

CONVEX SEPARABLE MINIMIZATION PROBLEMS WITH A LINEAR CONSTRAINT AND BOUNDED VARIABLES

STEFAN M. STEFANOV

Received 30 June 2004 and in revised form 19 April 2005

Consider the minimization problem with a convex separable objective function over a feasible region defined by linear equality constraint(s)/linear inequality constraint of the form “greater than or equal to” and bounds on the variables. A necessary and sufficient condition and a sufficient condition are proved for a feasible solution to be an optimal solution to these two problems, respectively. Iterative algorithms of polynomial complexity for solving such problems are suggested and convergence of these algorithms is proved. Some convex functions, important for problems under consideration, as well as computational results are presented.

1. Introduction

In many cases, we have to minimize a convex separable function over a region defined by a linear equality or inequality “ \geq ” constraint with positive coefficients, and two-sided bounds on the variables.

Such problems and problems related to them arise, for example, in production planning and scheduling [2] and Problem 4, Section 5, in allocation of resources [2, 31] and Problem 1, Section 5, in allocation of effort resources among competing activities [16] and Problems 3, 5, and 6, Section 5, in the theory of search [6], in subgradient optimization [11], in facility location [24], and in the implementation of projection methods when the feasible region is of the considered form [28] and Problem 2, Section 5, and so forth.

The problems under consideration can mathematically be formulated as follows:

$$\min \left\{ c(\mathbf{x}) = \sum_{j \in J} c_j(x_j) \right\} \quad (1.1)$$

subject to

$$\sum_{j \in J} d_j x_j \begin{cases} = \\ \geq \end{cases} \alpha, \quad (1.2)$$

$$a_j \leq x_j \leq b_j, \quad j \in J, \quad (1.3)$$

where $c_j(x_j)$ are twice differentiable convex functions, defined on the open convex sets X_j in \mathbb{R} , $j \in J$, respectively; $d_j > 0$ for every $j \in J$, $\mathbf{x} = (x_j)_{j \in J}$, and $J \equiv \{1, \dots, n\}$.

Denote this problem by $(C^=)$ in the first case (problem (1.1), (1.2), and (1.3) with equality constraint (1.2)), and by $(C^≥)$ in the second case (problem (1.1), (1.2), and (1.3) with inequality “ $≥$ ” constraint (1.2)). Denote by $X^=$ and $X^≥$ the feasible region (1.2)-(1.3) in the two cases, respectively. A constraint like (1.2) is known as the *knapsack constraint*.

Also, a generalization of problem $(C^=)$, denoted by $(C_m^=)$, is considered in which (1.2) is defined by m linear equality constraints.

The feasible region (1.2)-(1.3) is an intersection of the hyperplane(s)/halfspace (1.2) and the box (1.3) of dimension $|J| = n$. Therefore, (1.2)-(1.3) is a convex set. Therefore, $(C^=)$, $(C_m^=)$, and $(C^≥)$ are convex programming problems.

Problems like $(C^=)$ and $(C^≥)$ 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, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]. The solution of knapsack problems with arbitrary convex or concave objective functions is studied in [2, 16, 20, 25, 31], and so forth. Algorithms for solving convex quadratic minimization problems with a linear equality/inequality constraint and box constraints are proposed in [27]. Quadratic knapsack problems and problems related to them are studied in [4, 22, 23], and so forth. A nonconvex variant of these problems is considered in [29], and algorithms for the case of convex quadratic objective function are proposed in [4, 9, 12, 21], and so forth. Bounded knapsack sharing is considered in [3]. Algorithms for bound constrained quadratic programming problems are proposed in [19], and minimization of quadratic functions subject to box constraints is considered in [8]. Quasi-Newton updates with bounds are suggested in [5]. A Lagrangian relaxation algorithm for the constrained matrix problem is proposed in [7]. Analytic solutions of nonlinear programs subject to one or two equality constraints are studied in [10], and minimization subject to one linear equality constraint and bounds on the variables is considered in [17]. Iterative quadratic optimization algorithms for pairs of inequalities are proposed in [13]. A polynomial-time algorithm for the resource allocation problem with a convex objective function and nonnegative integer variables is suggested in [14]. Algorithms for the least-distance problem are proposed in [1, 30]. Polynomial algorithms for projecting a point onto a region defined by a linear constraint and box constraints in \mathbb{R}^n are suggested in [28]. An algorithm for finding a projection onto a simple polytope is proposed in [18]. A method for solving a convex integer programming problem is suggested in [26]. The problems of maximizing and minimizing subsums subject to a sum- and a Schur-convex constraint are solved in [15] without the use of the theory of convex programming.

In this paper, we propose iterative algorithms (Sections 3.2 and 3.4) which are based on Theorems 2.1 and 2.4 (Sections 2.1 and 2.3, resp.). Convergence of these algorithms is based on Theorem 3.2 (Section 3.3). Then we pay our attention to some extensions concerning theoretical and computational aspects of the proposed approach (Section 4). In Section 5, we give examples of convex functions $c_j(x_j)$, which are involved in problems under consideration, and some computational results.

This paper is a continuation of the author's work [25] in which we proposed polynomial-time algorithms for a convex separable problem subject to convex separable

inequality constraint of the form “ \leq ” and bounded variables, and generalization of the author’s paper [27], in which the special case of *quadratic* separable objective function is studied.

2. Main results: characterization theorems

2.1. Problem $(C^=)$. First consider the following problem $(C^=)$:

$$\min \left\{ c(\mathbf{x}) = \sum_{j \in J} c_j(x_j) \right\} \quad (2.1)$$

subject to

$$\sum_{j \in J} d_j x_j = \alpha, \quad d_j > 0, \quad j \in J, \quad (2.2)$$

$$a_j \leq x_j \leq b_j, \quad j \in J. \quad (2.3)$$

Suppose that the following assumptions are satisfied.

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

(2a) $\sum_{j \in J} d_j a_j \leq \alpha \leq \sum_{j \in J} d_j b_j$. Otherwise, the constraints (2.2) and (2.3) are inconsistent and $X^= = \emptyset$, where $X^=$ is defined by (2.2) and (2.3).

Let $h_j^=$, $j \in J$, be the value of x_j for which $c'_j(x_j) = 0$. If a finite value with this property does not exist, this means that the function $c_j(x_j)$ does not change the type of its monotonicity, that is, $c_j(x_j)$ is a nondecreasing or nonincreasing function in the interval $(-\infty, +\infty)$. That is why, in case there does not exist a finite value $h_j^=$ such that $c'_j(h_j^=) = 0$, we adopt the following:

- (i) $h_j^= := -\infty$, if $c_j(x_j)$ is a nondecreasing function;
- (ii) $h_j^= := +\infty$, if $c_j(x_j)$ is a nonincreasing function.

The Lagrangian for problem $(C^=)$ is

$$L(\mathbf{x}, \mathbf{u}, \mathbf{v}, \lambda) = \sum_{j \in J} c_j(x_j) + \lambda \left(\sum_{j \in J} d_j x_j - \alpha \right) + \sum_{j \in J} u_j (a_j - x_j) + \sum_{j \in J} v_j (x_j - b_j), \quad (2.4)$$

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 solution $\mathbf{x}^* = (x_j^*)_{j \in J}$ for problem $(C^=)$ are

$$c'_j(x_j^*) + \lambda d_j - u_j + v_j = 0, \quad j \in J, \quad (2.5)$$

$$u_j (a_j - x_j^*) = 0, \quad j \in J, \quad (2.6)$$

$$v_j (x_j^* - b_j) = 0, \quad j \in J, \quad (2.7)$$

$$\lambda \in \mathbb{R}^1, \quad u_j \in \mathbb{R}_+^1, \quad v_j \in \mathbb{R}_+^1, \quad j \in J, \quad (2.8)$$

$$\sum_{j \in J} d_j x_j^* = \alpha, \quad (2.9)$$

$$a_j \leq x_j^* \leq b_j, \quad j \in J, \quad (2.10)$$

where $\lambda, u_j, v_j, j \in J$, are the Lagrange multipliers associated with the constraints (2.2), $a_j \leq x_j, x_j \leq b_j, j \in J$, respectively. If $a_j = -\infty$ or $b_j = +\infty$ for some j , we do not consider the corresponding condition (2.6) ((2.7), resp.) and Lagrange multiplier u_j (v_j , resp.).

Since $c_j(x_j), j \in J$, are convex differentiable functions in one variable, then $c'_j(x_j)$ are monotone nondecreasing functions of $x_j, j \in J$, that is,

$$[c'_j(x_j^1) - c'_j(x_j^2)](x_j^1 - x_j^2) \geq 0 \quad \forall x_j^1, x_j^2, j \in J. \quad (2.11)$$

Since $u_j \geq 0, v_j \geq 0, j \in J$, and since the complementary conditions (2.6) and (2.7) must be satisfied, in order to find $x_j^*, j \in J$, from system (2.5), (2.6), (2.7), (2.8), (2.9), and (2.10), we have to consider all possible cases for u_j, v_j : all u_j, v_j equal to 0; all 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 $2^{|J|} = 2^{2n}$, where $2n$ is the number of all $u_j, v_j, j \in J$, and $|J| = n$. Obviously, this is an enormous number of cases, especially for large-scale problems. For example, when $n = 1500$, we have to consider $2^{3000} \approx 10^{900}$ cases. Moreover, in each case, we have to solve a large-scale system of (nonlinear) equations in $x_j^*, \lambda, u_j, v_j, j \in J$. Therefore, the *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 efficient methods to solve the problems under consideration.

Theorem 2.1 gives necessary and sufficient condition (characterization) of the optimal solution to problem $(C^=)$. 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 $(C^=)$ with $n = 1500$ variables for about 0.0001 seconds on a personal computer.

THEOREM 2.1 (characterization of the optimal solution to problem $(C^=)$). *A feasible solution $\mathbf{x}^* = (x_j^*)_{j \in J} \in X^=$ is an optimal solution to problem $(C^=)$ if and only if there exists a $\lambda \in \mathbb{R}^1$ such that*

$$x_j^* = a_j, \quad j \in J_a^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : \lambda \geq -\frac{c'_j(a_j)}{d_j} \right\}, \quad (2.12)$$

$$x_j^* = b_j, \quad j \in J_b^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : \lambda \leq -\frac{c'_j(b_j)}{d_j} \right\}, \quad (2.13)$$

$$x_j^* : \lambda d_j = -c'_j(x_j^*), \quad j \in J^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : -\frac{c'_j(b_j)}{d_j} \leq \lambda \leq -\frac{c'_j(a_j)}{d_j} \right\}. \quad (2.14)$$

When $c_j(x_j)$ are strictly convex, inequalities defining J^λ in (2.14) are strict.

Proof. Necessity. Let $\mathbf{x}^* = (x_j^*)_{j \in J}$ be an optimal solution to $(C^=)$. Then there exist constants $\lambda, u_j, v_j, j \in J$, such that KKT conditions (2.5), (2.6), (2.7), (2.8), (2.9), and (2.10) are satisfied.

(a) If $x_j^* = a_j$, then $u_j \geq 0$ and $v_j = 0$ according to (2.6), (2.7), and (2.8). Therefore, (2.5) implies that $c'_j(x_j^*) = u_j - \lambda d_j \geq -\lambda d_j$. Since $d_j > 0$, then

$$\lambda \geq -\frac{c'_j(x_j^*)}{d_j} \equiv -\frac{c'_j(a_j)}{d_j}. \quad (2.15)$$

(b) If $x_j^* = b_j$, then $u_j = 0$ and $v_j \geq 0$ according to (2.6), (2.7), and (2.8). Therefore, (2.5) implies that $c'_j(x_j^*) = -v_j - \lambda d_j \leq -\lambda d_j$. Hence,

$$\lambda \leq -\frac{c'_j(x_j^*)}{d_j} \equiv -\frac{c'_j(b_j)}{d_j}. \quad (2.16)$$

(c) If $a_j < x_j^* < b_j$, then $u_j = v_j = 0$ according to (2.6) and (2.7). Therefore, (2.5) implies that $c'_j(x_j^*) = -\lambda d_j$. Since $c_j(x_j)$, $j \in J$, are convex differentiable functions, then $c'_j(x_j)$ are nondecreasing functions, and since $b_j > x_j^*$, $x_j^* > a_j$, $j \in J$, by the assumptions, it follows that

$$c'_j(b_j) \geq c'_j(x_j^*) \geq c'_j(a_j), \quad j \in J. \quad (2.17)$$

Multiplying these inequalities by $-1/d_j$ and using that

$$d_j > 0, \quad \lambda = -\frac{c'_j(x_j^*)}{d_j}, \quad (2.18)$$

we obtain

$$-\frac{c'_j(b_j)}{d_j} \leq \lambda \leq -\frac{c'_j(a_j)}{d_j}. \quad (2.19)$$

When $c_j(x_j)$, $j \in J$, are *strictly* convex, then $a_j < x_j^* < b_j$ implies that $c'_j(b_j) > c'_j(x_j^*) > c'_j(a_j)$, $j \in J$, strictly. Then inequalities in (2.19) are strict.

To describe cases (a), (b), and (c), we introduce the index sets J_a^λ , J_b^λ , J^λ defined by (2.12), (2.13), and (2.14), respectively. It is obvious that $J_a^\lambda \cup J_b^\lambda \cup J^\lambda = J$. The “necessity” part of Theorem 2.1 is proved.

Sufficiency. Conversely, let $\mathbf{x}^* = (x_j^*)_{j \in J} \in X^+$ and components of \mathbf{x}^* satisfy (2.12), (2.13), and (2.14). Set

$$\begin{aligned} \lambda &= -\frac{c'_j(x_j^*)}{d_j}; \quad u_j = v_j = 0 \quad \text{for } j \in J^\lambda; \\ u_j &= c'_j(a_j) + \lambda d_j \quad (\geq 0), \quad v_j = 0 \quad \text{for } j \in J_a^\lambda; \\ u_j &= 0, \quad v_j = -c'_j(b_j) - \lambda d_j \quad (\geq 0) \quad \text{for } j \in J_b^\lambda. \end{aligned} \quad (2.20)$$

Clearly, $\mathbf{x}^*, \lambda, u_j, v_j$, $j \in J$, satisfy KKT conditions (2.5), (2.6), (2.7), (2.8), (2.9), and (2.10), which are necessary and sufficient conditions for a feasible solution to be an optimal solution to a convex minimization problem. Therefore, \mathbf{x}^* is an optimal solution to problem (C^+) defined by (2.1), (2.2), and (2.3).

When $c_j(x_j)$, $j \in J$, are *strictly convex*, this optimal solution is unique. \square

The importance of Theorem 2.1 consists in the fact that it describes components of the optimal solution to problem $(C^=)$ only through the Lagrange multiplier λ associated with the equality constraint (2.2).

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 the convergence of this process in Section 3.

Using that $d_j > 0$, $j \in J$, from monotonicity of $c'_j(x_j)$ and from $a_j \leq b_j$, $j \in J$, it follows that $ub_j \stackrel{\text{def}}{=} -c'_j(b_j)/d_j \leq -c'_j(a_j)/d_j \stackrel{\text{def}}{=} la_j$, $j \in J$, for the expressions by means of which we define the sets J_a^λ , J_b^λ , and J^λ .

The problem how to ensure a feasible solution to problem $(C^=)$ defined by (2.1), (2.2), and (2.3), which is an assumption for Theorem 2.1, is discussed in Section 3.3.

The question whether x_j , $j \in J^\lambda$, are always uniquely determined from (2.14) in $[a_j, b_j]$ is important. In general, if $c(\mathbf{x}) \equiv \sum_{j \in J} c_j(x_j)$ is a strictly convex function, then problem $(C^=)$ has a unique optimal solution in the feasible region $X^=$ defined by (2.2) and (2.3) in case $(C^=)$ has a feasible solution, that is, x_j^* , $j \in J^\lambda$, are uniquely determined from (2.14) in this case. If the parameters a_j , b_j , and so forth of particular problems of type $(C^=)$ are generated in intervals where the functions $c_j(x_j)$ are strictly convex, then the optimal solution to the corresponding problem $(C^=)$, if it has an optimal solution, is unique.

If $c(\mathbf{x})$ is a convex function but not necessarily a strictly convex function, then, as it is known, any local minimum point of $c(\mathbf{x})$ is a global minimum point as well and the set of optimal solutions to a convex minimization problem is convex. Therefore, the optimal value of the objective function subject to (2.2) and (2.3) is the same for all optimal solutions to $(C^=)$ if it has more than one optimal solution. If, for example, (2.14) is a linear equation of x_j^* , then x_j^* , $j \in J^\lambda$, are uniquely determined from (2.14) in this case as well.

2.2. Problem $(C_m^=)$. Denote by $(C_m^=)$ the problem

$$\min \left\{ c(\mathbf{x}) = \sum_{j \in J} c_j(x_j) \right\} \quad (2.21)$$

subject to

$$D\mathbf{x} = \boldsymbol{\alpha}, \quad (2.22)$$

$$\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}, \quad (2.23)$$

where $c_j(x_j)$ are differentiable strictly convex functions, $j \in J$, $D = (d_{ij}) \in \mathbb{R}^{m \times n}$, $\boldsymbol{\alpha} \in \mathbb{R}^m$, $\mathbf{a} = (a_1, \dots, a_n)$, and $\mathbf{b} = (b_1, \dots, b_n) \in \mathbb{R}^n$.

The feasible region (2.22)-(2.23) is an intersection of m hyperplanes (2.22) and the box (2.23).

Problem $(C^=)$, considered in Section 2.1, is a special case of problem $(C_m^=)$ with $m = 1$. We consider problem $(C^=)$ separately because Theorem 2.1 and the algorithm suggested for $(C^=)$ in Section 3.2 are used in the proof of Theorem 2.4 and in the statement of Algorithm 3.2 for problem $(C^=)$, respectively.

Denote by $P_c(D, \alpha, \mathbf{a}, \mathbf{b})$ the solution to problem $(C_m^=)$. Since $c(\mathbf{x})$ is strictly convex as a sum of strictly convex functions, then $P_c(D, \alpha, \mathbf{a}, \mathbf{b})$ is uniquely defined, that is, there is at most one minimum which is both local and global.

Denote $\mathbf{y} = [\mathbf{x}]_{\mathbf{a}}^{\mathbf{b}}$, where $y_j = \min\{\max\{x_j, a_j\}, b_j\}$ for each $j \in J$.

The KKT conditions for $\mathbf{x}^* \in \mathbb{R}^n$ to be a local minimum of $(C_m^=)$ are

$$D\mathbf{x}^* = \alpha, \quad (2.24)$$

$$\mathbf{a} \leq \mathbf{x}^*, \quad (2.25)$$

$$\mathbf{x}^* \leq \mathbf{b}, \quad (2.26)$$

$$\mathbf{c}'(\mathbf{x}^*) + D^T \lambda - \mathbf{u} + \mathbf{v} = \mathbf{0}, \quad (2.27)$$

$$u_j(a_j - x_j^*) = 0, \quad j \in J, \quad (2.28)$$

$$v_j(x_j^* - b_j) = 0, \quad j \in J, \quad (2.29)$$

$$\mathbf{u} \geq \mathbf{0}, \quad (2.30)$$

$$\mathbf{v} \geq \mathbf{0}, \quad (2.31)$$

where $\lambda \in \mathbb{R}^m$, $\mathbf{u}, \mathbf{v} \in \mathbb{R}_+^n$ are the Lagrange multipliers associated with (2.22) and the two inequalities of (2.23), respectively.

The map $\mathbf{c}' \equiv \nabla c : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is strict monotone increasing since c is a strictly convex function. Therefore, $(\nabla c)^{-1} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is well defined.

THEOREM 2.2. *Let $c : \mathbb{R}^n \rightarrow \mathbb{R}$ be separable, differentiable, and strictly convex. Then,*

$$\{P_c(D, \alpha, \mathbf{a}, \mathbf{b})\} = \left\{(\mathbf{c}')^{-1}[-D^T \mathbf{t}]_{\mathbf{c}'(\mathbf{a})}^{\mathbf{c}'(\mathbf{b})} : \mathbf{t} \in \mathbb{R}^m\right\}, \quad (2.32)$$

where $D, \alpha, \mathbf{a}, \mathbf{b}$ are defined above.

Proof. Relation (2.32) is proved by two-way inclusion.

(i) Let $\mathbf{x}^* = P_c(D, \alpha, \mathbf{a}, \mathbf{b})$ for some $\alpha \in \mathbb{R}^m$. Then there exist $\lambda \in \mathbb{R}^m$, $\mathbf{u}, \mathbf{v} \in \mathbb{R}_+^n$ satisfying the KKT conditions (2.24), (2.25), (2.26), (2.27), (2.28), (2.29), (2.30), and (2.31) together with this \mathbf{x}^* .

It follows from (2.27) that

$$D^T \lambda = -\mathbf{c}'(\mathbf{x}^*) + \mathbf{u} - \mathbf{v}, \quad (2.33)$$

that is,

$$\langle \mathbf{D}_j, \lambda \rangle = -c'_j(x_j^*) + u_j - v_j \quad (2.34)$$

for each $j \in J$.

If $\langle \mathbf{D}_j, \lambda \rangle > -c'_j(x_j^*)$, then $u_j > v_j \geq 0$, so $x_j^* = a_j$ according to (2.28), that is,

$$\langle \mathbf{D}_j, \lambda \rangle > -c'_j(x_j^*) \text{ implies that } x_j^* = a_j. \quad (2.35)$$

Similarly, if $\langle \mathbf{D}_j, \lambda \rangle < -c'_j(x_j^*)$, then $v_j > u_j \geq 0$, so $x_j^* = b_j$ according to (2.29), that is,

$$\langle \mathbf{D}_j, \lambda \rangle < -c'_j(x_j^*) \text{ implies that } x_j^* = b_j. \quad (2.36)$$

Since $a_j \leq b_j$, $j \in J$, by assumption, we have three cases to consider.

Case 1. $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle > -c'_j(a_j)$. Then $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle > -c'_j(x_j^*)$ according to (2.25) and the monotonicity of c'_j . Hence, $x_j^* = a_j$ in accordance with (2.35).

Case 2. $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle < -c'_j(b_j)$. Then $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle < -c'_j(x_j^*)$ according to (2.26) and the monotonicity of c'_j . Hence, $x_j^* = b_j$ in accordance with (2.36).

Case 3. $-c'_j(b_j) \leq \langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle \leq -c'_j(a_j)$. If $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle < -c'_j(x_j^*)$, then $x_j^* = b_j$ according to (2.36). Therefore, $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle \geq -c'_j(x_j^*)$ because $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle \geq -c'_j(b_j)$ by the assumption of Case 3, a contradiction. Similarly, if we assume that $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle > -c'_j(x_j^*)$ strictly, this would imply that $x_j^* = a_j$ according to (2.35) and $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle \leq -c'_j(x_j^*)$, a contradiction.

Then $\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle = -c'_j(x_j^*)$, so it follows that $x_j^* = (c'_j)^{-1}(-\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle)$.

In the three cases considered, we have

$$x_j^* = (c'_j)^{-1}[-\langle \mathbf{D}_j, \boldsymbol{\lambda} \rangle]_{c'_j(a_j)}^{c'_j(b_j)}. \quad (2.37)$$

Hence, $\mathbf{x}^* = (\mathbf{c}')^{-1}[-D^T \boldsymbol{\lambda}]_{\mathbf{c}'(\mathbf{a})}^{\mathbf{c}'(\mathbf{b})}$, that is,

$$\{P_c(D, \boldsymbol{\alpha}, \mathbf{a}, \mathbf{b})\} \subseteq \{(\mathbf{c}')^{-1}[-D^T \mathbf{t}]_{\mathbf{c}'(\mathbf{a})}^{\mathbf{c}'(\mathbf{b})} : \mathbf{t} \in \mathbb{R}^m\}. \quad (2.38)$$

(ii) Conversely, suppose that $\mathbf{x}^* \in \mathbb{R}^n$ and $\mathbf{x}^* = (\mathbf{c}')^{-1}[-D^T \mathbf{t}]_{\mathbf{c}'(\mathbf{a})}^{\mathbf{c}'(\mathbf{b})}$ for some $\mathbf{t} \in \mathbb{R}^m$. Set

$$\begin{aligned} \boldsymbol{\alpha} &= D(\mathbf{c}')^{-1}[-D^T \mathbf{t}]_{\mathbf{c}'(\mathbf{a})}^{\mathbf{c}'(\mathbf{b})}, \\ \boldsymbol{\lambda} &= \mathbf{t}, \\ \mathbf{u} &= \mathbf{c}'(\mathbf{a}) + D^T \mathbf{t}, \\ \mathbf{v} &= -\mathbf{c}'(\mathbf{b}) - D^T \mathbf{t}. \end{aligned} \quad (2.39)$$

We have to prove that \mathbf{x}^* , $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$, \mathbf{u} , \mathbf{v} satisfy the KKT conditions (2.24), (2.25), (2.26), (2.27), (2.28), (2.29), (2.30), and (2.31).

Obviously, \mathbf{x}^* and $\boldsymbol{\alpha}$ satisfy (2.24), \mathbf{x}^* satisfies (2.25) and (2.26) (these are (2.23)) according to the definition of $[\mathbf{x}]_{\mathbf{a}}^{\mathbf{b}}$ and the monotonicity of \mathbf{c}' .

In order to verify (2.27), (2.28), (2.29), (2.30), and (2.31), we consider each $j \in J$. There are three possible cases.

Case 1. $\langle \mathbf{D}_j, \mathbf{t} \rangle > -c'_j(a_j)$. Then $c'_j(a_j) + \langle \mathbf{D}_j, \mathbf{t} \rangle > 0$, and since $a_j \leq b_j$, then $-c'_j(b_j) - \langle \mathbf{D}_j, \mathbf{t} \rangle < 0$. Therefore, $x_j^* = a_j$, $\boldsymbol{\lambda} = \mathbf{t}$, $u_j = c'_j(a_j) + \langle \mathbf{D}_j, \mathbf{t} \rangle$, $v_j = 0$.

Case 2. $\langle \mathbf{D}_j, \mathbf{t} \rangle < -c'_j(b_j)$. Then $-c'_j(b_j) - \langle \mathbf{D}_j, \mathbf{t} \rangle > 0$, and since $a_j \leq b_j$, then $c'_j(a_j) + \langle \mathbf{D}_j, \mathbf{t} \rangle < 0$. Therefore, $x_j^* = b_j$, $\boldsymbol{\lambda} = \mathbf{t}$, $u_j = 0$, $v_j = -c'_j(b_j) - \langle \mathbf{D}_j, \mathbf{t} \rangle$.

Case 3. $-c'_j(b_j) \leq \langle \mathbf{D}_j, \mathbf{t} \rangle \leq -c'_j(a_j)$. Then $-c'_j(b_j) - \langle \mathbf{D}_j, \mathbf{t} \rangle \leq 0$, $\langle \mathbf{D}_j, \mathbf{t} \rangle + c'_j(a_j) \leq 0$. Therefore, $x_j^* = (\mathbf{c}')^{-1}(-\langle \mathbf{D}_j, \mathbf{t} \rangle)$, $\boldsymbol{\lambda} = \mathbf{t}$, $u_j = v_j = 0$.

Obviously, in each of the three cases, x_j^* , u_j , v_j ($j \in J$), $\boldsymbol{\lambda}$ satisfy (2.27), (2.28), (2.29), (2.30), and (2.31) as well.

Therefore, $\mathbf{x}^*, \boldsymbol{\alpha}, \boldsymbol{\lambda}, \mathbf{u}, \mathbf{v}$ satisfy the KKT conditions (2.24), (2.25), (2.26), (2.27), (2.28), (2.29), (2.30), and (2.31), so $\mathbf{x}^* \in P_c(D, \boldsymbol{\alpha}, \mathbf{a}, \mathbf{b})$ according to the definition of $P_c(D, \boldsymbol{\alpha}, \mathbf{a}, \mathbf{b})$.

The two-way inclusion implies (2.32). \square

Define the functions $\mathbf{x} : \mathbb{R}^m \rightarrow \mathbb{R}^n$, $\boldsymbol{\alpha} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ by

$$\mathbf{x}(\mathbf{t}) = (\mathbf{c}')^{-1} [-D^T \mathbf{t}]_{\mathbf{c}'(\mathbf{a})}^{\mathbf{c}'(\mathbf{b})}, \quad (2.40)$$

$$\boldsymbol{\alpha}(\mathbf{t}) = D(\mathbf{c}')^{-1} [-D^T \mathbf{t}]_{\mathbf{c}'(\mathbf{a})}^{\mathbf{c}'(\mathbf{b})}. \quad (2.41)$$

Then the following corollary holds.

COROLLARY 2.3. *Vectors $\mathbf{x}^* \in \mathbb{R}^n$, $\boldsymbol{\alpha}^* \in \mathbb{R}^m$ satisfy $\mathbf{x}^* = P_c(D, \boldsymbol{\alpha}^*, \mathbf{a}, \mathbf{b})$ if and only if there exists $\mathbf{t}^* \in \mathbb{R}^m$ such that*

$$\mathbf{x}(\mathbf{t}^*) = \mathbf{x}^*, \quad (2.42)$$

$$\boldsymbol{\alpha}(\mathbf{t}^*) = \boldsymbol{\alpha}^*. \quad (2.43)$$

Proof of Corollary 2.3 follows from the statement of problem $(C_m^=)$ and (2.32).

It follows from Corollary 2.3 that $\mathbf{x}^* = P_c(D, \boldsymbol{\alpha}^*, \mathbf{a}, \mathbf{b})$ can be solved with respect to \mathbf{x}^* for given $\boldsymbol{\alpha}^*$ by first solving (2.43) for \mathbf{t}^* and then calculating \mathbf{x}^* by using (2.42).

Let S be the set of solutions to (2.43) for a particular value of $\boldsymbol{\alpha}^*$:

$$S = \{\mathbf{t} \in \mathbb{R}^m : \boldsymbol{\alpha}(\mathbf{t}) = \boldsymbol{\alpha}^*\}. \quad (2.44)$$

According to (2.41), each component of $\boldsymbol{\alpha}(\mathbf{t})$ is a linear combination of the same set of terms. Each term $(\mathbf{c}')_j^{-1} [-D_j^T \mathbf{t}]_{\mathbf{c}'_j(a_j)}^{\mathbf{c}'_j(b_j)}$ is a smooth function of \mathbf{t} except on the pair of break hyperplanes

$$\begin{aligned} A_j &= \{\mathbf{t} \in \mathbb{R}^m : \langle \mathbf{D}_j, \mathbf{t} \rangle = -c'_j(a_j)\}, \\ B_j &= \{\mathbf{t} \in \mathbb{R}^m : \langle \mathbf{D}_j, \mathbf{t} \rangle = -c'_j(b_j)\}. \end{aligned} \quad (2.45)$$

These break hyperplanes are generalizations of breakpoints considered in Section 3.1.

2.3. Problem $(C^=)$. Consider now the problem $(C^=)$ with linear inequality “ \geq ” constraint (1.2):

$$\min \left\{ c(\mathbf{x}) = \sum_{j \in J} c_j(x_j) \right\} \quad (2.46)$$

subject to

$$\sum_{j \in J} d_j x_j \geq \alpha, \quad d_j > 0, \quad j \in J, \quad (2.47)$$

$$a_j \leq x_j \leq b_j, \quad j \in J. \quad (2.48)$$

Suppose that the following assumptions are satisfied.

(1b) $a_j \leq b_j$ for all $j \in J$.

(2b) $\alpha \leq \sum_{j \in J} d_j b_j$. Otherwise, the constraints (2.47) and (2.48) are inconsistent and $X^\geq = \emptyset$, where X^\geq is defined by (2.47) and (2.48). In addition to this assumption, we suppose that $\sum_{j \in J} d_j a_j \leq \alpha$ (see comments after the proof of Theorem 2.4, Section 2.3).

Let h_j^\geq , $j \in J$, be the value of x_j for which $c'_j(x_j) = 0$. If such a value does not exist, since $c'_j(x_j)$ is a monotone nondecreasing function ($c_j(x_j)$ is convex), we adopt $h_j^\geq = +\infty$ for problem (C^\geq) .

Rewrite problem (C^\geq) in the form

$$\min \left\{ c(\mathbf{x}) = \sum_{j \in J} c_j(x_j) \right\} \quad (2.49)$$

subject to

$$-\sum_{j \in J} d_j x_j \leq -\alpha, \quad d_j > 0, \quad j \in J, \quad (2.50)$$

$$a_j \leq x_j \leq b_j, \quad j \in J. \quad (2.51)$$

Since the linear function $d(\mathbf{x}) \stackrel{\text{def}}{=} -\sum_{j \in J} d_j x_j + \alpha$ is both convex and concave, then (C^\geq) is a convex optimization problem.

Let λ^\geq be the Lagrange multipliers associated with (2.2) (problem (C^\geq)) and with (2.47) (problem (C^\geq)), and let x_j^*, x_j^\geq , $j \in J$, be components of the optimal solutions to (C^\geq) , (C^\geq) , respectively. For the sake of simplicity, we use u_j, v_j , $j \in J$, instead of u_j^\geq, v_j^\geq , $j \in J$, for the Lagrange multipliers associated with $a_j \leq x_j$, $x_j \leq b_j$, $j \in J$, from (2.51), respectively.

The Lagrangian for problem (C^\geq) is

$$L(\mathbf{x}, \mathbf{u}, \mathbf{v}, \lambda^\geq) = \sum_{j \in J} c_j(x_j) + \lambda^\geq \left(-\sum_{j \in J} d_j x_j + \alpha \right) + \sum_{j \in J} u_j (a_j - x_j) + \sum_{j \in J} v_j (x_j - b_j), \quad (2.52)$$

and the KKT conditions for (C^\geq) are

$$c'_j(x_j^\geq) - \lambda^\geq d_j - u_j + v_j = 0, \quad j \in J, \quad (2.53)$$

$$u_j (a_j - x_j^\geq) = 0, \quad j \in J, \quad (2.54)$$

$$v_j (x_j^\geq - b_j) = 0, \quad j \in J, \quad (2.55)$$

$$\lambda^\geq \left(\alpha - \sum_{j \in J} d_j x_j^\geq \right) = 0, \quad \lambda^\geq \in \mathbb{R}_+^1; \quad (2.56)$$

$$\sum_{j \in J} d_j x_j^\geq \geq \alpha, \quad (2.57)$$

$$a_j \leq x_j^\geq \leq b_j, \quad j \in J, \quad (2.58)$$

$$u_j \in \mathbb{R}_+^1, \quad v_j \in \mathbb{R}_+^1, \quad j \in J. \quad (2.59)$$

We can replace (2.53) and (2.56) by

$$c'_j(x_j^{\geq}) + \lambda^{\geq} d_j - u_j + v_j = 0, \quad j \in J, \quad (2.60)$$

$$\lambda^{\geq} \left(\sum_{j \in J} d_j x_j^{\geq} - \alpha \right) = 0, \quad \lambda^{\geq} \in \mathbb{R}_-^1, \quad d_j > 0, \quad (2.61)$$

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

Conditions (2.60), (2.54), (2.55), (2.58), and (2.59) with λ instead of λ^{\geq} are among the KKT conditions for problem (C^{\geq}) .

THEOREM 2.4 (sufficient condition for optimal solution to problem (C^{\geq})). (i) If $\lambda = -c'_j(x_j^*)/d_j \leq 0$, $j \in J^{\lambda}$, then x_j^* , $j \in J$, solve problem (C^{\geq}) as well.

(ii) If $\lambda = -c'_j(x_j^*)/d_j > 0$, $j \in J^{\lambda}$, then x_j^{\geq} , $j \in J$, defined as follows:

- (a) $x_j^{\geq} = b_j$, $j \in J_b^{\lambda}$,
- (b) $x_j^{\geq} = \min\{b_j, h_j^{\geq}\}$, $j \in J^{\lambda}$,
- (c) $x_j^{\geq} = \min\{b_j, h_j^{\geq}\}$ for all $j \in J_a^{\lambda}$ such that $c'_j(a_j) < 0$,
- (d) $x_j^{\geq} = a_j$ for all $j \in J_a^{\lambda}$ such that $c'_j(a_j) \geq 0$,
solve problem (C^{\geq}) .

Proof. (i) Let $\lambda = -c'_j(x_j^*)/d_j \leq 0$, $j \in J^{\lambda}$. Since x_j^* , $j \in J$, satisfy KKT conditions (2.5), (2.6), (2.7), (2.8), (2.9), and (2.10) for problem $(C^=)$ as components of optimal solution to $(C^=)$, then (2.60), (2.54), (2.55), and (2.57) with equality (and therefore (2.61)), (2.58), and (2.59) are satisfied as well (with λ instead of λ^{\geq}). Since they are the KKT necessary and sufficient conditions for problem (C^{\geq}) , then x_j^* , $j \in J$, solve problem (C^{\geq}) as well.

(ii) Let $\lambda = -c'_j(x_j^*)/d_j > 0$, $j \in J^{\lambda}$. Since $\mathbf{x}^* = (x_j^*)_{j \in J}$ is an optimal solution to problem $(C^=)$ by the assumption, then KKT conditions (2.5), (2.6), (2.7), (2.8), (2.9), and (2.10) for problem $(C^=)$ are satisfied. If $\mathbf{x}^{\geq} := (x_j^{\geq})_{j \in J}$ is an optimal solution to (C^{\geq}) , then \mathbf{x}^{\geq} satisfies (2.60), (2.54), (2.55), (2.61), (2.57), (2.58), and (2.59). Since $\lambda > 0$, then λ cannot play the role of λ^{\geq} in (2.60) and (2.61) because λ^{\geq} must be a *nonpositive* real number in (2.60) and (2.61). Therefore x_j^* , which satisfy KKT conditions (2.5), (2.6), (2.7), (2.8), (2.9), and (2.10) for problem $(C^=)$, cannot play the roles of x_j^{\geq} , $j \in J$, in (2.60), (2.54), (2.55), (2.61), (2.57), (2.58), and (2.59). Hence, in the general case, the equality $\sum_{j \in J} d_j x_j^{\geq} = \alpha$ is not satisfied for $x_j = x_j^{\geq}$. Therefore, in order that (2.61) be satisfied, λ^{\geq} must be equal to 0. This conclusion helps us to prove the theorem.

Let $\mathbf{x}^{\geq} := (x_j^{\geq})_{j \in J}$ be defined as in part (ii) of the statement of Theorem 2.4.

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

- (1) $u_j = 0$, $v_j = -c'_j(b_j)$ (≥ 0 since $\lambda > 0$, $d_j > 0$, and according to the definition of J_b^{λ} (2.13)) for $j \in J_b^{\lambda}$;
- (2) $u_j = v_j = 0$ for $j \in J_a^{\lambda}$ such that $c'_j(a_j) < 0$ and for $j \in J^{\lambda}$ such that $h_j^{\geq} < b_j$;
- (3) $u_j = 0$, $v_j = -c'_j(b_j)$ (≥ 0) for $j \in J^{\lambda}$ such that $h_j^{\geq} \geq b_j$;
- (4) $u_j = c'_j(a_j) \geq 0$, $v_j = 0$ for $j \in J_a^{\lambda}$ such that $c'_j(a_j) \geq 0$.

In case (2), we have $c'_j(a_j) < 0 \equiv c'_j(h_j^{\geq})$, therefore $a_j \leq h_j^{\geq} = x_j^{\geq}$ according to the monotonicity of $c'_j(x_j)$ and the definition of x_j^{\geq} in this case. In case (3), since $b_j \leq h_j^{\geq}$, then $c'_j(b_j) \leq c'_j(h_j^{\geq}) = 0$ according to monotonicity of $c'_j(x_j)$, and therefore $v_j := -c'_j(b_j) \geq 0$.

Consequently, conditions (2.58) and (2.59) are satisfied for all j according to (1), (2), (3), and (4).

As we have proved, (2.61) is satisfied with $\lambda^{\geq} = 0$. Since the equality constraint (2.2) $\sum_{j \in J} d_j x_j^* = \alpha$ is satisfied for the optimal solution \mathbf{x}^* to $(C^=)$; since the components of \mathbf{x}^{\geq} defined in the statement of Theorem 2.4(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 $c'_j(a_j) < 0$, are greater than the corresponding components $x_j^* = a_j$, $j \in J_a^{\lambda}$, of \mathbf{x}^* ; and since $d_j > 0$, $j \in J$, then obviously the inequality constraint (2.47) (condition (2.57)) holds for \mathbf{x}^{\geq} . It is easy to check that other conditions (2.60), (2.54), and (2.55) are also satisfied. Thus, x_j^{\geq} , $j \in J$, defined in the statement of Theorem 2.4(ii), satisfy the KKT conditions for problem (C^{\geq}) . Therefore, \mathbf{x}^{\geq} is an optimal solution to problem (C^{\geq}) . \square

According to Theorem 2.4, the optimal solution to problem (C^{\geq}) is obtained by using the optimal solution and optimal value of the Lagrange multiplier λ for problem $(C^=)$. That is why we suppose that $\sum_{j \in J} d_j a_j \leq \alpha$ in addition to assumption (2b) (see Step 1 of Algorithm 3.2, Section 3.4), as we assumed this in assumption (2a) for problem $(C^=)$.

3. Algorithms

3.1. Analysis of the optimal solution to problem $(C^=)$. Since the optimal solution \mathbf{x}^* to problem $(C^=)$ depends on λ , we consider the components of \mathbf{x}^* as functions of λ for different $\lambda \in \mathbb{R}^1$:

$$x_j(\lambda) = \begin{cases} a_j, & j \in J_a^{\lambda}, \\ b_j, & j \in J_b^{\lambda}, \\ x_j^* : c'_j(x_j^*) + \lambda d_j = 0, & j \in J^{\lambda}. \end{cases} \quad (3.1)$$

Functions $x_j(\lambda)$, $j \in J$, are piecewise linear, monotone, piecewise differentiable functions of λ , with two breakpoints at $\lambda = -c'_j(a_j)/d_j$ and $\lambda = -c'_j(b_j)/d_j$.

Let

$$\delta(\lambda) \stackrel{\text{def}}{=} \sum_{j \in J_a^{\lambda}} d_j a_j + \sum_{j \in J_b^{\lambda}} d_j b_j + \sum_{j \in J^{\lambda}} d_j x_j(\lambda) - \alpha. \quad (3.2)$$

According to $u_j = v_j = 0$, $j \in J^{\lambda}$, condition (2.5) becomes

$$c'_j(x_j(\lambda)) + \lambda d_j = 0, \quad j \in J^{\lambda}. \quad (3.3)$$

If we differentiate both sides of each of these expressions with respect to λ (using that $c''_j(x_j)$, $j \in J^{\lambda}$, exist by assumption; $x'_j(\lambda)$, $j \in J^{\lambda}$, exist because $x_j(\lambda)$ are defined by $x_j(\lambda) = x_j^*$ such that $c'_j(x_j^*) + \lambda d_j = 0$ for $j \in J^{\lambda}$), we obtain

$$c''_j(x_j(\lambda)) x'_j(\lambda) + d_j = 0, \quad j \in J^{\lambda}. \quad (3.4)$$

Therefore,

$$x'_j(\lambda) = -\frac{d_j}{c''_j(x_j(\lambda))}, \quad j \in J^\lambda, \quad (3.5)$$

and since $c''_j(x_j) \geq 0$, $j \in J$, as the second derivatives of convex differentiable functions; $d_j > 0$ by the assumption, then $x'_j(\lambda) < 0$, $j \in J^\lambda$. (If we assume that $c''_j(x_j(\lambda)) = 0$, then $d_j = 0$ according to (3.4). However, $d_j > 0$, $j \in J$, by the assumption, a contradiction.) Consequently,

$$\delta'(\lambda) \equiv \sum_{j \in J^\lambda} d_j x'_j(\lambda) < 0, \quad (3.6)$$

when $J^\lambda \neq \emptyset$, and $\delta'(\lambda) = 0$ when $J^\lambda = \emptyset$. Hence, $\delta(\lambda)$ is a *monotone nonincreasing function* of $\lambda \in \mathbb{R}^1$.

Using equation $\delta(\lambda) = 0$, where $\delta(\lambda)$ is defined by (3.2), we are always able to determine λ as an implicit function of $\mathbf{x} : \lambda = \lambda(\mathbf{x})$, because $\delta'(\lambda) < 0$ when $J^\lambda \neq \emptyset$ according to (3.6) (it is important that $\delta'(\lambda) \neq 0$). Moreover, since $\delta(\lambda)$ is a linear function of $\mathbf{x}(\lambda)$, it is always possible to obtain a *closed-form expression* of λ . It turns out that for our purpose, without loss of generality, we can assume that $\delta'(\lambda) \neq 0$, that is, $\delta(\lambda)$ depends of λ , which means that $J^\lambda \neq \emptyset$ (see the third paragraph of *Remark 3.3*).

At iteration k of the implementation of algorithms, denote by $\lambda^{(k)}$ the value of the Lagrange multiplier associated with the constraint (2.2) ((2.47), resp.), by $\alpha^{(k)}$ the right-hand side of (2.2) ((2.47), resp.); by $J^{(k)}$, $J_a^{(k)}$, $J_b^{(k)}$, J^λ the current sets J , J_a^λ , J_b^λ , J^λ , respectively.

3.2. Algorithm 3.1 (for problem $(C^=)$). According to Theorem 2.1 and the preliminary analysis, we can suggest the following algorithm for solving problem $(C^=)$ with strictly convex differentiable functions $c_j(x_j)$, see Algorithm 3.1.

Remark 3.1. To avoid a possible “endless loop” in programming Algorithm 3.1, the criterion of Step 5 to go to Step 8 at iteration k usually is not $\delta(\lambda^{(k)}) = 0$ but $\delta(\lambda^{(k)}) \in [-\varepsilon, \varepsilon]$ where $\varepsilon > 0$ is some tolerance value up to which the equality $\delta(\lambda^*) = 0$ (i.e., $\sum_{j \in J} d_j x_j^* = \alpha$) must be satisfied.

3.3. Convergence and complexity of Algorithm 3.1. Theorem 3.2 states convergence of Algorithm 3.1, that is, “convergence” of $\lambda^{(k)}$, $J^{(k)}$, $J_a^{(k)}$, $J_b^{(k)}$, generated by Algorithm 3.1, to the optimal λ , J^λ , J_a^λ , J_b^λ from Theorem 2.1, respectively.

THEOREM 3.2. *Let $\lambda^{(k)}$ be the sequence generated by Algorithm 3.1. Then,*

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

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 3.1.

Step 1 (initialization). $J := \{1, \dots, n\}$, $k := 0$, $J^{(0)} := J$, $\alpha^{(0)} := \alpha$, $n^{(0)} := n$, $J_a^\lambda := \emptyset$, $J_b^\lambda := \emptyset$, initialize $h_j^=$, $j \in J$. If $\sum_{j \in J} d_j a_j \leq \alpha \leq \sum_{j \in J} d_j b_j$, go to Step 2, else go to Step 9.

Step 2. $J^{\lambda(k)} := J^{(k)}$. Calculate $\lambda^{(k)}$ by using the closed-form expression of λ , determined from the equality constraint $\sum_{j \in J^{\lambda(k)}} d_j x_j = \alpha^{(k)}$, where x_j are given by (2.14). Go to Step 3.

Step 3. Construct the sets $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ through (2.12), (2.13), and (2.14) (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 Step 4.

Step 4. Calculate

$$\delta(\lambda^{(k)}) := \sum_{j \in J_a^{\lambda(k)}} d_j a_j + \sum_{j \in J_b^{\lambda(k)}} d_j b_j + \sum_{j \in J^{\lambda(k)}} d_j x_j^* - \alpha^{(k)}, \quad (3.7)$$

where x_j^* , $j \in J^{\lambda(k)}$, are determined from (2.14) with $\lambda = \lambda^{(k)}$. Go to Step 5.

Step 5. If $\delta(\lambda^{(k)}) = 0$ or $J^{\lambda(k)} = \emptyset$, then $\lambda := \lambda^{(k)}$, $J_a^\lambda := J_a^{\lambda(k)}$, $J_b^\lambda := J_b^{\lambda(k)}$, $J^\lambda := J^{\lambda(k)}$, go to Step 8;

else if $\delta(\lambda^{(k)}) > 0$, go to Step 6;

else if $\delta(\lambda^{(k)}) < 0$, go to Step 7.

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

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

Step 8. $x_j^* := a_j$ for $j \in J_a^\lambda$; $x_j^* := b_j$ for $j \in J_b^\lambda$; assign x_j^* the value determined from (2.14) for $j \in J^\lambda$. Go to Step 10.

Step 9. Problem $(C^=)$ has no optimal solution because $X^= = \emptyset$.

Step 10. End.

ALGORITHM 3.1

(i) Let $\delta(\lambda^{(k)}) > 0$. Using Step 6 of Algorithm 3.1, which is performed when $\delta(\lambda^{(k)}) > 0$, we get

$$\sum_{j \in J^{(k+1)}} d_j x_j^{(k)} \equiv \sum_{j \in J^{(k+1)}} d_j x_j^{(k)} = \sum_{j \in J^{(k)} \setminus J_a^{\lambda(k)}} d_j x_j^{(k)} = \alpha^{(k)} - \sum_{j \in J_a^{\lambda(k)}} d_j x_j^{(k)}. \quad (3.8)$$

Let $j \in J_a^{\lambda(k)}$. According to the definition (2.12) of $J_a^{\lambda(k)}$ and relation $\lambda^{(k)} d_j = -c'_j(x_j^{(k)})$, we have

$$-\frac{c'_j(a_j)}{d_j} \leq \lambda^{(k)} = -\frac{c'_j(x_j^{(k)})}{d_j}. \quad (3.9)$$

Multiplying this inequality by $-d_j < 0$, we obtain $c'_j(a_j) \geq c'_j(x_j^{(k)})$. Hence, $a_j \geq x_j^{(k)}$, $j \in J_a^{\lambda(k)}$, in accordance with monotonicity of $c'_j(x_j)$.

Taking into consideration (3.8), $d_j > 0$, $a_j \geq x_j^{(k)}$, $j \in J_a^{\lambda(k)}$, and Step 6, we get

$$\sum_{j \in J_a^{\lambda(k+1)}} d_j x_j^{(k)} = \alpha^{(k)} - \sum_{j \in J_a^{\lambda(k)}} d_j x_j^{(k)} \geq \alpha^{(k)} - \sum_{j \in J_a^{\lambda(k)}} d_j a_j = \alpha^{(k+1)} = \sum_{j \in J_a^{\lambda(k+1)}} d_j x_j^{(k+1)}. \quad (3.10)$$

Since $d_j > 0$, $j \in J$, then 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

$$\lambda^{(k)} = -\frac{c'_{j_0}(x_{j_0}^{(k)})}{d_{j_0}} \leq -\frac{c'_{j_0}(x_{j_0}^{(k+1)})}{d_{j_0}} = \lambda^{(k+1)}, \quad (3.11)$$

where we have used the relationship (2.14) between $\lambda^{(k)}$ and $x_j^{(k)}$ for $j \in J^{\lambda(k)}$ according to Step 2 of Algorithm 3.1, the fact that $-c'_{j_0}(x_j)$ is a monotone nonincreasing function, and $d_j > 0$, $j \in J$.

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

Remark 3.3. Since we do not know the optimal value of λ which is involved in the statement of Theorem 2.1, we approximate the value of λ until we obtain the optimal value of λ at the last iteration of algorithm performance. In order to determine the current value $\lambda^{(k)}$ of λ at each iteration (including the initial value), we assume that $J^{\lambda(k)} = J^{(k)}$ at the beginning of the corresponding iteration (Step 2).

Theorem 3.2, definitions of J_a^λ (2.12), J_b^λ (2.13), and Steps 6 and 7 of Algorithm 3.1 allow us to assert that the values of $\lambda^{(k)}$, $k = 0, 1, \dots$, calculated at Step 2, are such that $j \in J_a^{\lambda(k)}$ implies that $j \in J_a^{\lambda(k+1)}$, $j \in J_b^{\lambda(k)}$ implies that $j \in J_b^{\lambda(k+1)}$, and since $J^{\lambda(k)}$ is reduced (Steps 6 and 7), then $j \in J^{\lambda(k+1)}$ implies that $j \in J^{\lambda(k)}$; that is, we have $J_a^{\lambda(k)} \subseteq J_a^{\lambda(k+1)}$, $J_b^{\lambda(k)} \subseteq J_b^{\lambda(k+1)}$, and $J^{\lambda(k)} \supseteq J^{\lambda(k+1)}$. This means that if j belongs to current index set $J_a^{\lambda(k)}$, then j belongs to next index set $J_a^{\lambda(k+1)}$ and, therefore, to the “optimal” index set J_a^λ according to Theorem 3.2 and definition (2.12); the same holds true about the sets $J_b^{\lambda(k)}$ and J_b^λ (2.13). Therefore, $\lambda^{(k)}$ converges to the optimal value λ from Theorem 2.1 and $J_a^{\lambda(k)}, J_b^{\lambda(k)}, J^{\lambda(k)}$ “converge” to $J_a^\lambda, J_b^\lambda, J^\lambda$, 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 $J_a^\lambda, J_b^\lambda, J^\lambda$ are in accordance with Theorem 2.1. The final sets $J_a^\lambda, J_b^\lambda, J^\lambda$ are constructed at Step 1 or at Step 5 (when $\delta(\lambda^{(k)}) = 0$ or $J^{\lambda(k)} = \emptyset$) of iteration k , where k is the number of the last iteration of algorithm performance.

Since at the beginning of Algorithm 3.1, we have $J^{\lambda(0)} := J$ (Steps 1, 2) and since $J^{\lambda(k)} \supseteq J^{\lambda(k+1)}$, then $J^{\lambda(k)} \neq \emptyset$ for all $k \leq k_0$, where k_0 is some nonnegative integer. If we obtain $J^{\lambda(k_0)} = \emptyset$, this would mean that $J_a^{\lambda(k_0)} \cup J_b^{\lambda(k_0)} = J$, that is, the problem has been already solved at iteration k_0 , and $\delta(\lambda^{(k_0)}) = \text{const}$.

Algorithm 3.1 belongs to the group of so-called “active set” algorithms.

At each iteration, Algorithm 3.1 calculates the value of at least one variable (Steps 6, 7, and 8) and at each iteration, we solve a problem of the form $(C^=)$ but of *less dimension* (Steps 2, 3, 4, 5, 6, and 7). Therefore, Algorithm 3.1 is finite and it converges with at most $n = |J|$ iterations, that is, the iteration complexity of Algorithm 3.1 is $\mathcal{O}(n)$.

Step 1 takes time $\mathcal{O}(n)$. The calculation of $x_j^{(k)}$, $j \in J$, and $\lambda^{(k)}$ requires $\mathcal{O}(n)$ 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 $\mathcal{O}(n)$ because at these steps, we assign some of the x_j 's optimal value, and since the number of all x_j 's is n , then Steps 6, 7, and 8 take time $\mathcal{O}(n)$. Hence, Algorithm 3.1 has $\mathcal{O}(n^2)$ running time and it belongs to the class of strongly polynomially bounded algorithms.

As the computational experiments show (Section 5), 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 (2.12), (2.13), and (2.14). In practice, Algorithm 3.1 has $\mathcal{O}(n)$ running time.

Consider the *feasibility* of $\mathbf{x}^* = (x_j^*)_{j \in J}$ generated by Algorithm 3.1.

Components $x_j^* = a_j$, $j \in J_a^\lambda$, and $x_j^* = b_j$, $j \in J_b^\lambda$, obviously satisfy (2.3). Using $-c'_j(b_j)/d_j < \lambda \equiv -c'_j(x_j^*)/d_j < -c'_j(a_j)/d_j$, $j \in J^\lambda$, and $d_j > 0$, $j \in J$, it follows that $c'_j(a_j) < c'_j(x_j^*) < c'_j(b_j)$, $j \in J^\lambda$. Therefore, $a_j \leq x_j^* \leq b_j$ for $j \in J^\lambda$ as well according to the monotonicity of $c'_j(x_j)$. Hence, all x_j^* , $j \in J$, satisfy (2.3).

In the sequel, since at each iteration $\lambda^{(k)}$ is determined from the “current” equality constraint (2.2) (Step 2 of Algorithm 3.1) and since x_j , $j \in J$, are determined in accordance with $\lambda^{(k)}$ at each iteration (Steps 5, 6, 7, and 8 of Algorithm 3.1), then \mathbf{x}^* satisfies (2.2) as well.

Therefore \mathbf{x}^* , obtained by Algorithm 3.1, is feasible for $(C^=)$, which is an assumption of Theorem 2.1.

3.4. Algorithm 3.2 (for problem (C^\geq)). Algorithm 3.2 for solving problem (C^\geq) with strictly convex differentiable functions $c_j(x_j)$ is based on Theorem 2.4 and Algorithm 3.1 (see Algorithm 3.2).

Since Algorithm 3.2 is based on Theorem 2.4 and Algorithm 3.1 and since the “iterative” Steps 2, 3, 4, 5, 6, and 7 of Algorithms 3.1 and 3.2 are the same, then “convergence” of Algorithm 3.2 follows from Theorem 3.2 as well. Because of the same reason, computational complexity of Algorithm 3.2 is the same as that of Algorithm 3.1.

4. Extensions

4.1. Theoretical aspects. Up to now, we required $d_j > 0$, $j \in J$, in (2.2) and (2.47) of problems $(C^=)$ and (C^\geq) , respectively. However, if it is allowed that $d_j = 0$ for some $j \in J$ in problems $(C^=)$ and (C^\geq) , then for such indices j , we cannot construct the expressions $-c'_j(a_j)/d_j$ and/or $-c'_j(b_j)/d_j$, by means of which we define sets J_a^λ , J_b^λ , and J^λ for the corresponding problem. In these cases, x_j 's are not involved in (2.2) (in (2.47), resp.) for such indices j . It turns out that we can cope with this difficulty and solve problems $(C^=)$ and (C^\geq) with $d_j = 0$ for some indices $j \in J$.

Step 1 (initialization). $J := \{1, \dots, n\}$; $k := 0$; $J^{(0)} := J$; $\alpha^{(0)} := \alpha$, $n^{(0)} := n$; $J_a^\lambda := \emptyset$, $J_b^\lambda := \emptyset$, initialize h_j^\geq , $j \in J$. If $\sum_{j \in J} d_j a_j \leq \alpha \leq \sum_{j \in J} d_j b_j$, go to Step 2, else go to Step 9.

Steps 2, 3, 4, 5, 6, and 7 are the same as Steps 2, 3, 4, 5, 6, and 7 of Algorithm 3.1, respectively.

Step 8. If $\lambda \leq 0$, then $x_j^\geq := a_j$ for $j \in J_a^\lambda$, $x_j^\geq := b_j$ for $j \in J_b^\lambda$, assign x_j^\geq the value determined through (2.14) for $j \in J^\lambda$, go to Step 10; else if $\lambda > 0$, then

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

$$x_j^\geq := \min\{b_j, h_j^\geq\} \text{ for } j \in J^\lambda,$$

$$\text{if } j \in J_a^\lambda \text{ and } c'_j(a_j) < 0, \text{ then } x_j^\geq := \min\{b_j, h_j^\geq\};$$

$$\text{else if } j \in J_a^\lambda \text{ and } c'_j(a_j) \geq 0, \text{ then } x_j^\geq := a_j;$$

go to Step 10.

Step 9. Problem (C^\geq) has no optimal solution because $X^\geq = \emptyset$ or there do not exist $x_j^* \in [a_j, b_j]$, $j \in J$, such that $\sum_{j \in J} d_j x_j^* = \alpha$.

Step 10. End.

ALGORITHM 3.2

Denote

$$Z0 = \{j \in J : d_j = 0\}. \quad (4.1)$$

Here, “0” denotes the “computer zero.” In particular, when $J = Z0$ and $\alpha = 0$, then $X^=$ (or X^\geq) is defined only by (2.3) (by (2.48), resp.).

THEOREM 4.1 (characterization of the optimal solution to problem $(C^=)$: an extended version). *Problem $(C^=)$ can be decomposed into two subproblems: $(C1^=)$ for $j \in Z0$ and $(C2^=)$ for $j \in J \setminus Z0$. The optimal solution to $(C1^=)$ is*

$$x_j^* = \begin{cases} a_j, & j \in Z0, h_j^= \leq a_j, \\ b_j, & j \in Z0, h_j^= \geq b_j, \\ h_j^=, & j \in Z0, a_j < h_j^= < b_j. \end{cases} \quad (4.2)$$

The optimal solution to $(C2^=)$ is given by (2.12), (2.13), and (2.14) with $J := J \setminus Z0$.

Proof. Necessity. Let $\mathbf{x}^* = (x_j^*)_{j \in J}$ be an optimal solution to $(C^=)$.

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

$$c'_j(x_j^*) - u_j + v_j = 0, \quad j \in Z0, \quad (4.3)$$

and (2.6), (2.7), (2.8), (2.9), and (2.10).

(a) If $x_j^* = a_j$, then $u_j \geq 0$, $v_j = 0$ according to (2.6), (2.7), and (2.8). It follows from (4.3) and the definition of $h_j^=$ that $c'_j(x_j^*) = u_j \geq 0 \equiv c'_j(h_j^=)$. Therefore, $x_j^* = a_j \geq h_j^=$ according to monotonicity of $c'_j(x_j)$.

(b) If $x_j^* = b_j$, then $u_j = 0, v_j \geq 0$ according to (2.6), (2.7), and (2.8). Therefore, (4.3) implies that $c'_j(x_j^*) = -v_j \leq 0 \equiv c'_j(h_j^-)$. Using monotonicity of $c'_j(x_j)$, we obtain $x_j^* = b_j \leq h_j^-$.

(c) If $a_j < x_j^* < b_j$, then $u_j = v_j = 0$ according to (2.6) and (2.7). Therefore, (4.3) implies that $-c'_j(x_j^*) = 0$, that is, $x_j^* = h_j^-$ according to definition of h_j^- .

(2) Components of the optimal solution to $(C2^=)$ are obtained by using the approach as that of the necessity part of the proof of Theorem 2.1 but with the reduced index set $J := J \setminus Z0$.

Sufficiency. Conversely, let $\mathbf{x}^* \in X^=$ and components of \mathbf{x}^* satisfy (4.2) for $j \in Z0$, and (2.12), (2.13), and (2.14) for $j \in J \setminus Z0$. Set

$$\begin{aligned} \lambda &= 0; & u_j = v_j &= 0 \quad \text{for } a_j < x_j^* < b_j, j \in Z0; \\ u_j &= c'_j(a_j) \quad (\geq 0), & v_j &= 0 \quad \text{for } x_j^* = a_j, j \in Z0; \\ u_j &= 0, & v_j &= -c'_j(b_j) \quad (\geq 0) \quad \text{for } x_j^* = b_j, j \in Z0; \\ \lambda &= -\frac{c'_j(x_j^*)}{d_j}; & u_j = v_j &= 0 \quad \text{for } a_j < x_j^* < b_j, j \in J \setminus Z0; \\ u_j &= c'_j(a_j) + \lambda d_j \quad (\geq 0), & v_j &= 0 \quad \text{for } x_j^* = a_j, j \in J \setminus Z0; \\ u_j &= 0, & v_j &= -c'_j(b_j) - \lambda d_j \quad (\geq 0) \quad \text{for } x_j^* = b_j, j \in J \setminus Z0. \end{aligned} \tag{4.4}$$

It can be verified that $\mathbf{x}^*, \lambda, u_j, v_j, j \in J$, satisfy the KKT conditions (2.5), (2.6), (2.7), (2.8), (2.9), and (2.10). Then \mathbf{x}^* with components (4.2) for $j \in Z0$, and (2.12), (2.13), and (2.14) with $J := J \setminus Z0$ is an optimal solution to problem $(C^=) = (C1^=) \cup (C2^=)$. \square

Thus, with the use of Theorem 4.1, we can express components $x_j^*, j \in Z0$, of the optimal solution to $(C^=)$ (and therefore those to problem (C^{\geq})) *without* the necessity of constructing the expressions $-c'_j(a_j)/d_j$ and $-c'_j(b_j)/d_j$ with $d_j = 0$.

Since Theorem 2.4 and Algorithm 3.2 are based on the sets of indices $J_a^\lambda, J_b^\lambda, J^\lambda$ of problem $(C^=)$, then Theorem 4.1 solves the problem of decomposition of problem (C^{\geq}) as well.

With the use of set $Z0$, we can deduce the following about checking whether the feasible region is empty or nonempty when $J = Z0$ for problems $(C^=)$ and (C^{\geq}) .

When $J = Z0$, $a_j \leq b_j, j \in J$, $\alpha = 0$, the corresponding feasible regions are always nonempty and it is not necessary to check anything else in this case.

4.2. Computational aspects. Algorithms 3.1 and 3.2 are also applicable in cases when $a_j = -\infty$ for some $j \in J$ and/or $b_j = \infty$ for some $j \in J$. However, if we use the computer values of $-\infty$ and $+\infty$ at Step 1 of the algorithms to check whether the corresponding feasible region is empty or nonempty and at Step 3 in the expressions $-c'_j(x_j)/d_j$ with $x_j = -\infty$ and/or $x_j = +\infty$, by means of which we construct sets $J_a^\lambda, J_b^\lambda, J^\lambda$, 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 above discussion, it is convenient to do the following.

Construct the sets of indices

$$\begin{aligned} \text{SVN} &= \{j \in J \setminus Z0 : a_j > -\infty, b_j < +\infty\}, \\ \text{SV1} &= \{j \in J \setminus Z0 : a_j > -\infty, b_j = +\infty\}, \\ \text{SV2} &= \{j \in J \setminus Z0 : a_j = -\infty, b_j < +\infty\}, \\ \text{SV} &= \{j \in J \setminus Z0 : a_j = -\infty, b_j = +\infty\}. \end{aligned} \quad (4.5)$$

It is obvious that $Z0 \cup \text{SV} \cup \text{SV1} \cup \text{SV2} \cup \text{SVN} = J$, that is, the set $J \setminus Z0$ is partitioned into the four sets 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 $J_a^\lambda, J_b^\lambda, J^\lambda$ without the necessity of calculating the values $-c'_j(x_j)/d_j$ with $x_j = -\infty$ or $x_j = +\infty$, except for the sets $J, Z0, \text{SV}, \text{SV1}, \text{SV2}, \text{SVN}$, we need some subsidiary sets defined as follows.

For SVN,

$$\begin{aligned} J^{\lambda \text{SVN}} &= \left\{ j \in \text{SVN} : -\frac{c'_j(b_j)}{d_j} < \lambda < -\frac{c'_j(a_j)}{d_j} \right\}, \\ J_a^{\lambda \text{SVN}} &= \left\{ j \in \text{SVN} : \lambda \geq -\frac{c'_j(a_j)}{d_j} \right\}, \\ J_b^{\lambda \text{SVN}} &= \left\{ j \in \text{SVN} : \lambda \leq -\frac{c'_j(b_j)}{d_j} \right\}. \end{aligned} \quad (4.6)$$

For SV1,

$$\begin{aligned} J^{\lambda \text{SV1}} &= \left\{ j \in \text{SV1} : \lambda < -\frac{c'_j(a_j)}{d_j} \right\}, \\ J_a^{\lambda \text{SV1}} &= \left\{ j \in \text{SV1} : \lambda \geq -\frac{c'_j(a_j)}{d_j} \right\}. \end{aligned} \quad (4.7)$$

For SV2,

$$\begin{aligned} J^{\lambda \text{SV2}} &= \left\{ j \in \text{SV2} : \lambda > -\frac{c'_j(b_j)}{d_j} \right\}, \\ J_b^{\lambda \text{SV2}} &= \left\{ j \in \text{SV2} : \lambda \leq -\frac{c'_j(b_j)}{d_j} \right\}. \end{aligned} \quad (4.8)$$

For SV,

$$J^{\lambda \text{SV}} = \text{SV}. \quad (4.9)$$

Then,

$$\begin{aligned} J^\lambda &:= J^{\lambda \text{SVN}} \cup J^{\lambda \text{SV1}} \cup J^{\lambda \text{SV2}} \cup J^{\lambda \text{SV}}, \\ J_a^\lambda &:= J_a^{\lambda \text{SVN}} \cup J_a^{\lambda \text{SV1}}, \\ J_b^\lambda &:= J_b^{\lambda \text{SVN}} \cup J_b^{\lambda \text{SV2}}. \end{aligned} \quad (4.10)$$

We use the sets J^λ , J_a^λ , J_b^λ in (4.10) as the corresponding sets with the same names in Algorithms 3.1 and 3.2.

The reason to construct namely the sets (4.5), (4.6), (4.7), (4.8), and (4.9) is the following.

If $j \in \text{SVN}$, then none of the a_j 's is equal to $-\infty$ and none of the b_j 's is equal to $+\infty$. That is why there is not any peculiarity of the described type for such indices j .

If $j \in \text{SV1}$, that is, if $a_j > -\infty$ and $b_j = +\infty$, then $v_j = 0$, $j \in \text{SV1}$, according to (2.7) for problem $(C^=)$ and according to (2.55) for problem (C^{\geq}) . Then $j \in J_a^\lambda$ or $j \in J^\lambda$ using the same reasoning as in (a) and (c) from the proof (necessity part) of Theorem 2.1. Therefore, it is sufficient to consider only sets of the type of J_a^λ and J^λ for SV1, and we have denoted these sets by $J_a^{\lambda \text{SV1}}$ and $J^{\lambda \text{SV1}}$ in (4.7), respectively.

Similarly, if $j \in \text{SV2}$, then $b_j < +\infty$ and $a_j = -\infty$ for these j 's. Then $u_j = 0$ according to (2.6) for problem $(C^=)$ or according to (2.54) for problem (C^{\geq}) . Hence $j \in J_b^\lambda$ or $j \in J^\lambda$ according to cases (b) and (c) of the proof (necessity part) of Theorem 2.1. That is why it is sufficient to consider only sets of the type J_b^λ and J^λ for SV2. We have denoted these sets by $J_b^{\lambda \text{SV2}}$ and $J^{\lambda \text{SV2}}$ in (4.8), respectively.

If $j \in \text{SV}$, then $a_j = -\infty$ and $b_j = +\infty$. Therefore, $u_j = v_j = 0$ according to (2.6) and (2.7) for problem $(C^=)$, and according to (2.54) and (2.55) for problem (C^{\geq}) . Hence $j \in J^\lambda$ according to (c) from the proof (necessity part) of Theorem 2.1. Therefore $j \in \text{SV}$ implies that $j \in J^\lambda$, and we have denoted by $J^{\lambda \text{SV}}$ the set $\{j \in \text{SV} : -c'_j(b_j)/d_j < \lambda < -c'_j(a_j)/d_j\}$ in this case.

Since Theorem 2.4 (sufficient condition for solution to (C^{\geq})) is based on the index sets for problem $(C^=)$ from Theorem 2.1, these conclusions are also valid for problem (C^{\geq}) .

The assumption that $d_j \geq 0$, $j \in J$, for problems $(C^=)$ and (C^{\geq}) helps us to draw the following conclusions.

About problem $(C^=)$. (i) If $\text{SVN} = J \setminus Z_0$, that is, if a_j 's and b_j 's are finite for all $j \in J$ which are involved in (2.2) with $d_j \neq 0$, then the checking whether X is nonempty is

$$\sum_{j \in \text{SVN}} d_j a_j \leq \alpha \leq \sum_{j \in \text{SVN}} d_j b_j. \quad (4.11)$$

(ii) Else if $\text{SV1} \cup \text{SVN} = J \setminus Z_0$, that is, if all a_j 's are finite but some of (or all) b_j 's are equal to $+\infty$ for the variables which are involved in (2.2), then the checking is whether

$$\sum_{j \in J \setminus Z_0} d_j a_j \leq \alpha \quad (4.12)$$

and it is not necessary to check whether $\alpha \leq \sum_{j \in J \setminus Z_0} d_j b_j$ in this case.

(1¹) (initialization) $J := \{1, \dots, n\}$; $k := 0$; $J^{(0)} := J$; $\alpha^{(0)} := \alpha$, $n^{(0)} := n$; $J_a^\lambda := \emptyset$, $J_b^\lambda := \emptyset$, initialize h_j^\pm , $j \in J$.

Construct the set Z_0 . If $j \in Z_0$, then

if $h_j^\pm \leq a_j$, then $x_j^* := a_j$;
 else if $h_j^\pm \geq b_j$, then $x_j^* := b_j$;
 else if $a_j < h_j^\pm < b_j$, then $x_j^* := h_j^\pm$.

If $J = Z_0$ and $\alpha = 0$, go to Step 10, else if $J = Z_0$ and $\alpha \neq 0$, go to Step 9.

Set $J := J \setminus Z_0$, $J^{(0)} := J$, $n^{(0)} := n - |Z_0|$, $\alpha^{(0)} := \alpha - \sum_{j \in Z_0} d_j x_j^* \equiv \alpha$.

Construct the sets SVN, SV1, SV2, SV.

If SVN = J , then

if $\sum_{j \in J} d_j a_j \leq \alpha \leq \sum_{j \in J} d_j b_j$, go to Step 2;
 else go to Step 9 (feasible region X^\pm is empty)

else if $SV1 \cup SVN = J$, then

if $\sum_{j \in J} d_j a_j \leq \alpha$, go to Step 2;
 else go to Step 9 (feasible region X^\pm is empty)

else 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 (feasible region X^\pm is empty);

else if $SV \neq \emptyset$, go to Step 2 (feasible region X^\pm is always nonempty).

(3¹) Construct the sets $J_a^\lambda SVN$, $J_a^\lambda SVN$, $J_b^\lambda SVN$, $J_b^\lambda SVN$, $J_a^\lambda SV^1$, $J_b^\lambda SV^1$, $J_a^\lambda SV^2$, $J_b^\lambda SV^2$, $J^\lambda SV$ (with $J^{(k)}$ instead of J).

Construct the sets $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ by using (4.10) and find their cardinalities. Go to Step 4.

ALGORITHM 4.1. About Algorithm 3.1.

(iii) Else if $SV2 \cup SVN = J \setminus Z_0$, that is, if all b_j 's are finite but some of (or all) a_j 's are equal to $-\infty$ for variables which are involved in (2.2), then the checking is whether

$$\alpha \leq \sum_{j \in J \setminus Z_0} d_j b_j \quad (4.13)$$

and it is not necessary to check whether $\sum_{j \in J \setminus Z_0} d_j a_j \leq \alpha$ in this case.

(iv) Else if $SV \neq \emptyset$, that is, when there exists at least one variable x_j which is involved in (2.2) with $a_j = -\infty$ and $b_j = +\infty$, then $X^\pm \neq \emptyset$ and it is not necessary to check anything else in this case.

Similarly we can treat problem (C^\geq) .

With the use of results of this section, Steps 1 and 3 of Algorithm 3.1 can be modified as follows (see Algorithm 4.1), respectively.

Similarly, we can modify Steps 1 and 3 of Algorithm 3.2.

Modifications of the algorithms connected with theoretical and computational aspects do not influence their computational complexity, discussed in Section 3.3, because these modifications do not affect the “iterative” steps of algorithms.

5. Computational experiments

In this section, we present results of some computational experiments obtained by applying Algorithms 3.1 and 3.2, proposed in this paper, to problems $(C^=)$ and (C^{\geq}) . The computations were performed on an Intel Pentium II Celeron Processor 466 MHz/128 MB SDRAM IBM PC compatible. Each type of problems was run 30 times. Parameters and data were randomly generated in intervals where the functions $c_j(x_j)$ are strictly convex.

Problem 1. See Table 5.1,

$$c_j(x_j) = -\frac{s_j(x_j + c_j)}{x_j + m_j}, \quad m_j > c_j, s_j > 0, x_j > -m_j. \quad (5.1)$$

TABLE 5.1

Problem	$(C^=)$		(C^{\geq})	
Number of variables	$n = 1200$	$n = 1500$	$n = 1200$	$n = 1500$
Average number of iterations	4.07	7.1	4.1	7.17
Average run time (s)	0.0001	0.00019	0.00011	0.00021

Problem 2. See Table 5.2,

$$c_j(x_j) = \frac{1}{2}c_j \cdot (x_j - \hat{x}_j)^2, \quad c_j > 0, \hat{x}_j, j \in J, \text{ are known.} \quad (5.2)$$

Problem (2.1), (2.2), and (2.3) ((2.46), (2.47), and (2.48) respectively) with $c_j(x_j)$ defined above and $c_j = 1(> 0)$ is equivalent to projecting $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n)$ onto the feasible region (2.2)-(2.3) ((2.47)-(2.48), resp.).

TABLE 5.2

Problem	$(C^=)$		(C^{\geq})	
Number of variables	$n = 1200$	$n = 1500$	$n = 1200$	$n = 1500$
Average number of iterations	2.2	2.33	2.3	2.4
Average run time (s)	0.0001	0.00011	0.00012	0.00017

Problem 3. See Table 5.3,

$$c_j(x_j) = -s_j x_j + m_j x_j^2, \quad m_j > 0. \quad (5.3)$$

Here $c_j(x_j)$ is a strictly convex function with a minimum point at $h_j = s_j/2m_j$. We suppose that $a_j < h_j, j \in J, \alpha < \sum_{j \in J} \min\{b_j, h_j\}$.

Problem 4. See Table 5.4,

$$c_j(x_j) = \frac{1}{2} \left(\frac{h}{S} - \frac{x_j}{s_j} \right)^2, \quad s_j > 0, S \neq 0. \quad (5.4)$$

TABLE 5.3

Problem	$(C^=)$		(C^{\geq})	
Number of variables	$n = 1200$	$n = 1500$	$n = 1200$	$n = 1500$
Average number of iterations	2.4	3.2	2.5	3.33
Average run time (s)	0.00011	0.00012	0.00013	0.00015

TABLE 5.4

Problem	$(C^=)$		(C^{\geq})	
Number of variables	$n = 1200$	$n = 1500$	$n = 1200$	$n = 1500$
Average number of iterations	2.07	7.033	2.23	7.1
Average run time (s)	0.0001	0.00017	0.00012	0.00019

Problem 5. See Table 5.5,

$$c_j(x_j) = -s_j \ln(1 + m_j x_j), \quad m_j > 0, s_j > 0, x_j > -\frac{1}{m_j}. \quad (5.5)$$

TABLE 5.5

Problem	$(C^=)$		(C^{\geq})	
Number of variables	$n = 1200$	$n = 1500$	$n = 1200$	$n = 1500$
Average number of iterations	3.433	4.13	3.47	4.23
Average run time (s)	0.00009	0.00011	0.0001	0.00013

Problem 6. See Table 5.6,

$$c_j(x_j) = s_j(e^{-m_j x_j} - 1), \quad m_j > 0, s_j > 0. \quad (5.6)$$

TABLE 5.6

Problem	$(C^=)$		(C^{\geq})	
Number of variables	$n = 1200$	$n = 1500$	$n = 1200$	$n = 1500$
Average number of iterations	3.13	3.23	3.17	3.27
Average run time (s)	0.000096	0.00011	0.000101	0.00013

Problem 7. See Table 5.7,

$$c_j(x_j) = e^{k_j x_j}, \quad k_j > 0. \quad (5.7)$$

When $n < 1200$, 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.

TABLE 5.7

Problem	$(C^=)$		(C^{\geq})	
Number of variables	$n = 1200$	$n = 1500$	$n = 1200$	$n = 1500$
Average number of iterations	2.3	5.2	2.4	5.3
Average run time (s)	0.00001	0.000101	0.000012	0.000103

Similarly, we are able to consider other convex objective functions $c(\mathbf{x}) = \sum_{j \in J} c_j(x_j)$.

The effectiveness of Algorithms 3.1 and 3.2 for problems $(C^=)$ and (C^{\geq}) , respectively, 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 .

References

- [1] P. Berman, N. Kovoov, and P. M. Pardalos, *Algorithms for the least distance problem*, Complexity in Numerical Optimization (P. M. Pardalos, ed.), World Scientific, New Jersey, 1993, pp. 33–56.
- [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] J. R. Brown, *Bounded knapsack sharing*, Math. Program. Ser. A **67** (1994), no. 3, 343–382.
- [4] P. Brucker, *An $O(n)$ algorithm for quadratic knapsack problems*, Oper. Res. Lett. **3** (1984), no. 3, 163–166.
- [5] P. H. Calamai and J. J. Moré, *Quasi-Newton updates with bounds*, SIAM J. Numer. Anal. **24** (1987), no. 6, 1434–1441.
- [6] A. Charnes and W. W. Cooper, *The theory of search: optimum distribution of search effort*, Management Sci. **5** (1958), 44–50.
- [7] R. W. Cottle, S. G. Duvall, and K. Zikan, *A Lagrangean relaxation algorithm for the constrained matrix problem*, Naval Res. Logist. Quart. **33** (1986), no. 1, 55–76.
- [8] R. S. Dembo and U. Tulowitzki, *On the minimization of quadratic functions subject to box constraints*, Working Paper Series B 71, School of Organization and Management, Yale University, New Haven, 1983.
- [9] J.-P. Dussault, J. A. Ferland, and B. Lemaire, *Convex quadratic programming with one constraint and bounded variables*, Math. Program. **36** (1986), no. 1, 90–104.
- [10] J. A. Ferland, B. Lemaire, and P. Robert, *Analytic solutions for nonlinear programs with one or two equality constraints*, Publication 285, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal, 1978.
- [11] M. Held, P. Wolfe, and H. P. Crowder, *Validation of subgradient optimization*, Math. Program. **6** (1974), 62–88.
- [12] R. Helgason, J. Kennington, and H. Lall, *A polynomially bounded algorithm for a singly constrained quadratic program*, Math. Program. **18** (1980), no. 3, 338–343.
- [13] G. T. Herman and A. Lent, *A family of iterative quadratic optimization algorithms for pairs of inequalities, with application in diagnostic radiology*, Math. Programming Stud. **9** (1978), 15–29.
- [14] 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.
- [15] A. Kovačec, J. K. Merikoski, O. Pikhurko, and A. Virtanen, *Optimizers for sub-sums subject to a sum- and a Schur-convex constraint with applications to estimation of eigenvalues*, Math. Inequal. Appl. **6** (2003), no. 4, 745–763.

- [16] H. Luss and S. K. Gupta, *Allocation of effort resources among competing activities*, Oper. Res. **23** (1975), no. 2, 360–366.
- [17] R. K. McCord, *Minimization with one linear equality constraint and bounds on the variables*, Tech. Report SOL 79-20, System Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, 1979.
- [18] C. Michelot, *A finite algorithm for finding the projection of a point onto the canonical simplex of \mathbb{R}^n* , J. Optim. Theory Appl. **50** (1986), no. 1, 195–200.
- [19] J. J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming problems*, Numer. Math. **55** (1989), no. 4, 377–400.
- [20] J. J. Moré and S. A. Vavasis, *On the solution of concave knapsack problems*, Math. Program. Ser. A **49** (1991), no. 3, 397–411.
- [21] P. M. Pardalos and N. Kovoov, *An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds*, Math. Program. Ser. A **46** (1990), no. 3, 321–328.
- [22] P. M. Pardalos, Y. Ye, and C.-G. Han, *Algorithms for the solution of quadratic knapsack problems*, Linear Algebra Appl. **152** (1991), 69–91.
- [23] A. G. Robinson, N. Jiang, and C. S. Lerme, *On the continuous quadratic knapsack problem*, Math. Program. Ser. A **55** (1992), no. 1, 99–108.
- [24] 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.
- [25] ———, *Convex separable minimization subject to bounded variables*, Comput. Optim. Appl. **18** (2001), no. 1, 27–48.
- [26] ———, *Method for solving a convex integer programming problem*, Int. J. Math. Math. Sci. **2003** (2003), no. 44, 2829–2834.
- [27] ———, *Convex quadratic minimization subject to a linear constraint and box constraints*, Appl. Math. Res. Express **2004** (2004), no. 1, 17–42.
- [28] ———, *Polynomial algorithms for projecting a point onto a region defined by a linear constraint and box constraints in \mathbb{R}^n* , J. Appl. Math. **2004** (2004), no. 5, 409–431.
- [29] S. A. Vavasis, *Local minima for indefinite quadratic knapsack problems*, Math. Program. Ser. A **54** (1992), no. 2, 127–153.
- [30] P. Wolfe, *Algorithm for a least-distance programming problem*, Math. Programming Stud. **1** (1974), 190–205.
- [31] 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, Faculty of Natural Sciences and Mathematics, South-West University “Neofit Rilski,” 2700 Blagoevgrad, Bulgaria

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

Special Issue on Intelligent Computational Methods for Financial Engineering

Call for Papers

As a multidisciplinary field, financial engineering is becoming increasingly important in today's economic and financial world, especially in areas such as portfolio management, asset valuation and prediction, fraud detection, and credit risk management. For example, in a credit risk context, the recently approved Basel II guidelines advise financial institutions to build comprehensible credit risk models in order to optimize their capital allocation policy. Computational methods are being intensively studied and applied to improve the quality of the financial decisions that need to be made. Until now, computational methods and models are central to the analysis of economic and financial decisions.

However, more and more researchers have found that the financial environment is not ruled by mathematical distributions or statistical models. In such situations, some attempts have also been made to develop financial engineering models using intelligent computing approaches. For example, an artificial neural network (ANN) is a nonparametric estimation technique which does not make any distributional assumptions regarding the underlying asset. Instead, ANN approach develops a model using sets of unknown parameters and lets the optimization routine seek the best fitting parameters to obtain the desired results. The main aim of this special issue is not to merely illustrate the superior performance of a new intelligent computational method, but also to demonstrate how it can be used effectively in a financial engineering environment to improve and facilitate financial decision making. In this sense, the submissions should especially address how the results of estimated computational models (e.g., ANN, support vector machines, evolutionary algorithm, and fuzzy models) can be used to develop intelligent, easy-to-use, and/or comprehensible computational systems (e.g., decision support systems, agent-based system, and web-based systems)

This special issue will include (but not be limited to) the following topics:

- **Computational methods:** artificial intelligence, neural networks, evolutionary algorithms, fuzzy inference, hybrid learning, ensemble learning, cooperative learning, multiagent learning

- **Application fields:** asset valuation and prediction, asset allocation and portfolio selection, bankruptcy prediction, fraud detection, credit risk management
- **Implementation aspects:** decision support systems, expert systems, information systems, intelligent agents, web service, monitoring, deployment, implementation

Authors should follow the Journal of Applied Mathematics and Decision Sciences manuscript format described at the journal site <http://www.hindawi.com/journals/jamds/>. Prospective authors should submit an electronic copy of their complete manuscript through the journal Manuscript Tracking System at <http://mts.hindawi.com/>, according to the following timetable:

Manuscript Due	December 1, 2008
First Round of Reviews	March 1, 2009
Publication Date	June 1, 2009

Guest Editors

Lean Yu, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China; Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong; yulean@amss.ac.cn

Shouyang Wang, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China; sywang@amss.ac.cn

K. K. Lai, Department of Management Sciences, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong; mskklai@cityu.edu.hk