_2 Q_3$ ,  $i = 1, \dots, m$ ,  $Q_j = I - 2 \frac{w_j w_j^T}{\|w_j\|^2}$ ,  $j = 1, 2, 3$  and  $w_1 = -i + \text{rand}(n, 1)$ ,  $w_2 = -2i + \text{rand}(n, 1)$ ,  $w_3 = -3i + \text{rand}(n, 1)$ ,  $D^i = \text{rand}(n, n)$ ,  $c^i = i - i \cdot \text{rand}(n, 1)$ ,  $p^i = i + i \cdot \text{rand}(n, 1)$ ,  $i = 1, \dots, m$ ,  $\bar{A} = -1 + 2 \cdot \text{rand}(5, n)$ ,  $\bar{b} = 2 + 3 \cdot \text{rand}(5, 1)$  [12]. (We denote by  $\text{rand}(k_1, k_2)$  the random matrix with  $k_1$  rows,  $k_2$  columns and elements generated randomly on  $[0, 1]$ .)

As it is shown in Table 4, the number of iteration (*it*) of the LSM is almost independent of the number of variables ( $n$ ) but approximately proportional to the number of terms in the sum ( $m$ ). The run-time increased proportionally to  $n$  and  $m$ .

Computational simulations confirm the efficiency of the LSM developed, the performance of which naturally depends on the choice of the method or the software package (IBM ILOG CPLEX) employed to solve auxiliary problems.

Thus, LSM can be applied in future implementations of the global search algorithm for solving the sum of ratios fractional problems via problems with d.c. constraints.

## 6 Conclusions

In this paper, we considered the fractional programming problem as an optimization problem with d.c. constraints. To this end, we carried out the explicit representation of nonconvex functions as differences of two convex functions and applied the local search algorithm based on linearization of the functions defining the basic non-convexity of the problem under study.

We investigated the effectiveness of the local search method for solving problems with d.c. constraints that generate the nonconvexity of the feasible set.

The numerical experiments demonstrated that the local search algorithm can globally solve low-dimensional sum-of-ratios test problems. Moreover, the local search algorithm developed in this paper turned out to be rather efficient at finding critical points in randomly generated fractional programming problems of high dimension.

Therefore, the method developed can be applied within the global search procedures for fractional programming problems.

**Acknowledgments.** This work has been supported by the Russian Science Foundation, Project N 15-11-20015.

## References

1. Almgoy, Y., Levin, O.: Parametric analysis of a multistage stochastic shipping problem. In: Lawrence, J. (ed.) *Operational Research*, vol. 69, pp. 359–370. Tavistock Publications, London (1970)
2. Benson, H.P.: Global optimization algorithm for the nonlinear sum of ratios problem. *J. Optim. Theory Appl.* **112**(1), 1–29 (2002)
3. Bugarin, F., Henrion, D., Lasserre, J.-B.: Minimizing the sum of many rational functions. *Math. Prog. Comp.* **8**, 83–111 (2016)
4. Colantoni, C.S., Manes, R.P., Whinston, A.: Programming, profit rates and pricing decisions. *Account. Rev.* **13**, 467–481 (1969)
5. Drezner, Z., Schaible, S., Simchi-Levi, D.: Queuing-location problems on the plane. *Naval Res. Log.* **37**, 929–935 (1990)
6. Dur, M., Horst, R., Thoai, N.V.: Solving sum-of-ratios fractional programs using efficient points. *Optimization* **49**, 447–466 (2001)
7. Falk, J.E., Palocsay, S.W.: Optimizing the sum of linear fractional functions. In: Floudas, C.A., Pardalos, P.M. (eds.) *Recent Advances in Global Optimization*, Princeton Series in Computer Science, pp. 221–257. Princeton University Press, Stanford (1992)
8. Frenk, J.B.G., Schaible, S.: Fractional programming. In: Hadjisavvas, S.S.N., Komlosi, S. (eds.) *Handbook of Generalized Convexity and Generalized Monotonicity*, Series Nonconvex Optimization and Its Applications, vol. 76, pp. 335–386. Springer, Heidelberg (2002)
9. Freund, R.W., Jarre, F.: Solving the sum-of-ratios problem by an interior-point method. *J. Global Optim.* **19**(1), 83–102 (2001)
10. Horst, R., Tuy, H.: *Global Optimization. Deterministic approaches*. Springer-Verlag, Berlin (1993)

11. Ibm, I.: High-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming. <http://www-03.ibm.com/software/products/en/ibmilogcpleoptstud>
12. Jong, Y.-Ch.: An efficient global optimization algorithm for nonlinear sum-of-ratios problems. *Optim. Online*. [http://www.optimization-online.org/DB\\_HTML/2012/08/3586.html](http://www.optimization-online.org/DB_HTML/2012/08/3586.html)
13. Konno, H.: Watanabe: Bond portfolio optimization problems and their applications to index tracking. *J. Oper. Res. Soc. Japan* **39**, 295–306 (1996)
14. Ma, B., Geng, L., Yin, J., Fan, L.: An effective algorithm for globally solving a class of linear fractional programming problem. *J. Softw.* **8**(1), 118–125 (2013)
15. Nocedal, J., Wright, S.J.: *Numerical Optimization*, 2nd edn. Springer, New York (2006)
16. Kuno, T.: A branch-and-bound algorithm for maximizing the sum of several linear ratios. *J. Global Optim.* **22**, 155–174 (2002)
17. Rao, M.R.: Cluster analysis and mathematical programming. *J. Amer. Statist. Assoc.* **66**, 622–626 (1971)
18. Raouf, O.A., Hezam, I.M.: Solving fractional programming problems based on swarm intelligence. *J. Ind. Eng. Int.* **10**, 56–66 (2014)
19. Schaible, S.: Fractional programming. In: Horst, R., Pardalos, P.M. (eds.) *Handbook of Global Optimization*, pp. 495–608. Kluwer Academic Publishers, Dordrecht (1995)
20. Schaible, S., Shi, J.: Fractional programming: the sum-of-ratios case. *Optim. Methods Softw.* **18**, 219–229 (2003)
21. Strekalovsky, A.S.: On local search in d.c. optimization problems. *Appl. Math. Comput.* **255**, 73–83 (2015)
22. Strekalovsky, A.S.: On solving optimization problems with hidden nonconvex structures. In: Rassias, T.M., Floudas, C.A., Butenko, S. (eds.) *Optimization in Science and Engineering*, pp. 465–502. Springer, New York (2014)
23. Strekalovsky, A.S.: Minimizing sequences in problems with D.C. constraints. *Comput. Math. Math. Phys.* **45**(3), 418–429 (2005)
24. Strekalovsky, A.S.: *Elements of nonconvex optimization* [in Russian]. Nauka, Novosibirsk (2003)
25. Strekalovsky, A.S., Gruzdeva, T.V.: Local search in problems with nonconvex constraints. *Comput. Math. Math. Phys.* **47**(3), 381–396 (2007)
26. Strekalovsky, A.S., Gruzdeva, T.V., Ulianova, N.Y.: *Optimization Problems with Nonconvex Constraints* [in Russian]. Irk. State University, Irkutsk (2013)
27. Strekalovsky, A.S., Orlov, A.V.: *Bimatrix games and bilinear programming* [in Russian]. FizMatLit, Moscow (2007)
28. Strekalovsky, A.S., Yakovleva, T.V.: On a local and global search involved in nonconvex optimization problems. *Autom. Remote Control* **65**, 375–387 (2004)
29. Wu, W.-Y., Sheu, R.-L., Birbil, I.S.: Solving the sum-of-ratios problem by a stochastic search algorithm. *J. Global Optim.* **42**(1), 91–109 (2008)