

RESEARCH

Open Access



Linear decomposition approach for a class of nonconvex programming problems

Peiping Shen^{1,2*} and Chunfeng Wang^{1,2}

*Correspondence:

shenpp@htu.edu.cn

¹College of Mathematics and Information Science, Henan Normal University, Xinxiang, 453007, P.R. China

²Henan Engineering Laboratory for Big Data Statistical Analysis and Optimal Control, Henan Normal University, Xinxiang, 453007, P.R. China

Abstract

This paper presents a linear decomposition approach for a class of nonconvex programming problems by dividing the input space into polynomially many grids. It shows that under certain assumptions the original problem can be transformed and decomposed into a polynomial number of equivalent linear programming subproblems. Based on solving a series of linear programming subproblems corresponding to those grid points we can obtain the near-optimal solution of the original problem. Compared to existing results in the literature, the proposed algorithm does not require the assumptions of quasi-concavity and differentiability of the objective function, and it differs significantly giving an interesting approach to solving the problem with a reduced running time.

MSC: 90C30; 90C33; 90C15

Keywords: nonconvex programming; global optimization; linear decomposition approach; approximation algorithm; computational complexity

1 Introduction

Consider a class of nonconvex programming problems:

$$(P) : \begin{cases} \min f(x) = \varphi(a_1^\top x, a_2^\top x, \dots, a_k^\top x), \\ \text{s.t. } x \in \Omega = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}, \end{cases}$$

where $k \geq 2$, $\varphi : \mathbb{R}^k \rightarrow \mathbb{R}_+$ is a continuous function, Ω is a nonempty polytope, $b \in \mathbb{R}^s$, $A \in \mathbb{R}^{s \times n}$, and $a_1, a_2, \dots, a_k \in \mathbb{R}^n$ are linear independent vectors. The function f is called a low-rank function with rank k over a polytope Ω defined by Kelner and Nikovola [1]. With this broader definition, multiplicative programming, quadratic programming, bilinear programming, as well as polynomial programming can be all put into the category of problem (P), whose important applications can be found in some surveys (e.g., [2–7]). In general, nonconvex programming problems of this form (P) are known to be NP-hard, even minimizing the product of two linear functions with rank two over a polytope is NP-hard ([8]). As shown by Mittal and Schulz [9], the optimum value of problem (P) cannot be approximated to within any factor unless $P = NP$. Hence, for solving problem (P) some extra assumptions (A1)–(A3) on the properties of the function f will be required as follows:

- (A1) $\varphi(y) \leq \varphi(y')$, if $y_i \leq y'_i$, for each $i = 1, \dots, k$;
 (A2) $\varphi(\lambda y) \leq \lambda^c \varphi(y)$ for all $y \in \mathbb{R}_+^k$, $\lambda > 1$ and some constant c ;
 (A3) $a_i^\top x > 0$, for $i = 1, \dots, k$.

An exhaustive reference on optimizing low-rank functions can be found in Konno and Thach [10]. Konno et al. [11] proposed cutting plane and tabu-search algorithms for low-rank concave quadratic programming problems. Porembski [12] gave a cutting plane solution approach for general low-rank concave minimization problems with a small number of variables. Additionally, some solution algorithms have been developed for the special cases of problem (P) (e.g. [13–16]). The above solution methods are efficient heuristics, without providing a theoretical analysis on the running time or performance of the algorithms.

The main purpose of this article is to present an approximation scheme with provable performance bounds for solving globally problem (P) to obtain an ε -approximate solution for any $\varepsilon > 0$ in time polynomial in the input size and $\frac{1}{\varepsilon}$. For the special cases of problem (P), there exists extensive work about the solution of ε -approximation problems. Vavasis [17] gave an approximation scheme for low-rank quadratic optimization problems. Depetrini and Locatelli [18] presented a fully polynomial-time approximation scheme (FPTAS) for minimizing the sum or product of ratios of linear functions over a polyhedron. Kelner and Nikolova [1] developed an expected polynomial-time smoothed algorithm for a class of low-rank quasi-concave minimization problems whose the objective function satisfies the Lipschitz condition. Daniele and Locatelli [19] proposed an FPTAS for minimizing product of two linear functions over a polyhedral set. Additionally, for minimizing the product of two non-negative linear cost functions, Goyal et al. [20] gave an FPTAS under the condition of the convex hull of the feasible solutions in terms of linear inequalities known. The algorithm in [21] works for minimizing a class of low-rank quasi-concave functions over a convex set, and this algorithm solves a polynomial number of linear optimization problems. Mittal and Schulz [9] presented an FPTAS for minimizing a general class of low-rank functions over a polytope, and their algorithm is based on constructing an approximate Pareto-optimal front of the linear functions that constitute the objective function.

In this paper, by exploiting the feature of problem (P), a suitable nonuniform grid for solving problem (P) is first constructed over a given $(k - 1)$ -dimensional box. Based on the exploration of the grid nodes, the original problem (P) can then be transformed and decomposed into a polynomial number of subproblems, in which each subproblem is corresponding to a grid node and is easy to solve by considering a linear program. Thus, the main computational effort of the proposed algorithm only consists in solving linear programming problems related to all nodes, which do not grow in size from a grid node to the next node. Furthermore, it is verified that through solving these linear programs, we can obtain an ε -approximation solution of the primal problem (P). The proposed algorithm has several features as follows. First, in contrast with [19, 20, 22], the rank k of the objective function considered by the proposed algorithm is not limited to only around two. Second, the proposed algorithm does not require differentiable and the inverse of the single variable function about the objective function, and it works for minimizing a class of more general functions, while Goyal and Ravi [21] and Kelner and Nikolova [1] both require the quasi-concavity assumption of the objective function. Third, although

the nonuniform grid constructed for the algorithms in [21] and ours is based on subdividing a $(k - 1)$ -dimensional hyper-rectangle, the algorithm in [21] requires iterations that are not necessary for our algorithm and the one in [9]. Moreover, at each iteration of the algorithm in [21], it is required to solve a single variable equation and the corresponding linear optimization problem for each grid node. Finally, we emphasize here that the efficiency of the algorithms (of [9, 21] and ours) strongly depends upon the number of grid nodes (or subproblems solved) that are associated with the dimension of the grid points, under the condition of the same input size and the tolerance ε value. In fact, the nonuniform grid in [9] derives from parting a k -dimensional hypercube. Therefore, from the procedure of the algorithm and its computational complexity analysis it can be seen that our work is independent of [9, 21] and the proposed algorithm differs significantly giving an interesting alternative approach to solve the problem with a reduced running time.

The structure of this paper is as follows. The next section describes the equivalent problem and its decomposition technique. Section 3 presents the algorithm and the computational cost of such an algorithm. Finally, some conclusions are drawn in Sections 4 and 5, and discussions presented.

2 Equivalent problem and its decomposition technique

2.1 Equivalent problem

For solving problem (P), we will propose an equivalent problem (P). To this end, let us firstly denote

$$l_i = \min_{x \in \Omega} a_i^\top x, \quad u_i = \max_{x \in \Omega} a_i^\top x, \quad i = 1, \dots, k. \tag{2.1}$$

Assume that, without loss of generality, $k = \arg \max\{\frac{u_i}{l_i} | i = 1, \dots, k\}$, and define a rectangle H given by

$$H = [l_1, u_1] \times [l_2, u_2] \times \dots \times [l_{k-1}, u_{k-1}]. \tag{2.2}$$

Thus, by introducing variable $y \in R^{k-1}$, problem (P) is equivalent to the following problem:

$$(Q) : \begin{cases} \min \varphi(y_1, \dots, y_{k-1}, a_k^\top x) \\ \text{s.t. } a_i^\top x \leq y_i, i = 1, \dots, k - 1, \\ x \in \Omega, \\ y = (y_1, \dots, y_{k-1}) \in H. \end{cases}$$

The key equivalent theorem for problems (P) and (Q) is given as follows.

Theorem 1 $x^* \in R^n$ is a global optimum solution of problem (P) if and only if $(x^*, y^*) \in R^{n+k-1}$ is a global optimum solution of problem (Q), where $y_i^* = a_i^\top x^*$ for each $i = 1, \dots, k - 1$. In addition, the global optimal values of problems (P) and (Q) are equal.

Proof If x^* is a global optimal solution of problem (P), let

$$y_i^* = a_i^\top x^*, \quad i = 1, \dots, k - 1.$$

It is obvious that $(x^*, y^*) \in R^{n+k-1}$ is a feasible solution of problem (Q). Let (x, y) be any feasible solution of problem (Q), i.e.,

$$a_i^\top x \leq y_i, \quad i = 1, \dots, k - 1, x \in \Omega. \tag{2.3}$$

According to the definition of y^* and the optimality of x^* , we must have

$$\varphi(y_1^*, \dots, y_{k-1}^*, a_k^\top x^*) = \varphi(a_1^\top x^*, \dots, a_{k-1}^\top x^*, a_k^\top x^*) \leq \varphi(a_1^\top x, \dots, a_{k-1}^\top x, a_k^\top x). \tag{2.4}$$

Additionally, from (2.3) and the assumption (A1), it follows that

$$\varphi(a_1^\top x, \dots, a_{k-1}^\top x, a_k^\top x) \leq \varphi(y_1, \dots, y_{k-1}, a_k^\top x). \tag{2.5}$$

Thus, (2.4) and (2.5) mean that (x^*, y^*) is a global optimal solution to problem (Q).

Conversely, suppose that (x^*, y^*) is a global optimal solution for problem (Q), then we have

$$a_i^\top x^* \leq y_i^*, \quad i = 1, \dots, k - 1, x^* \in \Omega.$$

By the assumption of φ , we can obtain

$$\varphi(a_1^\top x^*, \dots, a_{k-1}^\top x^*, a_k^\top x^*) \leq \varphi(y_1^*, \dots, y_{k-1}^*, a_k^\top x^*).$$

For any given $x \in \Omega$, if we let $y_i = a_i^\top x, i = 1, \dots, k - 1$, then (x, y) is a feasible solution to problem (Q) with $y = (y_1, \dots, y_{k-1}) \in R^{k-1}$. Thus, from the optimality of (x^*, y^*) it follows that

$$\begin{aligned} \varphi(a_1^\top x^*, \dots, a_{k-1}^\top x^*, a_k^\top x^*) &\leq \varphi(y_1^*, \dots, y_{k-1}^*, a_k^\top x^*) \\ &\leq \varphi(y_1, \dots, y_{k-1}, a_k^\top x) = \varphi(a_1^\top x, \dots, a_{k-1}^\top x, a_k^\top x). \end{aligned}$$

This means that x^* is a global optimal solution to problem (P). □

By Theorem 1, we can conclude that, for solving the problem (P), we may globally solving its equivalent problem (Q) instead. Besides, it is easy to understand that the problems (P) and (Q) have the same global optimal value. Hence, we will propose a decomposition approach for the problem (Q) below.

2.2 Linear decomposition technique

Problem (Q) has a relatively low-rank decomposition structure because, in contrast to problem (P), the nonconvexity of the objective function only involves the term $a_k^\top x$ if we fix a $y = (y_1, \dots, y_{k-1}) \in H$. In order to solve problem (Q), based on this observation, for any given $\theta \in (0, 1)$ we want to construct a polynomial size grid by subdividing H into smaller rectangles, such that the ratio of successive divisions is equal to $(1 + \theta)$ in each dimension. Thus, a polynomial size grid will be generated over H , where the set of the grid nodes can be given by

$$B^\theta = \{(v_1, v_2, \dots, v_{k-1}) | v_i \in \mathcal{D}_i^\theta, i = 1, \dots, k - 1\}, \tag{2.6}$$

where $\mathcal{D}_i^\theta = \{l_i, l_i(1 + \theta), \dots, l_i(1 + \theta)^{r_i}\}$ with

$$r_i = \arg \max \{ \tau \in \mathbb{N} | l_i(1 + \theta)^\tau \leq u_i \}, \quad i = 1, \dots, k - 1. \tag{2.7}$$

Note that under the assumption (A3), $l_i > 0$ must hold for each i . Clearly, for any $(y_1, y_2, \dots, y_{k-1}) \in H$, there exists a point $(v_1, v_2, \dots, v_{k-1}) \in B^\theta$ such that

$$y_i \in [v_i, (1 + \theta)v_i] \quad \text{for each } i = 1, \dots, k - 1.$$

Thus, H can be approximated by the set B^θ . Next, for each grid node $v \in B^\theta$, consider the corresponding subproblem as follows:

$$\text{P1}(v): \begin{cases} \omega(v) = \min \varphi(v, a_k^\top x), \\ \text{s.t. } a_i^\top x \leq v_i, i = 1, \dots, k - 1, \\ x \in \Omega. \end{cases}$$

Notice that, by the assumption (A1) of φ , for a given $v \in B^\theta$, problem P1(v) is equivalent to a linear problem P2(v):

$$\text{P2}(v): \begin{cases} \min a_k^\top x, \\ \text{s.t. } a_i^\top x \leq v_i, i = 1, \dots, k - 1, \\ x \in \Omega. \end{cases}$$

That is, for a fixed point $v \in B^\theta$, x^v is the optimal solution of problem P1(v) if and only if x^v is an optimal solution for problem P2(v).

Clearly, for each $v \in B^\theta$, the corresponding subproblems P1(v) can easily be solved by a linear program P2(v). Thus, we can decompose a nonconvex programming problem (Q) into a series of subproblems, and we can obtain its approximation global solution via the solutions of those linear programming problems when concerning all nodes v over B^θ .

3 Algorithm and its computational complexity

In this section, we will propose an effective algorithm for getting the approximation solution to problem (P), and then analyze its computational complexity.

3.1 ϵ -approximation algorithm

In what follows we will introduce an algorithm for solving problem (P), and the algorithm is able to return an ϵ -approximate solution of problem (P).

Based on the particularities of problem (P), a given rectangle H is firstly subdivided to construct a necessary nonuniform grid B^θ . The prime problem (P) can then transformed and decomposed into a series of subproblems on the basis of the exploration of the grid nodes. Each subproblem is associated with a grid node in the proposed algorithm, and it can be solved by a linear program. An necessary and specific description is given as follows. Given $\epsilon \in (0, 1)$, let $\theta = (1 + \epsilon)^{\frac{1}{\epsilon}} - 1$. The grid nodes set B^θ can be generated by (2.6)-(2.7). For each $v \in B^\theta$, solve problem P2(v) to get the solution x^v , and the optimal value to the corresponding problem P1(v) is denoted $\omega(v) = \varphi(v, a_k^\top x^v)$, here, let $\omega(v) = +\infty$ if

Algorithm 1 Algorithm statement

Step 0 (Initialization) Let $\varepsilon \in (0, 1)$, $\theta = (1 + \varepsilon)^{\frac{1}{c}} - 1$. Set $\tilde{L} = +\infty$.

Step 1 Sub-divide H into smaller hypercubes, such that the ratio of two successive divisions is $1 + \theta$ in each dimension. Denote the corner of each subhypercube $v = (v_1, \dots, v_{k-1})$, which is stored in the set B^θ .

Step 2 while $B^\theta \neq \emptyset$ **do begin**

Select $v = (v_1, \dots, v_{k-1}) \in B^\theta$, solve problem P2(v) to obtain an optimal solution x^v , and denote the optimal value $w(v) = \varphi(v, a_k^\top x^v)$ to the corresponding problem P1(v).

if $\tilde{L} > w(v)$

update $\tilde{x} = x^v$ and $\tilde{L} = w(v)$,

end if

$B^\theta = B^\theta \setminus \{v\}$.

end while.

the feasible set to P2(v) is empty. The process is repeated until all the points of B^θ are considered. The detailed algorithm is Algorithm 1.

The following theorem shows that the proposed algorithm can reach an optimal solution to problem (P).

Theorem 2 *Given $\varepsilon > 0$, an ε -optimal solution \tilde{x} to problem (P) from the proposed algorithm can be obtained in the sense that*

$$f(\tilde{x}) \leq (1 + \varepsilon)f(x^*),$$

where x^* is the optimal solution of problem (P).

Proof Let

$$y_i^* = a_i^\top x^*, \quad i = 1, \dots, k - 1. \tag{3.1}$$

From x^* being the optimal solution of problem (P), we have

$$l_i \leq y_i^* = a_i^\top x^* \leq u_i, \quad i = 1, \dots, k - 1.$$

This implies that $(y_1^*, y_2^*, \dots, y_{k-1}^*) \in H$, so there exists some $v^* \in B^\theta$ which satisfies

$$(1 + \theta)^{-1}v_i^* \leq y_i^* \leq v_i^*, \quad i = 1, \dots, k - 1. \tag{3.2}$$

Thus, combining with the assumptions of φ , we have

$$f(x^*) = \varphi(y_1^*, \dots, y_{k-1}^*, a_k^\top x^*) \geq (1 + \theta)^{-c}\varphi(v^*, (1 + \theta)a_k^\top x^*) \geq (1 + \theta)^{-c}\varphi(v^*, a_k^\top x^*). \tag{3.3}$$

Now, suppose that \tilde{x} is the optimal solution of problem P1(v^*). Then $x^* \in \Omega$ together with (3.1)-(3.2) implies that x^* is a feasible solution of problem P1(v^*). Thus we have

$$\varphi(v^*, a_k^\top x^*) \geq \varphi(v^*, a_k^\top \tilde{x}). \tag{3.4}$$

Additionally, let $\tilde{v} = \arg \min\{\omega(v) | v \in B^\theta\}$. Since \tilde{x} is the optimal solution of problem P1(\tilde{v}), it follows that $a_i^\top \tilde{x} \leq \tilde{v}_i, i = 1, \dots, k - 1$, thus, we can get

$$\varphi(\tilde{v}, a_k^\top \tilde{x}) \geq \varphi(a_1^\top \tilde{x}, a_2^\top \tilde{x}, \dots, a_k^\top \tilde{x}) = f(\tilde{x}). \tag{3.5}$$

According to the definitions of \tilde{v} and \tilde{x} , we have

$$\varphi(v^*, a_k^\top \tilde{x}) \geq \varphi(\tilde{v}, a_k^\top \tilde{x}). \tag{3.6}$$

Hence, from (3.3)-(3.6) and $\theta = (1 + \varepsilon)^{\frac{1}{\varepsilon}} - 1$, we can conclude that

$$f(\tilde{x}) \leq (1 + \varepsilon)f(x^*),$$

and so \tilde{x} is the approximation solution to problem (P). □

By Theorem 1 we also have the following corollary.

According to the above discussion, the ε -approximation solution to problem (P) can be obtained by solving $|B^\theta|$ (the number of grid nodes in B^θ) linear programming problems P2(v) with $v \in B^\theta$. However, it is not necessary to solve each P2(v) associated with each $v \in B^\theta$ for searching the solution of problem (P), that is, by using the following proposition we can obtain an improvement of the algorithm.

Proposition 1 *Let $\hat{x} = \arg \min\{a_k^\top x | x \in \Omega\}$. Then \hat{x} is an optimal solution of problem P1(v) for any $v \in \hat{B}^\theta$, where*

$$\hat{B}^\theta = \{v \in B^\theta | a_i^\top \hat{x} \leq v_i, i = 1, \dots, k - 1\}. \tag{3.7}$$

Proof Suppose that \bar{x}^v is any feasible solution of problem P1(v) with $v \in \hat{B}^\theta$. By using the definition of \hat{x} we can see that \hat{x} is a feasible solution of problem P1(v) for any $v \in \hat{B}^\theta$. With the increase of the function φ , it follows that

$$\varphi(v, a_k^\top \hat{x}) \leq \varphi(v, a_k^\top \bar{x}^v), \quad \forall v \in \hat{B}^\theta,$$

which concludes the proof. □

Proposition 1 shows that \hat{x} is the optimal solution of subproblem P1(v) for any $v \in \hat{B}^\theta$. Therefore, in practical implementations, we only are required to solve the subproblem P2(v) associated with the points contained in the set $B^\theta \setminus \hat{B}^\theta$. A further note on \hat{B}^θ is as follows.

For any $\theta \in (0, 1)$, by the definition of H , let

$$q_i = \arg \min\{p \in \mathbb{N} | \hat{y}_i \leq l_i(1 + \theta)^p \leq u_i\} \quad \text{for } i = 1, \dots, k - 1, \tag{3.8}$$

where $\hat{y}_i = a_i^\top \hat{x}$ with $\hat{x} = \arg \min\{a_k^\top x | x \in \Omega\}$. Combining the definition of $r_i, i = 1, \dots, k - 1$ with the above result, the set \hat{B}^θ can be given by

$$\hat{B}^\theta = \{(l_1(1 + \theta)^{\sigma_1}, \dots, l_{k-1}(1 + \theta)^{\sigma_{k-1}}) | \sigma_i \in \{q_i, \dots, r_i\}, i = 1, \dots, k - 1\}. \tag{3.9}$$

Algorithm 2 The improved algorithm

Step 0 (Initialization) Let $\varepsilon \in (0, 1)$, $\theta = (1 + \varepsilon)^{\frac{1}{\varepsilon}} - 1$, $\hat{x} = \arg \min\{a_k^\top x | x \in \Omega\}$.

Step 1 By using (2.4) and (3.7) get the grid nodes sets B^θ and \hat{B}^θ . Set $T^\theta = B^\theta \setminus \hat{B}^\theta$ and let $\tilde{L} = \varphi(v, a_k^\top \hat{x})$ with $v \in \hat{B}^\theta$.

Step 2 while $T^\theta \neq \emptyset$ **do begin**
 Select $v = (v_1, \dots, v_{k-1}) \in T^\theta$, solve problem P2(v) to obtain an optimal solution $x^{(v)}$. The optimal value to the corresponding problem P1(v) is denoted $\omega(v)$.
 if $\tilde{L} > \omega(v)$
 let $\tilde{x} = x^{(v)}$, $\tilde{L} = \omega(v)$,
 end of if
 $T^\theta = T^\theta \setminus \{v\}$.
end while.

Let

$$T^\theta = B^\theta \setminus \hat{B}^\theta. \tag{3.10}$$

This means the ε -approximation solution to problem (P) can be obtained only by solving $|T^\theta|$ (the number of points in the set T^θ) linear programming subproblems P2(v) for all $v \in T^\theta$. Thus the proposed algorithm can be improved by Algorithm 2.

Notice that, when the proposed improved algorithm stops, we can obtain an ε -optimal solution \tilde{x} to problem (P) with the objective value \tilde{L} .

3.2 Computational complexity for the algorithm

Now we consider the complexity analysis of the proposed improved algorithm. By (3.8)-(3.10), we can conclude that the number of the grid nodes belonging to \hat{B}^θ is at least

$$\prod_{i=1}^{k-1} \log_{(1+\theta)} \left(\frac{u_i}{\hat{y}_i} \right), \tag{3.11}$$

where $\hat{y}_i = a_i^\top \hat{x}$ with $\hat{x} = \arg \min\{a_k^\top x | x \in \Omega\}$. On the other hand, we know from (2.4) that the total number of the points in the set B^θ is equal to $\prod_{i=1}^{k-1} r_i$, r_i satisfying (2.5). Thus, it follows that the number of the elements in B^θ is at most

$$\prod_{i=1}^{k-1} \log_{(1+\theta)} \left(\frac{u_i}{l_i} \right). \tag{3.12}$$

Combining (3.9) with (3.10), the proposed improvement algorithm requires that the number of the grid nodes considered in actually computation is not more than

$$\Xi = \prod_{i=1}^{k-1} \left(\log_{(1+\theta)} \frac{u_i}{l_i} \right) - \prod_{i=1}^{k-1} \left(\log_{(1+\theta)} \frac{u_i}{\hat{y}_i} \right). \tag{3.13}$$

Theorem 3 Let $\hat{x} = \arg \min\{a_k^\top x | x \in \Omega\}$, $L = \min_{i=1, \dots, k-1} \{\frac{u_i}{\hat{y}_i}\}$ with $\hat{y}_i = a_i^\top \hat{x}$, and let $U = \max_{i=1, \dots, k-1} \frac{u_i}{l_i}$. When k is fixed, the running time of the improved algorithm for obtaining

an ε -optimal solution for problem (P), is bounded from above by

$$O\left(\log \frac{U}{L} \cdot \frac{c^{k-1}\xi^{k-2}}{\varepsilon^{k-1}} \cdot \text{cost}(|\pi|, n)\right),$$

where $\xi \in (\log L, \log U)$, and $\text{cost}(|\pi|, n)$ is the time taken to solve a linear program in n variables and input size of $|\pi|$ bits.

Proof By Step 0 of the improved algorithm, it follows that

$$\log_{(1+\theta)}\left(\frac{u_i}{\hat{y}_i}\right) \geq \log_{(1+\theta)} L = \frac{c \log L}{\log(1 + \varepsilon)}$$

and

$$\log_{(1+\theta)}\left(\frac{u_i}{l_i}\right) = \frac{\log(\frac{u_i}{l_i})}{\log(1 + \theta)} = \frac{c \log(\frac{u_i}{l_i})}{\log(1 + \varepsilon)} \leq \frac{c \log U}{\log(1 + \varepsilon)}.$$

From the above results and (3.13), we have

$$\begin{aligned} \Xi &\leq \left(\frac{c \log U}{\log(1 + \varepsilon)}\right)^{k-1} - \left(\frac{c \log L}{\log(1 + \varepsilon)}\right)^{k-1} \\ &= \frac{c^{k-1}}{[\log(1 + \varepsilon)]^{k-1}} [(\log U)^{k-1} - (\log L)^{k-1}]. \end{aligned} \tag{3.14}$$

Thus, the upper bound of the number of grid points Ξ is

$$\frac{c^{k-1}}{\varepsilon^{k-1}} [(\log U)^{k-1} - (\log L)^{k-1}]. \tag{3.15}$$

The result of (3.15) holds because $\log(1 + \varepsilon) \approx \varepsilon$ for small ε values. By using the Lagrange mean value theorem, there exists some $\xi \in (\log L, \log U)$ such that

$$(\log U)^{k-1} - (\log L)^{k-1} = (k - 1)\xi^{k-2}(\log U - \log L). \tag{3.16}$$

Thus we can know from (3.14)-(3.16) that the total number of the grid nodes considered in the improved algorithm is not more than

$$\frac{(k - 1)\xi^{k-2}c^{k-1}(\log U - \log L)}{\varepsilon^{k-1}}.$$

Note that $\log U$ and $\log L$ are computed in polynomial time about the input size of the problem. Additionally, for each grid node v in the set T^θ , a corresponding linear programming problem $P2(v)$ is required to solve. Therefore, for a fixed k , the running time required by the improved algorithm for obtaining an ε -optimal solution for problem (P), is bounded from above by

$$O\left(\log \frac{U}{L} \cdot \frac{c^{k-1}\xi^{k-2}}{\varepsilon^{k-1}} \cdot \text{cost}(|\pi|, n)\right), \tag{3.17}$$

where $\xi \in (\log L, \log U)$. □

In view of the above theorem we can conclude that the running time of the proposed improved algorithm is polynomial in input size and $\frac{1}{\epsilon}$ for fixed k , hence the algorithm is an FPTAS (fully polynomial-time approximation scheme) for the problem (P).

Comparison with [9, 21]: The algorithm in [9] searches for the optimal objective value in a k -dimensional grid, in which requires one to check the feasible of a linear program for each grid node, thus the total number of linear programs solved by their method is $O(\frac{c^k(\log \frac{M}{m})^k}{\epsilon^k})$ with $M \setminus m = \max \setminus \min\{a_i^\top x | x \in \Omega, i = 1, \dots, k\}$. In the algorithm [21], the number of linear optimization problems that are solved over a convex set in each iteration is $O(\frac{c^{k-1}(\log R)^{k-1}}{\epsilon^{k-1}})$, where $R = \max\{\frac{U_i}{L_i} | i = 1, \dots, k\}$. Also, at each iteration of their algorithm [21], the ratio of the upper and lower bounds of the objective value can be reduced by a constant factor, hence the number of iterations is $O(\frac{\epsilon}{\epsilon} \cdot \log \frac{z_U^0}{z_L^0})$, where $z_U^0(z_L^0)$ denotes the initial upper (lower) bound on the objective value. This implies that the algorithm in [21] solves $O(\log \frac{z_U^0}{z_L^0} \cdot \frac{c^k(\log R)^{k-1}}{\epsilon^k})$ linear optimization problems over a convex set. In this article, as can be seen in (3.17), the proposed algorithm solves $O(\log \frac{U}{L} \cdot \frac{c^{k-1}\epsilon^{k-2}}{\epsilon^{k-1}})$ different linear programs, and the running time is associated with $(k - 1)$ th order in $\frac{1}{\epsilon}$, compared with the k th order in $\frac{1}{\epsilon}$ in [9, 21].

4 Conclusions

In this article, we present a new linear decomposition algorithm for globally solving a class of nonconvex programming problems. First, the original problem is transformed and decomposed into a polynomial number of equivalent linear programming subproblems, by exploiting a suitable nonuniform grid. Second, compared with existing results in the literature, the proposed algorithm does not require the assumptions of quasi-concavity and differentiability of the objective function, and further, the rank k of the objective function is not limited to only around two. Finally, the computational complexity of the algorithm is given to show that it differs significantly giving an interesting alternative approach to solve the problem (P) with a reduced running time.

5 Results and discussion

In this work, a new linear decomposition algorithm for globally solving a class of nonconvex programming problems is presented. As further work, we think the ideas can be extended to more general type optimization problems, in which each $a_i^\top x$ in the objective function to problem (P) is replaced with a convex function.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

PPS carried out the idea of this paper, the description of linear decomposition algorithm and drafted the manuscript. CFW carried out the analysis of computational complexity of the algorithm. All authors read and approved the final manuscript.

Acknowledgements

The authors are grateful to the responsible editor and the anonymous referees for their valuable comments and suggestions, which have greatly improved the earlier version of this paper. This paper is supported by National Natural Science Foundation of China (11671122), the Program for Innovative Research Team (in Science and Technology) in University of Henan Province (14IRTSTHN023).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 6 February 2017 Accepted: 23 March 2017 Published online: 13 April 2017

References

1. Kelner, JA, Nikolova, E: On the hardness and smoothed complexity of quasi-concave minimization. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, Providence, RI, pp. 472-482 (2007)
2. Bennett, KP: Global tree optimization: a non-greedy decision tree algorithm. In: Computing Sciences and Statistics 26, pp. 156-160 (1994)
3. Bloemhof-Ruwaard, JM, Hendrix, EMT: Generalized bilinear programming: an application in farm management. *Eur. J. Oper. Res.* **90**, 102-114 (1996)
4. Konno, H, Shirakawa, H, Yamazaki, H: A mean-absolute deviation-skewness portfolio optimization model. *Ann. Oper. Res.* **45**, 205-220 (1993)
5. Maranas, CD, Androulakis, IP, Floudas, CA, Berger, AJ, Mulvey, JM: Solving long-term financial planning problems via global optimization. *J. Econ. Dyn. Control* **21**, 1405-1425 (1997)
6. Pardalos, PM, Vavasis, SA: Quadratic programming with one negative eigenvalue is NP-hard. *J. Glob. Optim.* **1**, 15-22 (1991)
7. Quesada, I, Grossmann, IE: Alternative bounding approximations for the global optimization of various engineering design problems. In: *Global Optimization in Engineering Design* 9, pp. 309-331 (1996)
8. Matsui, T: NP-hardness of linear multiplicative programming and related problems. *J. Glob. Optim.* **9**, 113-119 (1996)
9. Mittal, S, Schulz, AS: An FPTAS for optimization a class of low-rank functions over a polytope. *Math. Program.* **141**, 103-120 (2013)
10. Konno, H, Thach, PT, Tuy, H: *Optimization on Low Rank Nonconvex Structures*. Kluwer Academic, Dordrecht (1996)
11. Konno, H, Gao, C, Saitoh, I: Cutting plane/tabu search algorithms for low rank concave quadratic programming problems. *J. Glob. Optim.* **13**, 225-240 (1998)
12. Porembski, M: Cutting planes for low-rank like concave minimization problems. *Oper. Res.* **52**, 942-953 (2004)
13. Peiping, S, Chunfeng, W: Global optimization for sum of linear ratios problem with coefficients. *Appl. Math. Comput.* **176**(1), 219-229 (2006)
14. Wang, C, Shen, P: A global optimization algorithm for linear fractional programming. *Appl. Math. Comput.* **204**, 281-287 (2008)
15. Wang, C, Liu, S: A new linearization method for generalized linear multiplicative programming. *Comput. Oper. Res.* **38**(7), 1008-1013 (2011)
16. Jiao, HW, Liu, SY: A practicable branch and bound algorithm for sum of linear ratios problem, *Eur. J. Oper. Res.* **243**, 723-730 (2015)
17. Vavasis, SA: Approximation algorithms for indefinite quadratic programming. *Math. Program.* **57**, 279-311 (1992)
18. Depetrini, D, Locatelli, M: Approximation algorithm for linear fractional multiplicative problems. *Math. Program.* **128**, 437-443 (2011)
19. Depetrini, D, Locatelli, M: A FPTAS for a class of linear multiplicative problems. *Comput. Optim. Appl.* **44**, 275-288 (2009)
20. Goyal, V, Genc-Kaya, L, Ravi, R: An FPTAS for minimizing the product of two non-negative linear cost functions. *Math. Program.* **126**, 401-405 (2011)
21. Goyal, V, Ravi, R: An FPTAS for minimizing a class of low-rank quasi-concave functions over convex set. *Oper. Res. Lett.* **41**, 191-196 (2013)
22. Kern, W, Woeginger, G: Quadratic programming and combinatorial minimum weight product problem. *Math. Program.* **100**, 641-649 (2007)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
