POLYNOMIAL ALGORITHMS FOR PROJECTING A POINT ONTO A REGION DEFINED BY A LINEAR CONSTRAINT AND BOX CONSTRAINTS IN $\mathbb{R}^n$

STEFAN M. STEFANOV

Received 29 September 2003 and in revised form 6 April 2004

We consider the problem of projecting a point onto a region defined by a linear equality or inequality constraint and two-sided bounds on the variables. Such problems are interesting because they arise in various practical problems and as subproblems of gradient-type methods for constrained optimization. Polynomial algorithms are proposed for solving these problems and their convergence is proved. Some examples and results of numerical experiments are presented.

1. Introduction

Consider the problem of projecting a point $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n) \in \mathbb{R}^n$ onto a set defined by a linear inequality constraint "$\leq$", linear equality constraint, or linear inequality constraint "$\geq$" with positive coefficients and box constraints. This problem can be mathematically formulated as the following quadratic programming problem:

$$\min \left\{ c(\mathbf{x}) \equiv \sum_{j=1}^{n} c_j(x_j) \equiv \frac{1}{2} \sum_{j=1}^{n} (x_j - \hat{x}_j)^2 \right\} \tag{1.1}$$

subject to

$$\mathbf{x} \in X, \tag{1.2}$$

where the feasible region X is defined by

$$\sum_{j=1}^{n} d_j x_j \leq \alpha, \quad d_j > 0,\ j = 1, \dots, n, \tag{1.3}$$

$$a_j \leq x_j \leq b_j, \quad j = 1, \dots, n, \tag{1.4}$$

Journal of Applied Mathematics 2004:5 (2004) 409–431
2000 Mathematics Subject Classification: 90C30, 90C20, 90C25
URL: http://dx.doi.org/10.1155/S1110757X04309071

in the first case, by

$$\sum_{j=1}^{n} d_j x_j = \alpha, \quad d_j > 0,\ j = 1,\dots,n, \tag{1.5}$$

$$a_j \le x_j \le b_j, \quad j = 1,\dots,n, \tag{1.6}$$

in the second case, or by

$$\sum_{j=1}^{n} d_j x_j \ge \alpha, \quad d_j > 0,\ j = 1,\dots,n, \tag{1.7}$$

$$a_j \le x_j \le b_j, \quad j = 1,\dots,n, \tag{1.8}$$

in the third case.

Denote this problem by $(P^{\le})$ in the first case (problem (1.1)-(1.2) with X defined by (1.3)-(1.4)), by $(P^{=})$ in the second case (problem (1.1)-(1.2) with X defined by (1.5)-(1.6)), and by $(P^{\ge})$ in the third case (problem (1.1)-(1.2) with X defined by (1.7)-(1.8)).

Since $c(\mathbf{x})$ is a strictly convex function and X is a convex closed set, then this is a convex programming problem and it always has a *unique* optimal solution when $X \ne \emptyset$.

Problems of the form (1.1)-(1.2) with X defined by (1.3)-(1.4), (1.5)-(1.6), or (1.7)-(1.8) arise in production planning and scheduling (see [2]), in allocation of resources (see [2, 7, 8, 15]), in the theory of search (see [4]), in facility location (see [10, 11, 12, 13, 14]), and so forth. Problems $(P^{\le})$, $(P^{=})$, and $(P^{\ge})$ also arise as subproblems of some projection optimization methods of gradient (subgradient) type for constrained optimization when the feasible region is of the form (1.3)-(1.4), (1.5)-(1.6), or (1.7)-(1.8) (see, e.g., [6]). These projection problems are to be solved at *each* iteration of algorithm performance because current points generated by these methods must be projected on the feasible region at *each* iteration. That is why projection is the most onerous and time-consuming part of any projection gradient-type method for constrained optimization and we need efficient algorithms for solving these problems. This is the motivation to study the problems under consideration.

Problems like $(P^{\le})$, $(P^{=})$, and $(P^{\ge})$ are subject of intensive study. Related problems and methods for them are considered in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15].

An algorithm for finding a projection onto a simple polytope is proposed, for example, in [9]. Projections in the implementation of stochastic quasigradient methods are studied in [10]. Projected Newton-type methods are suggested in [1, 5].

This paper is devoted to the development of new efficient polynomial algorithms for finding a projection onto the set X defined by (1.3)-(1.4), (1.5)-(1.6), or (1.7)-(1.8). The paper is organized as follows. In Section 2, characterization theorems (necessary and sufficient conditions or sufficient conditions) for the optimal solutions to the considered problems are proved. In Section 3, new algorithms of polynomial complexity are suggested and their convergence is proved. In Section 4, we consider some theoretical and numerical aspects of implementation of the algorithms and give some extensions of both characterization theorems and algorithms. In Section 5, we present results of some numerical experiments.

2. Main results. Characterization theorems

2.1. Problem ($P^{\leq}$). First consider the following problem:

($P^{\leq}$)

$$\min\left\{c(\mathbf{x}) \equiv \sum_{j=1}^{n} c_j(x_j) \equiv \frac{1}{2}\sum_{j=1}^{n}(x_j - \hat{x}_j)^2\right\} \tag{2.1}$$

subject to (1.3) and (1.4).

Suppose that the following assumptions are satisfied:

(1.a) $a_j \leq b_j$ for all $j = 1,\dots,n$. If $a_k = b_k$ for some k, $1 \leq k \leq n$, then the value $x_k := a_k = b_k$ is determined in advance;

(1.b) $\sum_{j=1}^{n} d_j a_j \leq \alpha$; otherwise the constraints (1.3)-(1.4) are inconsistent and $X = \emptyset$ where X is defined by (1.3)-(1.4).

In addition to this assumption we suppose that $\alpha \leq \sum_{j=1}^{n} d_j b_j$ in some cases which are specified below.

The Lagrangian for problem ($P^{\leq}$) is

$$L(\mathbf{x},\mathbf{u},\mathbf{v},\lambda) = \frac{1}{2}\sum_{j=1}^{n}(x_j - \hat{x}_j)^2 + \lambda\left(\sum_{j=1}^{n} d_j x_j - \alpha\right) + \sum_{j=1}^{n} u_j(a_j - x_j) + \sum_{j=1}^{n} v_j(x_j - b_j), \tag{2.2}$$

where $\lambda \in \mathbb{R}^1_+$, $\mathbf{u},\mathbf{v} \in \mathbb{R}^n_+$, and $\mathbb{R}^n_+$ consists of all vectors with n real nonnegative components.

The Karush-Kuhn-Tucker (KKT) necessary and sufficient optimality conditions for the minimum $\mathbf{x}^* = (x_1^*,\dots,x_n^*)$ are

$$x_j^* - \hat{x}_j + \lambda d_j - u_j + v_j = 0, \quad j = 1,\dots,n, \tag{2.3}$$

$$u_j(a_j - x_j^*) = 0, \quad j = 1,\dots,n, \tag{2.4}$$

$$v_j(x_j^* - b_j) = 0, \quad j = 1,\dots,n, \tag{2.5}$$

$$\lambda\left(\sum_{j=1}^{n} d_j x_j^* - \alpha\right) = 0, \quad \lambda \in \mathbb{R}^1_+, \tag{2.6}$$

$$\sum_{j=1}^{n} d_j x_j^* \leq \alpha, \tag{2.7}$$

$$a_j \leq x_j^* \leq b_j, \quad j = 1,\dots,n, \tag{2.8}$$

$$u_j \in \mathbb{R}^1_+, \quad v_j \in \mathbb{R}^1_+, \quad j = 1,\dots,n. \tag{2.9}$$

Here, λ, u_j, v_j, $j = 1,\dots,n$, are the Lagrange multipliers associated with the constraints (1.3), $a_j \le x_j$, $x_j \le b_j$, $j = 1,\dots,n$, respectively. If $a_j = -\infty$ or $b_j = +\infty$ for some j, we do not consider the corresponding condition (2.4), (2.5) and Lagrange multiplier $u_j[v_j]$.

Since $\lambda \ge 0$, $u_j \ge 0$, $v_j \ge 0$, $j = 1,\dots,n$, and since the complementary conditions (2.4), (2.5), (2.6) must be satisfied, in order to find x_j^*, $j = 1,\dots,n$, from system (2.3)–(2.9), we have to consider all possible cases for λ, u_j, v_j: all λ, u_j, v_j equal to 0; all λ, u_j, v_j different from 0; some of them equal to 0 and some of them different from 0. The number of these cases is 2^{2n+1}, where $2n+1$ is the number of λ, u_j, v_j, $j = 1,\dots,n$. Obviously this is an enormous number of cases, especially for large-scale problems. For example, when $n = 1\,500$, we have $2^{3001} \approx 10^{900}$ cases. Moreover, in each case we have to solve a large-scale system of (nonlinear) equations in x_j^*, λ, u_j, v_j, $j = 1,\dots,n$. Therefore *direct* application of the KKT theorem, using explicit enumeration of all possible cases, for solving large-scale problems of the considered form would not give a result and we need results and efficient methods to cope with these problems.

The following theorem gives a characterization of the optimal solution to problem $(P^{\le})$. Its proof, of course, is based on the KKT theorem. As we will see in Section 5, by using Theorem 2.1, we can solve problem $(P^{\le})$ with $n = 10000$ variables in 0.00055 seconds on a personal computer.

Theorem 2.1 (characterization of the optimal solution to problem $(P^{\le})$). *A feasible solution $\mathbf{x}^* = (x_1^*,\dots,x_n^*) \in X$ (1.3)-(1.4) is the optimal solution to problem $(P^{\le})$ if and only if there exists some $\lambda \in \mathbb{R}^1_+$ such that*

$$x_j^* = a_j, \quad j \in J_a^\lambda \stackrel{\text{def}}{=} \left\{ j : \lambda \ge \frac{\hat{x}_j - a_j}{d_j} \right\}, \tag{2.10}$$

$$x_j^* = b_j, \quad j \in J_b^\lambda \stackrel{\text{def}}{=} \left\{ j : \lambda \le \frac{\hat{x}_j - b_j}{d_j} \right\}, \tag{2.11}$$

$$x_j^* = \hat{x}_j - \lambda d_j, \quad j \in J^\lambda \stackrel{\text{def}}{=} \left\{ j : \frac{\hat{x}_j - b_j}{d_j} < \lambda < \frac{\hat{x}_j - a_j}{d_j} \right\}. \tag{2.12}$$

Proof. (i) Let $\mathbf{x}^* = (x_1^*,\dots,x_n^*)$ be the optimal solution to $(P^{\le})$. Then there exist constants λ, u_j, v_j, $j = 1,\dots,n$, such that the KKT conditions (2.3)–(2.9) are satisfied. Consider both possible cases for λ.

(1) Let $\lambda > 0$. Then system (2.3)–(2.9) becomes (2.3), (2.4), (2.5), (2.8), (2.9), and

$$\sum_{j=1}^{n} d_j x_j^* = \alpha, \tag{2.13}$$

that is, the inequality constraint (1.3) is satisfied with an equality for x_j^*, $j = 1,\dots,n$, in this case.

(a) If $x_j^* = a_j$, then $u_j \ge 0$ and $v_j = 0$ according to (2.5). Therefore (2.3) implies $x_j^* - \hat{x}_j = u_j - \lambda d_j \ge -\lambda d_j$. Since $d_j > 0$, then

$$\lambda \ge \frac{\hat{x}_j - x_j^*}{d_j} \equiv \frac{\hat{x}_j - a_j}{d_j}. \tag{2.14}$$

(b) If $x_j^* = b_j$, then $u_j = 0$ according to (2.4) and $v_j \geq 0$. Therefore (2.3) implies $x_j^* - \hat{x}_j = -v_j - \lambda d_j \leq -\lambda d_j$. Hence

$$\lambda \leq \frac{\hat{x}_j - x_j^*}{d_j} \equiv \frac{\hat{x}_j - b_j}{d_j}. \tag{2.15}$$

(c) If $a_j < x_j^* < b_j$, then $u_j = v_j = 0$ according to (2.4) and (2.5). Therefore (2.3) implies $x_j^* = \hat{x}_j - \lambda d_j$. Since $d_j > 0$, $j = 1,\dots,n$, $\lambda > 0$ by the assumptions, then $\hat{x}_j > x_j^*$. It follows from $b_j > x_j^*$, $x_j^* > a_j$ that

$$\hat{x}_j - b_j < \hat{x}_j - x_j^*, \qquad \hat{x}_j - x_j^* < \hat{x}_j - a_j. \tag{2.16}$$

Using $d_j > 0$, we obtain $\lambda = (\hat{x}_j - x_j^*)/d_j < (\hat{x}_j - a_j)/d_j$, $\lambda = (\hat{x}_j - x_j^*)/d_j > (\hat{x}_j - b_j)/d_j$, that is,

$$\frac{\hat{x}_j - b_j}{d_j} < \lambda < \frac{\hat{x}_j - a_j}{d_j}. \tag{2.17}$$

(2) Let $\lambda = 0$. Then system (2.3)–(2.9) becomes

$$x_j^* - \hat{x}_j - u_j + v_j = 0, \quad j = 1,\dots,n, \tag{2.18}$$

and (2.4), (2.5), (2.7), (2.8), (2.9).

(a) If $x_j^* = a_j$, then $u_j \geq 0$, $v_j = 0$. Therefore $a_j - \hat{x}_j \equiv x_j^* - \hat{x}_j = u_j \geq 0$. Multiplying both sides of this inequality by $-(1/d_j)$ (< 0 by the assumption), we obtain

$$\frac{\hat{x}_j - a_j}{d_j} \leq 0 \equiv \lambda. \tag{2.19}$$

(b) If $x_j^* = b_j$, then $u_j = 0$, $v_j \geq 0$. Therefore $b_j - \hat{x}_j \equiv x_j^* - \hat{x}_j = -v_j \leq 0$. Multiplying this inequality by $-(1/d_j) < 0$, we get

$$\frac{\hat{x}_j - b_j}{d_j} \geq 0 \equiv \lambda. \tag{2.20}$$

(c) If $a_j < x_j^* < b_j$, then $u_j = v_j = 0$. Therefore $x_j^* - \hat{x}_j = 0$, that is, $x_j^* = \hat{x}_j$. Since $b_j > x_j^*$, $x_j^* > a_j$, $j = 1,\dots,n$, by the assumption, then

$$\hat{x}_j - b_j < \hat{x}_j - x_j^* = 0, \quad 0 = \hat{x}_j - x_j^* < \hat{x}_j - a_j. \tag{2.21}$$

Multiplying both inequalities by $1/d_j > 0$, we obtain

$$\frac{\hat{x}_j - b_j}{d_j} < 0 \equiv \lambda, \quad \lambda \equiv 0 < \frac{\hat{x}_j - a_j}{d_j}, \tag{2.22}$$

that is, in case (c) we have

$$\frac{\hat{x}_j - b_j}{d_j} < \lambda < \frac{\hat{x}_j - a_j}{d_j}. \tag{2.23}$$

In order to describe cases (a), (b), (c) for both (1) and (2), it is convenient to introduce the index sets J_a^λ, J_b^λ, J^λ defined by (2.10), (2.11), and (2.12), respectively. Obviously $J_a^\lambda \cup J_b^\lambda \cup J^\lambda = \{1,\dots,n\}$. The "necessity" part is proved.

(ii) Conversely, let $\mathbf{x}^* \in X$ and let components of $\mathbf{x}^*$ satisfy (2.10), (2.11), and (2.12), where $\lambda \in \mathbb{R}^1_+$.

(1) If $\lambda > 0$, then $x_j^* - \hat{x}_j < 0$, $j \in J^\lambda$, according to (2.12) and $d_j > 0$. Set

$$\begin{aligned} &\lambda = \frac{\hat{x}_j - x_j^*}{d_j}\ (>0) \text{ obtained from } \sum_{j\in J_a^\lambda} d_j a_j + \sum_{j\in J_b^\lambda} d_j b_j + \sum_{j\in J^\lambda} d_j(\hat{x}_j - \lambda d_j) = \alpha; \\ &u_j = v_j = 0 \quad \text{for } j \in J^\lambda; \\ &u_j = a_j - \hat{x}_j + \lambda d_j\ (\geq 0 \text{ according to the definition of } J_a^\lambda), \quad v_j = 0 \quad \text{for } j \in J_a^\lambda; \\ &u_j = 0, \quad v_j = \hat{x}_j - b_j - \lambda d_j\ (\geq 0 \text{ according to the definition of } J_b^\lambda) \quad \text{for } j \in J_b^\lambda. \end{aligned} \tag{2.24}$$

By using these expressions, it is easy to check that conditions (2.3), (2.4), (2.5), (2.6), (2.9) are satisfied; conditions (2.7) and (2.8) are also satisfied according to the assumption $\mathbf{x}^* \in X$.

(2) If $\lambda = 0$, then $x_j^* = \hat{x}_j$, $j \in J^\lambda$, according to (2.12), and

$$J^{\lambda=0} = \left\{ j : \frac{\hat{x}_j - b_j}{d_j} < 0 < \frac{\hat{x}_j - a_j}{d_j} \right\}. \tag{2.25}$$

Since $d_j > 0$, $\hat{x}_j - b_j < 0$, $\hat{x}_j - a_j > 0$, $j \in J^0$. Therefore $x_j^* = \hat{x}_j \in (a_j, b_j)$. Set

$$\begin{aligned} &\lambda = \frac{\hat{x}_j - x_j^*}{d_j}\ (=0), \quad u_j = v_j = 0 \quad \text{for } j \in J^{\lambda=0}, \\ &u_j = a_j - \hat{x}_j + \lambda d_j = a_j - \hat{x}_j\ (\geq 0), \quad v_j = 0 \quad \text{for } j \in J_a^{\lambda=0}, \\ &u_j = 0, \quad v_j = \hat{x}_j - b_j - \lambda d_j = \hat{x}_j - b_j\ (\geq 0) \quad \text{for } j \in J_b^{\lambda=0}. \end{aligned} \tag{2.26}$$

Obviously conditions (2.3), (2.4), (2.5), (2.9) are satisfied; conditions (2.7), (2.8) are also satisfied according to the assumption $\mathbf{x}^* \in X$, and condition (2.6) is obviously satisfied for $\lambda = 0$.

In both cases (1) and (2) of part (ii), x_j^*, λ, u_j, v_j, $j = 1,\dots,n$, satisfy KKT conditions (2.3)–(2.9) which are necessary and sufficient conditions for a feasible solution to be an optimal solution to a convex minimization problem. Therefore $\mathbf{x}^*$ is the (unique) optimal solution to problem $(P^{\leq})$. □

In view of the discussion above, the importance of Theorem 2.1 consists in the fact that it describes components of the optimal solution to $(P^{\leq})$ only through the Lagrange multiplier λ associated with the inequality constraint (1.3).

Since we do not know the optimal value of λ from Theorem 2.1, we define an iterative process with respect to the Lagrange multiplier λ and we prove convergence of this process in Section 3.

It follows from $d_j > 0$ and $a_j \le b_j$, $j = 1,\ldots,n$, that

$$ub_j \stackrel{\text{def}}{=} \frac{\hat{x}_j - b_j}{d_j} \le \frac{\hat{x}_j - a_j}{d_j} \stackrel{\text{def}}{=} la_j, \quad j = 1,\ldots,n, \tag{2.27}$$

for the expressions by means of which we define the sets J_a^λ, J_b^λ, J^λ.

The problem how to ensure a feasible solution to problem $(P^{\le})$, which is an assumption of Theorem 2.1, is discussed after the statement of the corresponding algorithm.

2.2. Problem $(P^{=})$**.** Consider problem $(P^{=})$ of finding a projection of $\hat{\mathbf{x}}$ onto a set X of the form (1.5)-(1.6):

$(P^{=})$

$$\min\left\{c(\mathbf{x}) \equiv \sum_{j=1}^{n} c_j(x_j) \equiv \frac{1}{2}\sum_{j=1}^{n}(x_j - \hat{x}_j)^2\right\} \tag{2.28}$$

subject to (1.5) and (1.6).

We have the following assumptions:

(2.a) $a_j \le b_j$ for all $j = 1,\ldots,n$;

(2.b) $\sum_{j=1}^{n} d_j a_j \le \alpha \le \sum_{j=1}^{n} d_j b_j$; otherwise the constraints (1.5)-(1.6) are inconsistent and the feasible region (1.5)-(1.6) is empty.

The KKT conditions for problem $(P^{=})$ are

$$\begin{gathered}
x_j^* - \hat{x}_j + \lambda d_j - u_j + v_j = 0, \quad j = 1,\ldots,n,\ \lambda \in \mathbb{R}^1, \\
u_j(a_j - x_j^*) = 0, \quad j = 1,\ldots,n, \\
v_j(x_j^* - b_j) = 0, \quad j = 1,\ldots,n, \\
\sum_{j=1}^{n} d_j x_j^* = \alpha, \\
a_j \le x_j^* \le b_j, \quad j = 1,\ldots,n, \\
u_j \in \mathbb{R}^1_+, \quad v_j \in \mathbb{R}^1_+, \quad j = 1,\ldots,n.
\end{gathered} \tag{2.29}$$

In this case the following theorem, which is analogous to Theorem 2.1, holds true.

Theorem 2.2 (characterization of the optimal solution to problem $(P^{=})$). *A feasible solution $\mathbf{x}^* = (x_1^*,\ldots,x_n^*) \in X$ (1.5)-(1.6) is the optimal solution to problem $(P^{=})$ if and only* *if there exists some* $\lambda \in \mathbb{R}^1$ *such that*

$$x_j^* = a_j, \quad j \in J_a^\lambda \overset{\text{def}}{=} \left\{ j : \lambda \geq \frac{\hat{x}_j - a_j}{d_j} \right\}, \tag{2.30}$$

$$x_j^* = b_j, \quad j \in J_b^\lambda \overset{\text{def}}{=} \left\{ j : \lambda \leq \frac{\hat{x}_j - b_j}{d_j} \right\}, \tag{2.31}$$

$$x_j^* = \hat{x}_j - \lambda d_j, \quad j \in J^\lambda \overset{\text{def}}{=} \left\{ j : \frac{\hat{x}_j - b_j}{d_j} < \lambda < \frac{\hat{x}_j - a_j}{d_j} \right\}. \tag{2.32}$$

The proof of Theorem 2.2 is omitted because it is similar to that of Theorem 2.1.

2.3. Problem $(P^{\geq})$**.** Consider problem $(P^{\geq})$ of finding a projection of $\hat{\mathbf{x}}$ onto a set X of the form (1.7)-(1.8):

$(P^{\geq})$

$$\min \left\{ c(\mathbf{x}) \equiv \sum_{j=1}^{n} c_j(x_j) \equiv \frac{1}{2} \sum_{j=1}^{n} (x_j - \hat{x}_j)^2 \right\} \tag{2.33}$$

subject to (1.7) and (1.8).

We have the following assumptions:

(3.a) $a_j \leq b_j$ for all $j = 1, \dots, n$;

(3.b) $\alpha \leq \sum_{j=1}^{n} d_j b_j$; otherwise constraints (1.7)-(1.8) are inconsistent and $X = \varnothing$, where X is defined by (1.7)-(1.8).

Rewrite $(P^{\geq})$ in the form (2.33), (2.34), (1.8), where

$$-\sum_{j=1}^{n} d_j x_j \leq -\alpha, \quad d_j > 0, \ j = 1, \dots, n. \tag{2.34}$$

Since the linear function $d(\mathbf{x}) := -\sum_{j=1}^{n} d_j x_j + \alpha$ is both convex and concave, $(P^{\geq})$ is a convex optimization problem.

Let λ, $\lambda^{\geq}$ be the Lagrange multipliers associated with (1.5) (problem $(P^{=})$) and with (2.34) (problem $(P^{\geq})$), and let x_j^*, $x_j^{\geq}$, $j = 1, \dots, n$, be components of the optimal solutions to $(P^{=})$, $(P^{\geq})$, respectively. For the sake of simplicity, we use u_j, v_j, $j = 1, \dots, n$, instead of $u_j^{\geq}$, $v_j^{\geq}$, $j = 1, \dots, n$, for the Lagrange multipliers associated with $a_j \leq x_j$, $x_j \leq b_j$, $j = 1, \dots, n$, from (1.8), respectively.

The Lagrangian for problem $(P^{\geq})$ is

$$\begin{aligned} L(\mathbf{x}, \mathbf{u}, \mathbf{v}, \lambda^{\geq}) = {} & \frac{1}{2} \sum_{j=1}^{n} (x_j - \hat{x}_j)^2 + \lambda^{\geq} \left(-\sum_{j=1}^{n} d_j x_j + \alpha \right) \\ & + \sum_{j=1}^{n} u_j (a_j - x_j) + \sum_{j=1}^{n} v_j (x_j - b_j) \end{aligned} \tag{2.35}$$

and the KKT conditions for $(P^{\geq})$ are

$$x_j^{\geq} - \hat{x}_j - \lambda^{\geq} d_j - u_j + v_j = 0, \quad j = 1,\dots,n, \tag{2.36}$$

$$u_j(a_j - x_j^{\geq}) = 0, \quad j = 1,\dots,n, \tag{2.37}$$

$$v_j(x_j^{\geq} - b_j) = 0, \quad j = 1,\dots,n, \tag{2.38}$$

$$\lambda^{\geq}\left(\alpha - \sum_{j=1}^{n} d_j x_j^{\geq}\right) = 0, \quad \lambda^{\geq} \in \mathbb{R}_+^1, \tag{2.39}$$

$$-\sum_{j=1}^{n} d_j x_j^{\geq} \leq -\alpha, \tag{2.40}$$

$$a_j \leq x_j^{\geq} \leq b_j, \quad j = 1,\dots,n, \tag{2.41}$$

$$u_j \in \mathbb{R}_+^1, \quad v_j \in \mathbb{R}_+^1, \quad j = 1,\dots,n. \tag{2.42}$$

We can replace (2.36) and (2.39) by

$$x_j^{\geq} - \hat{x}_j + \lambda^{\geq} d_j - u_j + v_j = 0, \quad j = 1,\dots,n, \tag{2.43}$$

$$\lambda^{\geq}\left(\sum_{j=1}^{n} d_j x_j^{\geq} - \alpha\right) = 0, \quad \lambda^{\geq} \in \mathbb{R}_-^1,\ d_j > 0, \tag{2.44}$$

respectively, where we have redenoted $\lambda^{\geq} := -\lambda^{\geq} \in \mathbb{R}_-^1$.

Conditions (2.43) with λ instead of $\lambda^{\geq}$, (2.37), (2.38), (2.41), (2.42) are among the KKT conditions for problem $(P^{=})$.

Theorem 2.3 (sufficient condition for optimal solution). (i) *If $\lambda = (\hat{x}_j - x_j^*)/d_j \leq 0$, then x_j^*, $j = 1,\dots,n$, solve problem $(P^{\geq})$ as well.*

(ii) *If $\lambda = (\hat{x}_j - x_j^*)/d_j > 0$, then $x_j^{\geq}$, $j = 1,\dots,n$, defined as*

$$\begin{gathered} x_j^{\geq} = b_j, \quad j \in J_b^{\lambda}; \\ x_j^{\geq} = \min\{b_j, \hat{x}_j\}, \quad j \in J^{\lambda}; \\ x_j^{\geq} = \min\{b_j, \hat{x}_j\} \quad \forall j \in J_a^{\lambda} \text{ such that } a_j < \hat{x}_j; \\ x_j^{\geq} = a_j \quad \forall j \in J_a^{\lambda} \text{ such that } a_j \geq \hat{x}_j \end{gathered} \tag{2.45}$$

solve problem $(P^{\geq})$.

Proof. (i) Let $\lambda = (\hat{x}_j - x_j^*)/d_j \leq 0$ (i.e., $\hat{x}_j \leq x_j^*$, $j \in J^{\lambda}$, because $d_j > 0$). Since x_j^*, $j = 1,\dots,n$, satisfy KKT conditions for problem $(P^{=})$ as components of the optimal solution to $(P^{=})$, then (2.43), (2.37), (2.38), (2.40) with equality (and therefore (2.44)), (2.41), (2.42) are satisfied as well (with λ instead of $\lambda^{\geq}$). Since they are the KKT necessary and sufficient conditions for $(P^{\geq})$, then x_j^*, $j = 1,\dots,n$, solve $(P^{\geq})$.

(ii) Let $\lambda = (\hat{x}_j - x_j^*)/d_j > 0$ (i.e., $\hat{x}_j > x_j^*$, $j \in J^{\lambda}$). Since $\mathbf{x}^* = (x_j^*)_{j=1}^n$ is the optimal solution to $(P^{=})$ by the assumption, then KKT conditions for $(P^{=})$ are satisfied.

If $\mathbf{x}^{\geq} := (x_j^{\geq})_{j=1}^n$ is the optimal solution to $(P^{\geq})$, then $\mathbf{x}^{\geq}$ satisfies (2.43), (2.37), (2.38), (2.44), (2.40), (2.41), (2.42). Since $\lambda > 0$, then λ cannot play the role of $\lambda^{\geq}$ in (2.43) and (2.44) because $\lambda^{\geq}$ must be a nonpositive real number in (2.43) and (2.44). Therefore x_j^*, which satisfy KKT conditions for problem $(P^=)$, cannot play the roles of $x_j^{\geq}$, $j = 1,\ldots,n$, in (2.43), (2.37), (2.38), (2.44), (2.40), (2.41), (2.42). Hence, in the general case the equality $\sum_{j=1}^n d_j x_j = \alpha$ is not satisfied for $x_j = x_j^{\geq}$. Therefore, in order that (2.44) be satisfied, $\lambda^{\geq}$ must be equal to 0. This conclusion helps us to prove the theorem.

Let $\mathbf{x}^{\geq} := (x_j^{\geq})_{j=1}^n$ be defined as in part (ii) of the statement of Theorem 2.3.

Set $\lambda^{\geq} = 0$;

(1) $u_j = 0$, $v_j = \hat{x}_j - b_j$ (≥ 0 according to the definition of J_b^λ (2.31), $\lambda > 0$, $d_j > 0$) for $j \in J_b^\lambda$;
(2) $u_j = v_j = 0$ for $j \in J_a^\lambda$ such that $a_j < \hat{x}_j$ and for $j \in J^\lambda$ such that $\hat{x}_j < b_j$;
(3) $u_j = 0$, $v_j = \hat{x}_j - b_j$ (≥ 0) for $j \in J^\lambda$ such that $\hat{x}_j \geq b_j$;
(4) $u_j = a_j - \hat{x}_j$ (≥ 0), $v_j = 0$ for $j \in J_a^\lambda$ such that $a_j \geq \hat{x}_j$.

In case (2) we have $a_j < \hat{x}_j$, therefore $a_j < \hat{x}_j = x_j^{\geq}$ according to the definition of $x_j^{\geq}$ in this case. In case (3), since $b_j \leq \hat{x}_j$, that is, $b_j - \hat{x}_j \leq 0$, then $v_j := \hat{x}_j - b_j \geq 0$. Consequently, conditions (2.41) and (2.42) are satisfied for all j according to (1), (2), (3), and (4).

As we have proved, (2.44) is satisfied with $\lambda^{\geq} = 0$. Since the equality constraint (1.5) $\sum_{j=1}^n d_j x_j^* = \alpha$ is satisfied for the optimal solution $\mathbf{x}^*$ to $(P^=)$, since the components of $\mathbf{x}^{\geq}$ defined in the statement of Theorem 2.3(ii) are such that some of them are the same as the corresponding components of $\mathbf{x}^*$, since some of the components of $\mathbf{x}^{\geq}$, namely those for $j \in J_a^\lambda$ with $a_j < \hat{x}_j$, are greater than the corresponding components $x_j^* = a_j$, $j \in J_a^\lambda$, of $\mathbf{x}^*$, and since $d_j > 0$, $j = 1,\ldots,n$, then obviously the inequality constraint (2.40) holds for $\mathbf{x}^{\geq}$. It is easy to check that other conditions (2.43), (2.37), (2.38) are also satisfied. Thus, $x_j^{\geq}$, $j = 1,\ldots,n$, defined above satisfy the KKT conditions for $(P^{\geq})$. Therefore $\mathbf{x}^{\geq}$ is the optimal solution to problem $(P^{\geq})$. □

According to Theorem 2.3, the optimal solution to problem $(P^{\geq})$ is obtained by using the optimal solution and optimal value of the Lagrange multiplier λ for problem $(P^=)$. That is why we suppose that $\sum_{j=1}^n d_j a_j \leq \alpha$ in addition to assumption (3.b) (see Step (1) of Algorithm 3 below) as we assumed this in assumption (2.b) for problem $(P^=)$.

3. The algorithms

3.1. Analysis of the optimal solution to problem $(P^{\leq})$. Before the formal statement of the algorithm for problem $(P^{\leq})$, we discuss some properties of the optimal solution to this problem, which turn out to be useful.

Using (2.10), (2.11), and (2.12), condition (2.6) can be written as follows:

$$\lambda\left(\sum_{j\in J_a^\lambda} d_j a_j + \sum_{j\in J_b^\lambda} d_j b_j + \sum_{j\in J^\lambda} d_j(\hat{x}_j - \lambda d_j) - \alpha\right) = 0, \quad \lambda \geq 0. \tag{3.1}$$

Since the optimal solution $\mathbf{x}^*$ to problem $(P^{\leq})$ obviously depends on λ, we consider components of $\mathbf{x}^*$ as functions of λ for different $\lambda \in \mathbb{R}^1_+$:

$$x_j^* = x_j(\lambda) = \begin{cases} a_j, & j \in J_a^\lambda, \\ b_j, & j \in J_b^\lambda, \\ \hat{x}_j - \lambda d_j, & j \in J^\lambda. \end{cases} \tag{3.2}$$

Functions $x_j(\lambda)$, $j = 1,\dots,n$, are piecewise linear, monotone nonincreasing, piecewise differentiable functions of λ with two breakpoints at $\lambda = (\hat{x}_j - a_j)/d_j$ and $\lambda = (\hat{x}_j - b_j)/d_j$.

Let

$$\delta(\lambda) \stackrel{\text{def}}{=} \sum_{j \in J_a^\lambda} d_j a_j + \sum_{j \in J_b^\lambda} d_j b_j + \sum_{j \in J^\lambda} d_j \hat{x}_j - \lambda \sum_{j \in J^\lambda} d_j^2 - \alpha. \tag{3.3}$$

If we differentiate (3.3) with respect to λ, we get

$$\delta'(\lambda) \equiv -\sum_{j \in J^\lambda} d_j^2 < 0, \tag{3.4}$$

when $J^\lambda \neq \varnothing$, and $\delta'(\lambda) = 0$ when $J^\lambda = \varnothing$. Hence $\delta(\lambda)$ is a *monotone nonincreasing function* of λ, $\lambda \in \mathbb{R}^1_+$, and $\max_{\lambda \ge 0} \delta(\lambda)$ is attained at the minimum admissible value of λ, that is, at $\lambda = 0$.

Case 1. If $\delta(0) > 0$, in order that (3.1) and (2.7) be satisfied, there exists some $\lambda^* > 0$ such that $\delta(\lambda^*) = 0$, that is,

$$\sum_{j=1}^n d_j x_j^* = \alpha, \tag{3.5}$$

which means that the inequality constraint (1.3) is satisfied with an equality for λ^* in this case.

Case 2. If $\delta(0) < 0$, then $\delta(\lambda) < 0$ for all $\lambda \ge 0$, and the maximum of $\delta(\lambda)$ with $\lambda \ge 0$ is $\delta(0) = \max_{\lambda \ge 0} \delta(\lambda)$ and it is attained at $\lambda = 0$ in this case. In order that (3.1) be satisfied, λ must be equal to 0. Therefore $x_j^* = \hat{x}_j$, $j \in J^{\lambda=0}$, according to (2.12).

Case 3. In the special case when $\delta(0) = 0$, the maximum $\delta(0) = \max_{\lambda \ge 0} \delta(\lambda)$ of $\delta(\lambda)$ is also attained at the minimum admissible value of λ, that is, for $\lambda = 0$, because $\delta(\lambda)$ is a monotone nonincreasing function in accordance with the above consideration.

As we have seen, for the optimal value of λ, we have $\lambda \ge 0$ in all possible cases, as the KKT condition (2.6) requires. We have shown that in Case 1 we need an algorithm for finding λ^* which satisfies the KKT conditions (2.3)–(2.9) but such that λ^* satisfies (2.7) with an equality. In order that this be fulfilled, the set (1.5)-(1.6) (i.e., feasible region of problem $(P^=)$) must be nonempty. That is why we have required $\alpha \le \sum_{j=1}^n d_j b_j$ in some cases in addition to the assumption $\sum_{j=1}^n d_j a_j \le \alpha$ (see assumption (1.b)). We have also used this in the proof of Theorem 2.1, part (ii), when $\lambda > 0$.

Using the equation $\delta(\lambda) = 0$, where $\delta(\lambda)$ is defined by (3.3), we are able to obtain a closed-form expression for λ

$$\lambda = \left(\sum_{j\in J^\lambda} d_j^2\right)^{-1} \left(\sum_{j\in J_a^\lambda} d_j a_j + \sum_{j\in J_b^\lambda} d_j b_j + \sum_{j\in J^\lambda} d_j \hat{x}_j - \alpha\right), \tag{3.6}$$

because $\delta'(\lambda) < 0$ according to (3.4) when $J^\lambda \neq \varnothing$ (it is important that $\delta'(\lambda) \neq 0$). This expression of λ is used in the algorithm suggested for problem $(P^\leq)$. It turns out that for our purposes, without loss of generality, we can assume that $\delta'(\lambda) \neq 0$, that is, $\delta(\lambda)$ depends on λ, which means that $J^\lambda \neq \varnothing$.

At iteration k of the implementation of the algorithms, denote by $\lambda^{(k)}$ the value of the Lagrange multiplier associated with constraint (1.3) (resp., (1.5), (1.7)), by $\alpha^{(k)}$ the right-hand side of (1.3) (resp., (1.5), (1.7)), and by $J^{(k)}$, $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ the current sets $J = \{1,\dots,n\}$, J_a^λ, J_b^λ, J^λ, respectively.

3.2. Algorithm 1. The following algorithm for solving problem $(P^\leq)$ is based on Theorem 2.1.

Algorithm 1 (for problem $(P^\leq)$).

(0) (Initialization). $J := \{1,\dots,n\}$, $k := 0$, $\alpha^{(0)} := \alpha$, $n^{(0)} := n$, $J^{(0)} := J$, $J_a^\lambda := \varnothing$, $J_b^\lambda := \varnothing$, initialize $\hat{x}_j$, $j \in J$. If $\sum_{j\in J} d_j a_j \leq \alpha$, go to (1) else go to (9).

(1) Construct the sets J_a^0, J_b^0, J^0 (for $\lambda = 0$). Calculate

$$\delta(0) := \sum_{j\in J_a^0} d_j a_j + \sum_{j\in J_b^0} d_j b_j + \sum_{j\in J^0} d_j \hat{x}_j - \alpha. \tag{3.7}$$

If $\delta(0) \leq 0$, then $\lambda := 0$, go to (8)
else if $\delta(0) > 0$, then
 if $\alpha \leq \sum_{j\in J} d_j b_j$, go to (2)
 else if $\alpha > \sum_{j\in J} d_j b_j$, go to (9) (there does not exist $\lambda^* > 0$ such that $\delta(\lambda^*) = 0$).

(2) $J^{\lambda(k)} := J^{(k)}$. Calculate $\lambda^{(k)}$ by using the explicit expression of λ (3.6). Go to (3).

(3) Construct the sets $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ through (2.10), (2.11), (2.12) (with $j \in J^{(k)}$ instead of $j \in J$) and find their cardinalities $|J_a^{\lambda(k)}|$, $|J_b^{\lambda(k)}|$, $|J^{\lambda(k)}|$, respectively. Go to (4).

(4) Calculate

$$\delta(\lambda^{(k)}) := \sum_{j\in J_a^{\lambda(k)}} d_j a_j + \sum_{j\in J_b^{\lambda(k)}} d_j b_j + \sum_{j\in J^{\lambda(k)}} d_j \hat{x}_j - \lambda^{(k)} \sum_{j\in J^{\lambda(k)}} d_j^2 - \alpha^{(k)}. \tag{3.8}$$

Go to (5).

(5) If $\delta(\lambda^{(k)}) = 0$ or $J^{\lambda(k)} = \varnothing$, then $\lambda := \lambda^{(k)}$, $J_a^\lambda := J_a^\lambda \cup J_a^{\lambda(k)}$, $J_b^\lambda := J_b^\lambda \cup J_b^{\lambda(k)}$, $J^\lambda := J^{\lambda(k)}$, go to (8)
else if $\delta(\lambda^{(k)}) > 0$, go to (6)
else if $\delta(\lambda^{(k)}) < 0$, go to (7).

(6) $x_j^* := a_j$ for $j \in J_a^{\lambda(k)}$, $\alpha^{(k+1)} := \alpha^{(k)} - \sum_{j\in J_a^{\lambda(k)}} d_j a_j$, $J^{(k+1)} := J^{(k)} \setminus J_a^{\lambda(k)}$, $n^{(k+1)} := n^{(k)} - |J_a^{\lambda(k)}|$, $J_a^{\lambda} := J_a^{\lambda} \cup J_a^{\lambda(k)}$, $k := k+1$. Go to (2).

(7) $x_j^* := b_j$ for $j \in J_b^{\lambda(k)}$, $\alpha^{(k+1)} := \alpha^{(k)} - \sum_{j\in J_b^{\lambda(k)}} d_j b_j$, $J^{(k+1)} := J^{(k)} \setminus J_b^{\lambda(k)}$, $n^{(k+1)} := n^{(k)} - |J_b^{\lambda(k)}|$, $J_b^{\lambda} := J_b^{\lambda} \cup J_b^{\lambda(k)}$, $k := k+1$. Go to (2).

(8) $x_j^* := a_j$ for $j \in J_a^{\lambda}$; $x_j^* := b_j$ for $j \in J_b^{\lambda}$; $x_j^* := \hat{x}_j - \lambda d_j$ for $j \in J^{\lambda}$. Go to (10).

(9) The problem has no optimal solution because $X = \varnothing$ or there does not exist a $\lambda^* > 0$ satisfying Theorem 2.1.

(10) End.

3.3. Convergence and complexity of Algorithm 1. The following theorem states convergence of Algorithm 1.

Theorem 3.1. *Let $\lambda^{(k)}$ be the sequence generated by Algorithm 1. Then*

(i) *if $\delta(\lambda^{(k)}) > 0$, then $\lambda^{(k)} \le \lambda^{(k+1)}$;*

(ii) *if $\delta(\lambda^{(k)}) < 0$, then $\lambda^{(k)} \ge \lambda^{(k+1)}$.*

Proof. Denote by $x_j^{(k)}$ the components of $\mathbf{x}^{(k)} = (x_j)_{j\in J^{(k)}}$ at iteration k of implementation of Algorithm 1.

Taking into consideration (3.4), Case 1, Case 2, Case 3, and Step (1) (the sign of $\delta(0)$) and Step (2) of Algorithm 1, it follows that $\lambda^{(k)} \ge 0$ for each k. Since $x_j^{(k)}$ are determined from (2.12), $x_j^{(k)} = \hat{x}_j - \lambda^{(k)} d_j$, $j \in J^{\lambda(k)}$, substituted in $\sum_{j\in J^{\lambda(k)}} d_j x_j^{(k)} = \alpha^{(k)}$ at Step (2) of Algorithm 1, and since $\lambda^{(k)} \ge 0$, $d_j > 0$, then $\hat{x}_j \ge x_j^{(k)}$.

(i) Let $\delta(\lambda^{(k)}) > 0$. Using Step (6) of Algorithm 1 (which is performed when $\delta(\lambda^{(k)}) > 0$), we get

$$\sum_{j\in J^{\lambda(k+1)}} d_j x_j^{(k)} \equiv \sum_{j\in J^{(k+1)}} d_j x_j^{(k)} = \sum_{j\in J^{(k)}\setminus J_a^{\lambda(k)}} d_j x_j^{(k)} = \alpha^{(k)} - \sum_{j\in J_a^{\lambda(k)}} d_j x_j^{(k)}. \tag{3.9}$$

Let $j \in J_a^{\lambda(k)}$. According to definition (2.10) of $J_a^{\lambda(k)}$ we have

$$\frac{\hat{x}_j - a_j}{d_j} \le \lambda^{(k)} = \frac{\hat{x}_j - x_j^{(k)}}{d_j}. \tag{3.10}$$

Multiplying this inequality by $-d_j < 0$, we obtain $a_j - \hat{x}_j \ge x_j^{(k)} - \hat{x}_j$. Therefore $a_j \ge x_j^{(k)}$, $j \in J_a^{\lambda(k)}$.

Using that $d_j > 0$ and Step (6), from (3.9) we get

$$\begin{aligned}\sum_{j\in J^{\lambda(k+1)}} d_j x_j^{(k)} &= \alpha^{(k)} - \sum_{j\in J_a^{\lambda(k)}} d_j x_j^{(k)} \ge \alpha^{(k)} - \sum_{j\in J_a^{\lambda(k)}} d_j a_j \\ &= \alpha^{(k+1)} = \sum_{j\in J^{\lambda(k+1)}} d_j x_j^{(k+1)}.\end{aligned} \tag{3.11}$$

Since $d_j > 0$, $j = 1,\dots,n$, there exists at least one $j_0 \in J^{\lambda(k+1)}$ such that $x_{j_0}^{(k)} \geq x_{j_0}^{(k+1)}$. Then

$$\lambda^{(k)} = \frac{\hat{x}_{j_0} - x_{j_0}^{(k)}}{d_{j_0}} \leq \frac{\hat{x}_{j_0} - x_{j_0}^{(k+1)}}{d_{j_0}} = \lambda^{(k+1)}. \tag{3.12}$$

We have used that the relationship between $\lambda^{(k)}$ and $x_j^{(k)}$ is given by (2.12) for $j \in J^{\lambda(k)}$ according to Step (2) of Algorithm 1 and that $\hat{x}_j \geq x_j^{(k)}$, $j \in J^{\lambda(k)}$, according to (2.12) with $\lambda^{(k)} \geq 0$ and $d_j > 0$.

The proof of part (ii) is omitted because it is similar to that of part (i). □

Consider the feasibility of $\mathbf{x}^* = (x_j^*)_{j\in J}$, generated by Algorithm 1.

Components $x_j^* = a_j$, $j \in J_a^\lambda$, and $x_j^* = b_j$, $j \in J_b^\lambda$, obviously satisfy (1.4). It follows from

$$\frac{\hat{x}_j - b_j}{d_j} < \lambda \equiv \frac{\hat{x}_j - x_j^*}{d_j} < \frac{\hat{x}_j - a_j}{d_j}, \quad j \in J^\lambda, \tag{3.13}$$

and $d_j > 0$ that $a_j - \hat{x}_j < x_j^* - \hat{x}_j < b_j - \hat{x}_j$, $j \in J^\lambda$. Therefore $a_j < x_j^* < b_j$ for $j \in J^\lambda$. Hence all x_j^*, $j = 1,\dots,n$, satisfy (1.4).

We have proved that if $\delta(\lambda)_{|\lambda=0} \geq 0$ and $X \neq \emptyset$, where X is defined by (1.3)-(1.4), then there exists a $\lambda^* \geq 0$ such that $\delta(\lambda^*) = 0$. Since at Step (2) we determine $\lambda^{(k)}$ from the equality $\sum_{j\in J^{\lambda(k)}} d_j x_j^{(k)} = \alpha^{(k)}$ for each k, then (1.3) is satisfied with an equality in this case. Otherwise, if $\delta(0) < 0$, then we set $\lambda = 0$ (Step (1)) and we have $\sum_{j\in J} d_j x_j(0) - \alpha \equiv \delta(0) < 0$, that is, (1.3) is satisfied as a strict inequality in this case.

Therefore Algorithm 1 generates $\mathbf{x}^*$ which is feasible for problem $(P^{\leq})$.

Remark 3.2. Theorem 3.1, definitions of J_a^λ (2.10), J_b^λ (2.11), and J^λ (2.12), and Steps (6), (7), (8) of Algorithm 1 allow us to assert that $J_a^{\lambda(k)} \subseteq J_a^{\lambda(k+1)}$, $J_b^{\lambda(k)} \subseteq J_b^{\lambda(k+1)}$, and $J^{\lambda(k)} \supseteq J^{\lambda(k+1)}$. This means that if j belongs to current set $J_a^{\lambda(k)}$, then j belongs to the next index set $J_a^{\lambda(k+1)}$ and, therefore, to the optimal index set J_a^λ, the same holds true about the sets $J_b^{\lambda(k)}$ and J_b^λ. Therefore $\lambda^{(k)}$ "converges" to the optimal λ of Theorem 2.1 and $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ "converge" to the optimal index sets J_a^λ, J_b^λ, J^λ, respectively. This means that calculation of λ, operations $x_j^* := a_j$, $j \in J_a^{\lambda(k)}$ (Step (6)), $x_j^* := b_j$, $j \in J_b^{\lambda(k)}$ (Step (7)), and the construction of J_a^λ, J_b^λ, J^λ are in accordance with Theorem 2.1.

At each iteration, Algorithm 1 determines the value of at least one variable (Steps (6), (7), (8)), and at each iteration, we solve a problem of the form $(P^{\leq})$ but of *less* dimension (Steps (2)–(7)). Therefore Algorithm 1 is finite and it converges with at most $n = |J|$ iterations, that is, the iteration complexity of Algorithm 1 is $\mathcal{O}(n)$.

Step (0) takes time $\mathcal{O}(n)$. Step (1) (construction of sets J_a^0, J_b^0, J^0, calculation of $\delta(0)$, and checking whether X is empty) also takes time $\mathcal{O}(n)$. The calculation of $\lambda^{(k)}$ requires constant time (Step (2)). Step (3) takes $\mathcal{O}(n)$ time because of the construction of $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$. Step (4) also requires $\mathcal{O}(n)$ time and Step (5) requires constant time. Each of Steps (6), (7), and (8) takes time which is bounded by $\mathcal{O}(n)$: at these steps we assign the final value to some of x_j, and since the number of all x_j's is n, then Steps (6), (7), and (8) take time $\mathbb{O}(n)$. Hence the algorithm has $\mathbb{O}(n^2)$ running time and it belongs to the class of strongly polynomially bounded algorithms.

As the computational experiments show, the number of iterations of the algorithm performance is not only at most n but it is much, much less than n for large n. In fact, this number does not depend on n but only on the three index sets defined by (2.10), (2.11), (2.12). In practice, Algorithm 1 has $\mathbb{O}(n)$ running time.

3.4. Algorithm 2 (for problem ($P^=$)) and its convergence. After analysis of the optimal solution to problem ($P^=$), similar to that to problem ($P^\leq$), we suggest the following algorithm for solving problem ($P^=$).

Algorithm 2 (for problem ($P^=$)).

(1) (Initialization). $J := \{1,\dots,n\}$, $k := 0$, $\alpha^{(0)} := \alpha$, $n^{(0)} := n$, $J^{(0)} := J$, $J_a^\lambda := \varnothing$, $J_b^\lambda := \varnothing$, initialize $\hat{x}_j$, $j \in J$. If $\sum_{j\in J} d_j a_j \leq \alpha \leq \sum_{j\in J} d_j b_j$, go to (2), else go to (9).

(2) $J^{\lambda(k)} := J^{(k)}$. Calculate $\lambda^{(k)}$ by using the explicit expression of λ. Go to (3).

(3) Construct the sets $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ through (2.30), (2.31), (2.32) (with $j \in J^{(k)}$ instead of $j \in J$) and find their cardinalities $|J_a^{\lambda(k)}|$, $|J_b^{\lambda(k)}|$, $|J^{\lambda(k)}|$. Go to (4).

Steps (4)–(8) are the same as Steps (4)–(8) of Algorithm 1, respectively.

(9) Problem ($P^=$) has no optimal solution because the feasible set X (1.5)-(1.6) is empty.

(10) End.

A theorem analogous to Theorem 3.1 holds for Algorithm 2 which guarantees the "convergence" of $\lambda^{(k)}$, $J^{\lambda(k)}$, $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$ to the optimal λ, J^λ, J_a^λ, J_b^λ, respectively.

Theorem 3.3. *Let $\lambda^{(k)}$ be the sequence generated by Algorithm 2. Then*

(i) *if $\delta(\lambda^{(k)}) > 0$, then $\lambda^{(k)} \leq \lambda^{(k+1)}$;*

(ii) *if $\delta(\lambda^{(k)}) < 0$, then $\lambda^{(k)} \geq \lambda^{(k+1)}$.*

The proof of Theorem 3.3 is omitted because it is similar to that of Theorem 3.1.

It can be proved that Algorithm 2 has $\mathbb{O}(n^2)$ running time, and point $\mathbf{x}^* = (x_1^*,\dots,x_n^*)$ generated by this algorithm is feasible for problem ($P^=$), which is an assumption of Theorem 2.2.

The following algorithm for solving problem ($P^\geq$) is based on Theorem 2.3 and Algorithm 2.

3.5. Algorithm 3 (for problem ($P^\geq$))

Algorithm 3.

(1) (Initialization). $J := \{1,\dots,n\}$, $k := 0$, $J^{(0)} := J$, $\alpha^{(0)} := \alpha$, $n^{(0)} := n$, $J_a^\lambda := \varnothing$ $J_b^\lambda := \varnothing$, initialize $\hat{x}_j$, $j \in J$. If $\sum_{j\in J} d_j a_j \leq \alpha \leq \sum_{j\in J} d_j b_j$, then go to (2), else go to (9).

Steps (2)–(7) are the same as Steps (2)–(7) of Algorithm 2, respectively.

(8) If $\lambda \leq 0$, then $x_j^\geq := a_j$ for $j \in J_a^\lambda$, $x_j^\geq := b_j$ for $j \in J_b^\lambda$, $x_j^\geq := \hat{x}_j - \lambda d_j$ for $j \in J^\lambda$, go to (10)

else if $\lambda > 0$, then

$x_j^\geq := b_j$ for $j \in J_b^\lambda$,

$x_j^{\geq} := \min\{b_j, \hat{x}_j\}$ for $j \in J^\lambda$,
if $j \in J_a^\lambda$ and $a_j < \hat{x}_j$, then $x_j^{\geq} := \min\{b_j, \hat{x}_j\}$
else if $j \in J_a^\lambda$ and $a_j \geq \hat{x}_j$, then $x_j^{\geq} := a_j$;
go to (10).

(9) Problem $(P^{\geq})$ has no optimal solution because $X = \varnothing$, where X is defined by (1.7)-(1.8), or there do not exist $x_j^* \in [a_j, b_j]$, $j \in J$, such that $\sum_{j\in J} d_j x_j^* = \alpha$.

(10) End.

Since Algorithm 3 is based on Theorem 2.3 and Algorithm 2 and since the "iterative" steps (2)–(7) of Algorithm 2 and Algorithm 3 are the same, then "convergence" of Algorithm 3 follows from Theorem 3.3 as well. Because of the same reason, computational complexity of Algorithm 3 is the same as that of Algorithm 2.

3.6. Commentary. Methods proposed for solving problems $(P^{\leq})$, $(P^{=})$, and $(P^{\geq})$ are *first-order methods* because they use values of the first derivatives (see (2.3)) of the objective function $c(\mathbf{x}) \equiv \sum_{j=1}^n c_j(x_j) \equiv (1/2)\sum_{j=1}^n (x_j - \hat{x}_j)^2$. Also, they are *dual variables saddle point methods* because they are based on "convergence" with respect to the Lagrange multiplier (dual variable) λ associated with the single constraint (1.3) (resp., constraint (1.5) or (1.7)). Moreover, they are *exact* methods because there are only round-off errors and there is no error of the methods.

Methods suggested in this paper, due to specificity of problems solved, are less restrictive than other methods for solving general convex quadratic programming problems, such as active set methods and gradient projection methods, with respect to dimension (number of variables) of the problem, convergence conditions, subproblems to be solved at each iteration, and so forth.

4. Extensions

4.1. Theoretical aspects. If it is allowed that $d_j = 0$ for some j in problems $(P^{\leq})$, $(P^{=})$, and $(P^{\geq})$, then, for such indices j, we cannot construct the expressions $(\hat{x}_j - a_j)/d_j$ and $(\hat{x}_j - b_j)/d_j$ by means of which we define sets $J_a^\lambda, J_b^\lambda, J^\lambda$ for the corresponding problem. In such cases, x_j's are not involved in (1.3) (resp., in (1.5) or in (1.7)) for such indices j. It turns out that we can cope with this difficulty and solve problems $(P^{\leq})$, $(P^{=})$, $(P^{\geq})$ with $d_j = 0$ for some j's.

Denote

$$J = \{1, \dots, n\}, \qquad Z0 = \{j \in J : d_j = 0\}. \tag{4.1}$$

Here "0" means the "computer zero." In particular, when $J = Z0$ and $\alpha = 0$, X is defined only by (1.4) (resp., by (1.6), by (1.8)).

Theorem 4.1 (characterization of the optimal solution to problem $(P^{\leq})$: an extended version). *Problem $(P^{\leq})$ can be decomposed into two subproblems: $(P1^{\leq})$ for $j \in Z0$ and $(P2^{\leq})$ for $j \in J \setminus Z0$.*

The optimal solution to $(P1^{\le})$ *is*

$$x_j^* = \begin{cases} a_j, & j \in Z0,\ \hat{x}_j \le a_j, \\ b_j, & j \in Z0,\ \hat{x}_j \ge b_j, \\ \hat{x}_j, & j \in Z0,\ a_j < \hat{x}_j < b_j, \end{cases} \tag{4.2}$$

that is, the subproblem $(P1^{\le})$ *itself is decomposed into* $n_0 \equiv |Z0|$ *independent problems. The optimal solution to* $(P2^{\le})$ *is given by (2.10), (2.11), (2.12) with* $J := J \setminus Z0$.

Proof

Necessity. Let $\mathbf{x}^* = (x_j^*)_{j\in J}$ be the optimal solution to $(P^{\le})$.

(1) Let $j \in Z0$, that is, $d_j = 0$. The KKT conditions are

$$x_j^* - \hat{x}_j - u_j + v_j = 0, \quad j \in Z0 \quad \text{from (2.3) and (2.4)–(2.9)}. \tag{4.3}$$

(a) If $x_j^* = a_j$, then $u_j \ge 0$, $v_j = 0$. It follows from (4.3) that $x_j^* - \hat{x}_j = u_j \ge 0$; that is, $x_j^* \equiv a_j \ge \hat{x}_j$.
(b) If $x_j^* = b_j$, then $u_j = 0$, $v_j \ge 0$. Therefore $x_j^* - \hat{x}_j = -v_j \le 0$; that is, $x_j^* \equiv b_j \le \hat{x}_j$.
(c) If $a_j < x_j^* < b_j$, then $u_j = v_j = 0$. Therefore $x_j^* - \hat{x}_j = 0$; that is, $x_j^* = \hat{x}_j$.

(2) Components of the optimal solution to $(P2^{\le})$ are obtained by using the same approach as that of the proof of Theorem 2.1(i) but with the reduced index set $J := J \setminus Z0$.

Sufficiency. Conversely, let $\mathbf{x}^* \in X$ and let the components of $\mathbf{x}^*$ satisfy (4.2) for $j \in Z0$, and (2.10), (2.11), (2.12) with $J := J \setminus Z0$. Set

$$\begin{aligned} &u_j = v_j = 0 \quad \text{for } a_j < x_j^* < b_j,\ j \in Z0, \\ &u_j = a_j - \hat{x}_j, \quad v_j = 0 \quad \text{for } x_j^* = a_j,\ j \in Z0, \\ &u_j = 0, \quad v_j = \hat{x}_j - b_j \quad \text{for } x_j^* = b_j,\ j \in Z0. \end{aligned} \tag{4.4}$$

If $\lambda > 0$, set

$$\begin{aligned} &\lambda = \frac{\hat{x}_j - x_j^*}{d_j} = \lambda(\mathbf{x}^*)\ (>0) \quad \text{from (2.12)}, \\ &u_j = v_j = 0 \quad \text{for } a_j < x_j^* < b_j,\ j \in J \setminus Z0, \\ &u_j = a_j - \hat{x}_j + \lambda d_j\ (\ge 0), \quad v_j = 0 \quad \text{for } x_j^* = a_j,\ j \in J \setminus Z0, \\ &u_j = 0, \quad v_j = \hat{x}_j - b_j - \lambda d_j\ (\ge 0) \quad \text{for } x_j^* = b_j,\ j \in J \setminus Z0. \end{aligned} \tag{4.5}$$

If $\lambda = 0$, set

$$\begin{aligned} &u_j = v_j = 0 \quad \text{for } a_j < x_j^* < b_j,\ j \in J \setminus Z0, \\ &u_j = a_j - \hat{x}_j\ (\ge 0), \quad v_j = 0 \quad \text{for } x_j^* = a_j,\ j \in J \setminus Z0, \\ &u_j = 0, \quad v_j = \hat{x}_j - b_j\ (\ge 0) \quad \text{for } x_j^* = b_j,\ j \in J \setminus Z0. \end{aligned} \tag{4.6}$$

It can be verified that $\mathbf{x}^*$, λ, u_j, v_j, $j \in J$, satisfy the KKT conditions (4.3), (2.4)–(2.9). Then $\mathbf{x}^*$ with components (4.2), for $j \in Z0$, and (2.10), (2.11), (2.12), with $J := J \setminus Z0$, is the optimal solution to problem $(P^{\leq}) = (P1^{\leq}) \cup (P2^{\leq})$. □

An analogous result holds for problem $(P^{=})$.

THEOREM 4.2 (characterization of the optimal solution to problem $(P^{=})$: an extended version). *Problem $(P^{=})$ can be decomposed into two subproblems: $(P1^{=})$ for $j \in Z0$ and $(P2^{=})$ for $j \in J \setminus Z0$.*

The optimal solution to $(P1^{=})$ is also given by (4.2). The optimal solution to $(P2^{=})$ is given by (2.30), (2.31), (2.32) with $J := J \setminus Z0$.

The proof of Theorem 4.2 is omitted because it repeats in part the proofs of Theorems 2.1 and 4.1.

Thus, with the use of Theorems 4.1 and 4.2, we can express components of the optimal solutions to problems $(P^{\leq})$, $(P^{=})$, and $(P^{\geq})$ *without* the necessity of constructing the expressions $(\hat{x}_j - a_j)/d_j$ and $(\hat{x}_j - b_j)/d_j$ with $d_j = 0$.

Since Theorem 2.3 and Algorithm 3 are based on the sets of indices J_a^{λ}, J_b^{λ}, J^{λ} of problem $(P^{=})$, Theorem 4.2 solves the problem of decomposition of problem $(P^{\geq})$ as well.

4.2. Computational aspects. Algorithms 1, 2, and 3 are also applicable in cases when $a_j = -\infty$ for some j, $1 \leq j \leq n$ and/or $b_j = \infty$ for some j, $1 \leq j \leq n$. However, if we use the computer values of $-\infty$ and $+\infty$ at the first step of the algorithms to check whether the corresponding feasible region is empty or nonempty and at Step (3) in the expressions $(\hat{x}_j - x_j)/d_j$ with $x_j = -\infty$ and/or $x_j = +\infty$, by means of which we construct sets J_a^{λ}, J_b^{λ}, J^{λ}, this could sometimes lead to arithmetic overflow. If we use other values of $-\infty$ and $+\infty$ with smaller absolute values than those of the computer values of $-\infty$ and $+\infty$, it would lead to inconvenience and dependence on the data of the particular problems. To avoid these difficulties and to take into account the problems considered above, it is convenient to do the following.

Construct the sets of indices

$$\begin{aligned} SVN &= \{j \in J \setminus Z0 : a_j > -\infty,\ b_j < +\infty\}, \\ SV1 &= \{j \in J \setminus Z0 : a_j > -\infty,\ b_j = +\infty\}, \\ SV2 &= \{j \in J \setminus Z0 : a_j = -\infty,\ b_j < +\infty\}, \\ SV &= \{j \in J \setminus Z0 : a_j = -\infty,\ b_j = +\infty\}. \end{aligned} \tag{4.7}$$

It is obvious that $Z0 \cup SV \cup SV1 \cup SV2 \cup SVN = J$, that is, the set $J \setminus Z0$ is partitioned into the four subsets SVN, $SV1$, $SV2$, SV defined above.

When programming the algorithms, we use computer values of $-\infty$ and $+\infty$ for constructing the sets SVN, $SV1$, $SV2$, SV.

In order to construct the sets J_a^{λ}, J_b^{λ}, J^{λ} without the necessity of calculating the values $(\hat{x}_j - x_j)/d_j$ with $x_j = -\infty$ or $+\infty$, except for the sets J, $Z0$, SV, $SV1$, $SV2$, SVN, we need some subsidiary sets defined as follows.

For SVN,

$$J^{\lambda SVN} = \left\{ j \in SVN : \frac{\hat{x}_j - b_j}{d_j} < \lambda < \frac{\hat{x}_j - a_j}{d_j} \right\},$$
$$J_a^{\lambda SVN} = \left\{ j \in SVN : \lambda \geq \frac{\hat{x}_j - a_j}{d_j} \right\}, \tag{4.8}$$
$$J_b^{\lambda SVN} = \left\{ j \in SVN : \lambda \leq \frac{\hat{x}_j - b_j}{d_j} \right\};$$

for $SV1$,

$$J^{\lambda SV1} = \left\{ j \in SV1 : \lambda < \frac{\hat{x}_j - a_j}{d_j} \right\},$$
$$J_a^{\lambda SV1} = \left\{ j \in SV1 : \lambda \geq \frac{\hat{x}_j - a_j}{d_j} \right\}; \tag{4.9}$$

for $SV2$,

$$J^{\lambda SV2} = \left\{ j \in SV2 : \lambda > \frac{\hat{x}_j - b_j}{d_j} \right\},$$
$$J_b^{\lambda SV2} = \left\{ j \in SV2 : \lambda \leq \frac{\hat{x}_j - b_j}{d_j} \right\}; \tag{4.10}$$

for SV,

$$J^{\lambda SV} = SV. \tag{4.11}$$

Then

$$J^{\lambda} := J^{\lambda SVN} \cup J^{\lambda SV1} \cup J^{\lambda SV2} \cup J^{\lambda SV},$$
$$J_a^{\lambda} := J_a^{\lambda SVN} \cup J_a^{\lambda SV1}, \tag{4.12}$$
$$J_b^{\lambda} := J_b^{\lambda SVN} \cup J_b^{\lambda SV2}.$$

We use the sets J^{λ}, J_a^{λ}, J_b^{λ} (4.12) as the corresponding sets with the same names in Algorithms 1, 2, and 3.

With the use of results of this section, Steps (0), (1), and (3) of Algorithm 1 can be modified, respectively as follows.

About Algorithm 1.

Step $(0)^1$. (Initialization). $J := \{1, \dots, n\}$, $k := 0$, $\alpha^{(0)} := \alpha$, $n^{(0)} := n$,
$J^{(0)} := J$, $J_a^{\lambda} := \emptyset$, $J_b^{\lambda} := \emptyset$, initialize $\hat{x}_j$, $j \in J$.
Construct the set $Z0$. If $j \in Z0$, then
if $\hat{x}_j \leq a_j$, then $x_j^* := a_j$,
else if $\hat{x}_j \geq b_j$, then $x_j^* := b_j$,
else if $a_j < \hat{x}_j < b_j$, then $x_j^* := \hat{x}_j$.

If $J = Z0$ and $\alpha = 0$, go to Step (10)
 else if $J = Z0$ and $\alpha \neq 0$, go to Step (9).
Set $J := J \setminus Z0$, $J^{(0)} := J$, $n^{(0)} := n - |Z0|$.
Construct the sets SVN, $SV1$, $SV2$, SV.
If $SVN \cup SV1 = J$, then
 if $\sum_{j\in J} d_j a_j \leq \alpha$, go to Step (1)
 else go to Step (9) (feasible region X is empty)
else if $SV2 \cup SV \neq \varnothing$, then
 if $SVN \cup SV1 \neq \varnothing$, then
  if $\sum_{j\in SVN\cup SV1} d_j a_j \leq \alpha$, go to Step (1)
  else go to Step (9) (feasible region is empty)
 else go to Step (1) (feasible region is always nonempty).

Step (1)[1]. Construct the sets $J^{0SVN}, J_a^{0SVN}, J_b^{0SVN}, J^{0SV1}, J_a^{0SV1}, J^{0SV2}, J_b^{0SV2}, J^{0SV}$ (for $\lambda = 0$). Construct the sets J_a^0, J_b^0, J^0 through (4.12). Calculate

$$\delta(0) := \sum_{j\in J_a^0} d_j a_j + \sum_{j\in J_b^0} d_j b_j + \sum_{j\in J^0} d_j \hat{x}_j - \alpha. \tag{4.13}$$

If $\delta(0) \leq 0$, then $\lambda := 0$, go to Step (8)
else if $\delta(0) > 0$, then

 if $SV2 \cup SVN = J$, then
  if $\alpha \leq \sum_{j\in J} d_j b_j$, go to Step (2)
  else go to Step (9) (there does not exist a $\lambda^* > 0$ such that $\delta(\lambda^*) = 0$)
 else if $SV1 \cup SV \neq \varnothing$, go to Step (2) (there exists a $\lambda^* > 0$ such that $\delta(\lambda^*) = 0$).

Step (3)[1]. Construct the sets $J^{\lambda SVN}, J_a^{\lambda SVN}, J_b^{\lambda SVN}, J^{\lambda SV1}, J_a^{\lambda SV1}, J^{\lambda SV2}, J_b^{\lambda SV2}, J^{\lambda SV}$ (with $J^{(k)}$ instead of J).

Construct the sets $J_a^{\lambda(k)}, J_b^{\lambda(k)}, J^{\lambda(k)}$ by using (4.12) and find their cardinalities $|J_a^{\lambda(k)}|$, $|J_b^{\lambda(k)}|$, $|J^{\lambda(k)}|$, respectively. Go to Step (4).

Similarly, we can modify Steps (1) and (3) of both Algorithm 2 and Algorithm 3.

Modifications of the algorithms connected with theoretical and computational aspects do not influence their computational complexity, which has been discussed in Section 3, because these modifications do not affect the "iterative" steps of algorithms.

5. Computational experiments

In this section, we present results (Table 5.1) of some numerical experiments obtained by applying algorithms suggested in this paper to problems under consideration. The computations were performed on an Intel Pentium II Celeron Processor 466 MHz/128 MB SDRAM IBM PC compatible. Each problem was run 30 times. Coefficients $d_j > 0$, $j = 1,\ldots,n$, and data $\hat{\mathbf{x}} = (\hat{x}_1,\ldots,\hat{x}_n)$ were randomly generated according to uniform and Gaussian (normal) distribution. It is also possible to use values generated according to

Table 5.1

Problem	$(P^{\leq})$			
Number of variables	$n = 1200$	$n = 1500$	$n = 5000$	$n = 10\,000$
Average number of iterations	2.09	2.11	3.58	5.53
Average run time (in seconds)	0.0001	0.00011	0.00026	0.00055
Problem	$(P^{=})$			
Number of variables	$n = 1200$	$n = 1500$	$n = 5000$	$n = 10\,000$
Average number of iterations	2.07	2.10	3.67	5.64
Average run time (in seconds)	0.00009	0.0001	0.00032	0.00057
Problem	$(P^{\geq})$			
Number of variables	$n = 1200$	$n = 1500$	$n = 5000$	$n = 10\,000$
Average number of iterations	2.09	2.12	3.78	5.57
Average run time (in seconds)	0.000101	0.00011	0.00037	0.00061

other distributions as well as data of real problems. Efficiency of Algorithms 1, 2, and 3 does not depend on the data.

When $n < 1\,200$, the run time of the algorithms is so small that the timer does not recognize the corresponding value from its computer zero. In such cases the timer displays 0 seconds.

Effectiveness of algorithms for problems $(P^{\leq})$, $(P^{=})$, and $(P^{\geq})$ has been tested by many other examples. As we can observe, the (average) number of iterations is much less than the number of variables n for large n.

We provide below the solution of two simple particular problems of the form $(P^{=})$ obtained by using the approach suggested in this paper.

Example 5.1. Find the projection of $\hat{\mathbf{x}} = (55, 12, 15, 85, 30)$ on a set defined as follows:

$$\begin{gathered} x_1 + x_2 + 2x_3 + 3x_4 + x_5 = 200, \\ 0 \leq x_1 \leq 50, \\ 0 \leq x_2 \leq 7, \\ 0 \leq x_3 \leq 7, \\ 0 \leq x_4 \leq 80, \\ 0 \leq x_5 \leq 25. \end{gathered} \tag{5.1}$$

The projection of $\hat{\mathbf{x}}$ on this set is

$$\mathbf{x}^* = (42.27, 0.0, 0.0, 46.81, 17.27). \tag{5.2}$$

The Euclidean distance between $\hat{\mathbf{x}}$ and $\mathbf{x}^*$ is

$$\operatorname{dist}(\mathbf{x}^*, \hat{\mathbf{x}}) = \|\mathbf{x}^* - \hat{\mathbf{x}}\| = \left(\sum_{j=1}^{5} (x_j^* - \hat{x}_j)^2 \right)^{1/2} = 46.37691. \tag{5.3}$$

Example 5.2. Find the projection of $\hat{\mathbf{x}} = (2,3,1,2)$ on a feasible region defined as follows:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 1, \\ 0 \le x_j \le 1, \quad j &= 1,2,3,4. \end{aligned} \tag{5.4}$$

The projection of $\hat{\mathbf{x}}$ on this region is

$$\mathbf{x}^* = (0,1,0,0). \tag{5.5}$$

The Euclidean distance between $\hat{\mathbf{x}}$ and $\mathbf{x}^*$ is

$$\operatorname{dist}(\mathbf{x}^*, \hat{\mathbf{x}}) = \|\mathbf{x}^* - \hat{\mathbf{x}}\| = \left(\sum_{j=1}^{4} (x_j^* - \hat{x}_j)^2 \right)^{1/2} = 3.60555. \tag{5.6}$$

6. Concluding remarks

The approach proposed in this paper could be modified for strictly convex quadratic objective functions and, more generally, for strictly convex separable objective functions.

Acknowledgment

The author is grateful to the anonymous referees for their comments and suggestions.

References

[1] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim. **20** (1982), no. 2, 221–246.

[2] G. R. Bitran and A. C. Hax, *Disaggregation and resource allocation using convex knapsack problems with bounded variables*, Management Sci. **27** (1981), no. 4, 431–441.

[3] P. Brucker, *An $O(n)$ algorithm for quadratic knapsack problems*, Oper. Res. Lett. **3** (1984), no. 3, 163–166.

[4] A. Charnes and W. W. Cooper, *The theory of search: optimum distribution of search effort*, Management Sci. **5** (1958), 44–50.

[5] S. M. Grzegórski, *Orthogonal projections on convex sets for Newton-like methods*, SIAM J. Numer. Anal. **22** (1985), no. 6, 1208–1219.

[6] M. Held, P. Wolfe, and H. P. Crowder, *Validation of subgradient optimization*, Math. Programming **6** (1974), 62–88.

[7] N. Katoh, T. Ibaraki, and H. Mine, *A polynomial time algorithm for the resource allocation problem with a convex objective function*, J. Oper. Res. Soc. **30** (1979), no. 5, 449–455.

[8] H. Luss and S. K. Gupta, *Allocation of effort resources among competing activities*, Operations Res. **23** (1975), no. 2, 360–366.

[9] C. Michelot, *A finite algorithm for finding the projection of a point onto the canonical simplex of* $\mathbf{R}^n$, J. Optim. Theory Appl. **50** (1986), no. 1, 195–200.

[10] R. T. Rockafellar and R. J.-B. Wets, *A note about projections in the implementation of stochastic quasigradient methods*, Numerical Techniques for Stochastic Optimization (Yu. Ermoliev and R. J.-B. Wets, eds.), Springer Ser. Comput. Math., vol. 10, Springer, Berlin, 1988, pp. 385–392.

[11] S. M. Stefanov, *On the implementation of stochastic quasigradient methods to some facility location problems*, Yugosl. J. Oper. Res. **10** (2000), no. 2, 235–256.

[12] ______, *Convex separable minimization subject to bounded variables*, Comput. Optim. Appl. **18** (2001), no. 1, 27–48.

[13] ______, *Separable Programming. Theory and Methods*, Applied Optimization, vol. 53, Kluwer Academic Publishers, Dordrecht, 2001.

[14] ______, *Convex separable minimization problems with a linear constraint and bounds on the variables*, Applications of Mathematics in Engineering and Economics (Sozopol, 2001), Heron Press, Sofia, 2002, pp. 392–402.

[15] P. H. Zipkin, *Simple ranking methods for allocation of one resource*, Management Sci. **26** (1980), no. 1, 34–43.

Stefan M. Stefanov: Department of Mathematics, South-West University "Neofit Rilski," 2700 Blagoevgrad, Bulgaria

E-mail address: stefm@aix.swu.bg