Minimizing a convex separable exponential function subject to linear equality constraint and bounded variables

Stefan M. Stefanov*

Department of Mathematics

Neofit Rilski South-Western University

2700 Blagoevgrad

Bulgaria

Abstract

In this paper, we consider the problem of minimizing a convex separable exponential function over a region defined by a linear equality constraint and bounds on the variables. Such problems are interesting from both theoretical and practical point of view because they arise in some mathematical programming problems as well as in various practical problems. Polynomial algorithms are proposed for solving problems of this form and their convergence is proved. Some examples and results of numerical experiments are also presented.

Keywords and phrases : *Exponential function, convex programming, separable programming, polynomial algorithms, computational complexity.*

1. Introduction

Consider the following convex separable program with an exponential objective function, linear equality constraint and bounded variables

$$\text{(CSE)} \qquad \min \left\{ c(\mathbf{x}) \equiv \sum_{j \in J} c_j(x_j) \equiv \sum_{j \in J} s_j (e^{-m_j x_j} - 1) \right\} \tag{1}$$

$$\text{subject to } \sum_{j \in J} d_j x_j = \alpha \tag{2}$$

$$a_j \le x_j \le b_j , \quad j \in J , \tag{3}$$

where $s_j > 0$, $m_j > 0$, $d_j > 0$, $j \in J$, $\mathbf{x} = (x_j)_{j \in J}$, and $J \stackrel{\text{def}}{=} \{1, \ldots, n\}$.

E-mail: stefm@aix.swu.bg

Journal of Interdisciplinary Mathematics
Vol. 9 (2006), No. 1, pp. 207–226
© Taru Publications

Since $c_j''(x_j) = s_j m_j^2 e^{-m_j x_j} > 0$, then $c_j(x_j)$, $j \in J$, are strictly convex functions, and since $c_j'(x_j) = -s_j m_j e^{-m_j x_j} < 0$ under the assumptions, then functions $c_j(x_j)$, $j \in J$, are decreasing.

Also, we can consider the convex exponential separable program with a linear equality constraint and bounded variables, which is similar to problem (1)-(3)

$$\text{(CESP)} \qquad \min \left\{ c(\mathbf{x}) \equiv \sum_{j \in J} c_j(x_j) \equiv \sum_{j \in J} e^{k_j x_j} \right\} \tag{4}$$

$$\text{subject to } \sum_{j \in J} d_j x_j = \alpha \tag{5}$$

$$a_j \le x_j \le b_j \,, \; j \in J \,, \tag{6}$$

where $k_j > 0$, $d_j > 0$, $j \in J$. Since $c_j''(x_j) = k_j^2 e^{k_j x_j} > 0$ then $c_j(x_j)$ are strictly convex functions, and since $c_j'(x_j) = k_j e^{k_j x_j} > 0$ under the assumptions, then functions $c_j(x_j)$, $j \in J$, are increasing.

Problems (CSE) and (CESP) are convex separable programming problems because the objective functions and constraint functions are convex and separable.

Problems (CSE) and (CESP), defined by (1)-(3) and (4)-(6), respectively, arise in production planning and scheduling, in allocation of resources, in the theory of search, in subgradient optimization, in facility location ([1], [4], [5], [6], [8], [10]), etc.

Problems like (CSE) and (CESP) and related to them are subject of intensive study. Related problems and methods for them are considered in [1]-[10].

Algorithms for resource allocation problems are proposed in [1], [4], [5], [10], and algorithms for facility location problems are suggested in [6], [8], etc. Singly constrained quadratic programs with bounded variables are considered in [2] and [3], and some separable programs are considered and methods for solving them are suggested in [7], [8], etc.

This paper is devoted to development of new efficient polynomial algorithms for solving problems (CSE) and (CESP). The paper is organized as follows. In section 2, characterization theorems (necessary and 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. Characterization theorems

2.1 *Problem* (*CSE*)

First consider problem (CSE) defined by (1)-(3).

Suppose that following assumptions are satisfied.

(1.a) $a_j \le b_j$ 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.

(1.b) $\sum_{j\in J} d_j a_j \le \alpha \le \sum_{j\in J} d_j b_j$. Otherwise the constraints (2)-(3) are inconsistent and $X = \emptyset$ where X is defined by (2)-(3).

The Lagrangian for problem (CSE) is

$$\begin{aligned} L(\mathbf{x}, \mathbf{u}, \mathbf{v}, \lambda) \; = \; & \sum_{j\in J} s_j(e^{-m_j x_j} - 1) + \lambda \left(\sum_{j\in J} d_j x_j - \alpha \right) \\ & + \sum_{j\in J} u_j(a_j - x_j) + \sum_{j\in J} v_j(x_j - b_j)\,, \end{aligned}$$

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

$$-s_j m_j e^{-m_j x_j^*} + \lambda d_j - u_j + v_j = 0\,, \quad j \in J \tag{7}$$

$$u_j(a_j - x_j^*) = 0\,, \quad j \in J \tag{8}$$

$$v_j(x_j^* - b_j) = 0\,, \quad j \in J \tag{9}$$

$$\sum_{j\in J} d_j x_j^* = \alpha \tag{10}$$

$$a_j \le x_j^* \le b_j\,, \quad j \in J \tag{11}$$

$$u_j \in \mathbb{R}^1_+,\; v_j \in \mathbb{R}^1_+\,,\; j \in J\,, \tag{12}$$

where λ, u_j, v_j, $j \in J$, are the Lagrange multipliers associated with the constraints (2), $a_j \le x_j$, $x_j \le b_j$, $j \in J$, respectively. If $a_j = -\infty$ or $b_j = +\infty$ for some j, we do not consider the corresponding condition (8) [(9)] and Lagrange multiplier u_j [v_j, respectively].

Since $u_j \geq 0$, $v_j \geq 0$, $j \in J$, and since the complementary conditions (8), (9) must be satisfied, in order to find x_j^*, $j \in J$, from system (7)-(12), 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^{2n} where $2n$ is the number of all $u_j, v_j, j \in J$, $|J| = n$. 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^*, λ, u_j, v_j, $j \in J$. Therefore the direct application of the Karush-Kuhn-Tucker (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 problems under consideration.

The following Theorem 1 gives a characterization of the optimal solution to problem (CSE). Its proof, of course, is based on the Karush-Kuhn-Tucker theorem. As we will see in section 5, by using Theorem 1 we can solve problem (CSE) with $n = 1500$ variables for a ten-thousandth of a second on a personal computer.

Theorem 1 (Characterization of the optimal solution to problem (CSE)). *A feasible solution $\mathbf{x}^* = (x_j^*)_{j \in J} \in X$ (2)-(3) is the optimal solution to problem (CSE) if and only if there exists some $\lambda \in \mathbb{R}^1$ such that*

$$x_j^* = a_j, \quad j \in J_a^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : \lambda \geq \frac{s_j m_j e^{-m_j a_j}}{d_j} \right\} \tag{13}$$

$$x_j^* = b_j, \quad j \in J_b^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : \lambda \leq \frac{s_j m_j e^{-m_j b_j}}{d_j} \right\} \tag{14}$$

$$x_j^* = \frac{\ln(s_j m_j) - \ln(\lambda d_j)}{m_j}, \qquad j \in J^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : \frac{s_j m_j e^{-m_j b_j}}{d_j} < \lambda < \frac{s_j m_j e^{-m_j a_j}}{d_j} \right\}. \tag{15}$$

We will show below that $\lambda > 0$, so that the expressions of x_j^*, $j \in J^\lambda$, in (15) (especially expressions under the sign of logarithm) are correct.

Proof. *Necessity.* Let $\mathbf{x}^* = (x_j^*)_{j \in J}$ be the optimal solution to (CSE). Then there exist constants λ, u_j, v_j, $j \in J$, such that KKT conditions (7)-(12) are satisfied.

(a) If $x_j^* = a_j$ then $u_j \geq 0$ and $v_j = 0$ according to (9). Therefore (7) implies $-s_j m_j e^{-m_j x_j^*} = u_j - \lambda d_j \geq -\lambda d_j$. Since $d_j > 0,\ j \in J$, then

$$\lambda \geq \frac{s_j m_j e^{-m_j x_j^*}}{d_j} \equiv \frac{s_j m_j e^{-m_j a_j}}{d_j}\,.$$

(b) If $x_j^* = b_j$ then $u_j = 0$ according to (8) and $v_j \geq 0$. Therefore (7) implies $-s_j m_j e^{-m_j x_j^*} = -v_j - \lambda d_j \leq -\lambda d_j$. Hence

$$\lambda \leq \frac{s_j m_j e^{-m_j x_j^*}}{d_j} \equiv \frac{s_j m_j e^{-m_j b_j}}{d_j}\,.$$

(c) If $a_j < x_j^* < b_j$ then $u_j = v_j = 0$ according to (8) and (9). Therefore (7) implies $s_j m_j e^{-m_j x_j^*} = \lambda d_j$. Hence $\lambda = \dfrac{s_j m_j e^{-m_j x_j^*}}{d_j}$, and $x_j^* = \dfrac{\ln(s_j m_j) - \ln(\lambda d_j)}{m_j}$. Since $d_j > 0,\ j \in J$, by the assumption and $b_j > x_j^*,\ x_j^* > a_j$, it follows that $\lambda = \dfrac{s_j m_j e^{-m_j x_j^*}}{d_j} < \dfrac{s_j m_j e^{-m_j a_j}}{d_j}$, $\lambda = \dfrac{s_j m_j e^{-m_j x_j^*}}{d_j} > \dfrac{s_j m_j e^{-m_j b_j}}{d_j}$, that is,

$$\frac{s_j m_j e^{-m_j b_j}}{d_j} < \lambda < \frac{s_j m_j e^{-m_j a_j}}{d_j}\,.$$

In particular, if we assume that $\lambda = 0$, since $s_j > 0,\ m_j > 0,\ d_j > 0$, then obviously $J_a^{\lambda=0} = J^{\lambda=0} = \emptyset$ and $J = J_b^{\lambda=0}$. Similarly, if we assume that $\lambda < 0$, since $s_j > 0,\ m_j > 0,\ d_j > 0$, then $J_a^\lambda = J^\lambda = \emptyset$ and $J = J_b^\lambda$.

To describe cases (a), (b), (c), it is convenient to introduce the index sets $J_a^\lambda, J_b^\lambda, J^\lambda$ defined by (13), (14) and (15), respectively. It is obvious that $J_a^\lambda \cup J_b^\lambda \cup J^\lambda = J$. The 'necessity' part is proved.

Sufficiency. Conversely, let $\mathbf{x}^* \in X$ and components of $\mathbf{x}^*$ satisfy (13), (14) and (15), where $\lambda \in \mathbb{R}^1$.

Set:

$$\lambda = \frac{s_j m_j e^{-m_j x_j^*}}{d_j} \text{ obtained from } \sum_{j\in J_a^\lambda} d_j a_j + \sum_{j\in J_b^\lambda} d_j b_j + \sum_{j\in J^\lambda} d_j \frac{\ln(s_j m_j) - \ln(\lambda d_j)}{m_j} = \alpha;$$

$u_j = v_j = 0 \quad \text{for } \ j \in J^\lambda;$

$u_j \;=\; -s_j m_j e^{-m_j a_j} + \lambda d_j$

$(\geq 0$ according to the definition of $J_a^\lambda)$, $v_j = 0$ for $j \in J_a^\lambda$;

$u_j \;=\; 0\,,\; v_j = s_j m_j e^{-m_j b_j} - \lambda d_j$

$(\geq 0$ according to the definition of $J_b^\lambda)$ for $j \in J_b^\lambda$.

By using these expressions, it is easy to check that conditions (7), (8), (9), (12) are satisfied; conditions (10) and (11) are also satisfied according to the assumption $\mathbf{x}^* \in X$.

We have proved that $x_j^*, \lambda, u_j, v_j,\; j \in J$, satisfy KKT conditions (7)-(12) 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 (CSE), and since $c(\mathbf{x})$ is strictly convex then this optimal solution is unique. □

In view of the discussion above, the importance of Theorem 1 consists in the fact that it describes components of the optimal solution to problem (CSE) only through the Lagrange multiplier λ associated with the equality constraint (2).

Since we do not know the optimal value of λ from Theorem 1, we define an iterative process with respect to the Lagrange multiplier λ and we prove convergence of this process in section 3 *The algorithms*.

From $d_j > 0$, $s_j > 0$, $m_j > 0$ and $a_j \leq b_j$, $j \in J$, it follows that

$$ub_j \stackrel{\text{def}}{=} \frac{s_j m_j e^{-m_j b_j}}{d_j} \leq \frac{s_j m_j e^{-m_j a_j}}{d_j} \stackrel{\text{def}}{=} la_j\,, \quad j \in J$$

for the expressions by which we define the sets $J_a^\lambda, J_b^\lambda, J^\lambda$.

The problem how to ensure a feasible solution to problem (CSE), which is an assumption of Theorem 1, is discussed in subsection 3.3.

2.2 *Problem* (*CESP*)

Consider the convex exponential separable program with a linear equality constraint and box constraints (CESP) (4)-(6).

Assumptions:

(2.a) $a_j \leq b_j$ for all $j \in J$.

(2.b) $\sum_{j\in J} d_j a_j \;\leq\; \alpha \;\leq\; \sum_{j\in J} d_j b_j$. Otherwise the constraints (5)-(6) are inconsistent and the feasible region X, defined by (5)-(6), is empty.

The KKT conditions for problem (CESP) are

$$k_j e^{k_j x_j^*} + \lambda d_j - u_j + v_j = 0, \quad j \in J$$

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

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

$$\sum_{j\in J} d_j x_j^* = \alpha$$

$$a_j \le x_j^* \le b_j\,, \quad j \in J$$

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

In this case, the following Theorem 2, which is similar to Theorem 1, holds true.

Theorem 2 (Characterization of the optimal solution to problem (CESP)). *A feasible solution* $\mathbf{x}^* = (x_j^*)_{j\in J} \in X$ (5)-(6) *is the optimal solution to problem (CESP) if and only if there exists some* $\lambda \in \mathbb{R}^1$ *such that*

$$x_j^* = a_j\,, \quad j \in J_a^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : \lambda \ge -\frac{k_j e^{k_j a_j}}{d_j} \right\} \tag{16}$$

$$x_j^* = b_j\,, \quad j \in J_b^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : \lambda \le -\frac{k_j e^{k_j b_j}}{d_j} \right\} \tag{17}$$

$$x_j^* = \frac{1}{k_j} \ln \left(-\frac{\lambda d_j}{k_j} \right), \qquad j \in J^\lambda \stackrel{\text{def}}{=} \left\{ j \in J : -\frac{k_j e^{k_j b_j}}{d_j} < \lambda < -\frac{k_j e^{k_j a_j}}{d_j} \right\}. \tag{18}$$

As we will show below, $\lambda < 0$, so that the expressions of x_j^*, $j \in J^\lambda$, in (18) (especially expressions under the sign of logarithm) are correct.

The proof of Theorem 2 is omitted because it is similar to that of Theorem 1.

3. The algorithms

3.1 *Analysis of the optimal solution to problem* (*CSE*)

Before the formal statement of the algorithm for problem (CSE), we discuss some properties of the optimal solution to this problem which turn out to be useful.

Using (13), (14) and (15), condition (10) can be written as follows:

$$\sum_{j\in J_a^\lambda} d_j a_j + \sum_{j\in J_b^\lambda} d_j b_j + \sum_{j\in J^\lambda} d_j \frac{\ln(s_j m_j) - \ln(\lambda d_j)}{m_j} = \alpha\,. \tag{10'}$$

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

$$x_j^* = x_j(\lambda) = \begin{cases} a_j\,, & j \in J_a^\lambda \\ b_j\,, & j \in J_b^\lambda \\ \dfrac{\ln(s_j m_j) - \ln(\lambda d_j)}{m_j}\,, & j \in J^\lambda\,. \end{cases} \tag{19}$$

Functions $x_j(\lambda)$, $j \in J$, are piecewise linear, monotone nonincreasing, piecewise differentiable functions of λ with two breakpoints at $\lambda = \dfrac{s_j m_j e^{-m_j a_j}}{d_j}$ and $\lambda = \dfrac{s_j m_j e^{-m_j b_j}}{d_j}$.

Let

$$\delta(\lambda) \stackrel{\text{def}}{=} \sum_{j\in J_a^\lambda} d_j a_j + \sum_{j\in J_b^\lambda} d_j b_j + \sum_{j\in J^\lambda} d_j \frac{\ln(s_j m_j) - \ln(\lambda d_j)}{m_j} - \alpha\,. \tag{20}$$

If we differentiate $\delta(\lambda)$ with respect to λ, we get

$$\delta'(\lambda) \equiv -\frac{1}{\lambda} \sum_{j\in J^\lambda} \frac{d_j}{m_j} < 0\,, \tag{21}$$

according to the remark (after statement of Theorem 1), that $\lambda > 0$, when $J^\lambda \neq \emptyset$, and $\delta'(\lambda) = 0$ when $J^\lambda = \emptyset$. Hence $\delta(\lambda)$ is a *monotone nonincreasing* function of $\lambda \in \mathbb{R}^1$.

From the equation $\delta(\lambda) = 0$, where $\delta(\lambda)$ is defined by (20), we are able to obtain a closed form expression for λ

$$\lambda = \exp\left\{ \left[\sum_{j\in J^\lambda} \frac{d_j}{m_j} \right]^{-1} \left[\sum_{j\in J_a^\lambda} d_j a_j + \sum_{j\in J_b^\lambda} d_j b_j + \sum_{j\in J^\lambda} \frac{d_j}{m_j} \ln \frac{s_j m_j}{d_j} - \alpha \right] \right\}, \tag{22}$$

because $\delta'(\lambda) < 0$ according to (21) when $J^\lambda \neq \emptyset$ (it is important that $\delta'(\lambda) \neq 0$). This expression of λ shows that $\lambda > 0$, and it is used in the algorithm suggested for problem (CSE). It turns out that without loss of generality we can assume that $\delta'(\lambda) \neq 0$, that is, $\delta(\lambda)$ depends on λ, which means that $J^\lambda \neq \emptyset$.

At iteration k of the implementation of the algorithms, denote by

$\lambda^{(k)}$ the value of Lagrange multiplier associated with constraint (2) [(5), respectively], by $\alpha^{(k)}$ the right-hand side of (2) [of (5), respectively]; by $J^{(k)}$, $J_\alpha^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ the current sets $J, J_a^\lambda, J_b^\lambda, J^\lambda$, respectively.

3.2 *Algorithm* 1 (*for problem* (*CSE*))

The following algorithm for solving problem (CSE) is based on Theorem 1.

Algorithm 1 (*for problem* (*CSE*))

1. (*Initialization*) $J := \{1, \ldots, n\}$, $k := 0$, $\alpha^{(0)} := \alpha$, $n^{(0)} := n$, $J^{(0)} := J$, $J_a^\lambda := \emptyset$, $J_b^\lambda := \emptyset$.
   If $\sum_{j \in J} d_j a_j \le \alpha \le \sum_{j \in J} d_j b_j$, go to 2 else go to 9.
2. $J^{\lambda(k)} := J^{(k)}$. Calculate $\lambda^{(k)}$ by using the explicit expression (22) of λ. Go to 3.
3. Construct the sets $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ through (13), (14), (15) (with $j \in J^{(k)}$ instead of $j \in J$) and find their cardinal numbers $|J_a^{\lambda(k)}|$, $|J_b^{\lambda(k)}|$, $|J^{\lambda(k)}|$, respectively. Go to 4.
4. Calculate
$$\delta(\lambda^{(k)}) := \sum_{j \in J_a^{\lambda(k)}} d_j a_j + \sum_{j \in J_b^{\lambda(k)}} d_j b_j + \sum_{j \in J^{\lambda(k)}} d_j \frac{\ln(s_j m_j) - \ln(\lambda^{(k)} d_j)}{m_j} - \alpha^{(k)}.$$
   Go to 5.
5. If $\delta(\lambda^{(k)}) = 0$ or $J^{\lambda(k)} = \emptyset$ 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 $j \in J_b^\lambda$; $x_j^* := \dfrac{\ln(s_j m_j) - \ln(\lambda d_j)}{m_j}$ for $j \in J^\lambda$. Go to 10.

9. Problem (CSE) has no optimal solution because the feasible set X (2)-(3) is empty.
10. End.

3.3 *Convergence and complexity of Algorithm* 1

The following Theorem 3 states convergence of Algorithm 1.

Theorem 3. *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.

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

$$\begin{aligned} \sum_{j \in J^{\lambda(k+1)}} d_j x_j^{(k)} &\equiv \sum_{j \in J^{(k+1)}} d_j x_j^{(k)} = \sum_{j \in J^{(k)} \setminus J_a^{\lambda(k)}} d_j x_j^{(k)} \\ &= \alpha^{(k)} - \sum_{j \in J_a^{\lambda(k)}} d_j x_j^{(k)} . \end{aligned} \tag{23}$$

Let $j \in J_a^{\lambda(k)}$. According to definition (13) of $J_a^{\lambda(k)}$ we have

$$\frac{s_j m_j e^{-m_j a_j}}{d_j} \le \lambda^{(k)} = \frac{s_j m_j e^{-m_j x_j^{(k)}}}{d_j} .$$

Multiplying this inequality by $\dfrac{d_j}{s_j m_j e^{-m_j a_j}} > 0$ we obtain $1 \le \dfrac{e^{-m_j x_j^{(k)}}}{e^{-m_j a_j}} \equiv e^{m_j(a_j - x_j^{(k)})}$. Therefore $x_j^{(k)} \le a_j$ because $m_j > 0$ and according to properties of the exponential function.

From (23), using that $d_j > 0$, $a_j \ge x_j^{(k)}$, $j \in J_a^{\lambda(k)}$, and Step 6, we get

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

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

$$\lambda^{(k)} = \frac{s_{j_0} m_{j_0} e^{-m_{j_0} x_{j_0}^{(k)}}}{d_{j_0}} \leq \frac{s_{j_0} m_{j_0} e^{-m_{j_0} x_{j_0}^{(k+1)}}}{d_{j_0}} = \lambda^{(k+1)}.$$

We have used that the relationship between $\lambda^{(k)}$ and $x_j^{(k)}$ is given by (15) for $j \in J^{\lambda(k)}$ according to Step 2 of Algorithm 1, and $d_j > 0,\ s_j > 0,\ m_j > 0,\ j \in J$.

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 (3). From

$$\frac{s_j m_j e^{-m_j b_j}}{d_j} < \lambda \equiv \frac{s_j m_j e^{-m_j x_j^*}}{d_j} < \frac{s_j m_j e^{-m_j a_j}}{d_j}, \quad j \in J^\lambda,$$

and $d_j > 0,\ s_j > 0,\ m_j > 0,\ j \in J$, it follows that $a_j < x_j^* < b_j$ for $j \in J^\lambda$. Hence all $x_j^*,\ j \in J$, satisfy (3).

Since at each iteration $\lambda^{(k)}$ is determined from the current equality constraint (2) (Step 2 of Algorithm 1) and since $x_j^*,\ j \in J$, are determined in accordance with $\lambda^{(k)}$ at each iteration (Steps 5, 6, 7, 8 of Algorithm 1) then $\mathbf{x}^*$ satisfies (2) as well.

Therefore Algorithm 1 generates $\mathbf{x}^*$ which is feasible for problem (CSE), which is an assumption of Theorem 1.

Remark 1. Theorem 3, definitions of J_a^λ (13), J_b^λ (14) and J^λ (15), and Steps 6, 7 and 8 of Algorithm 1 allow us to state 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 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 1.

At each iteration of Algorithm 1, we determine the value of at least one variable (Steps 6, 7, 8) and at each iteration we solve a problem of the form (CSE) 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 1 (initialization and checking whether X is empty) 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 $\mathcal{O}(n)$ because at these steps we assign some of x_j the final 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 1 has $\mathcal{O}(n^2)$ 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 (13), (14), (15). In practice, Algorithm 1 has $\mathcal{O}(n)$ running time.

3.4 *Algorithm* 2 (*for problem* (*CESP*)) *and its convergence*

After analysis of the optimal solution to problem (CESP), similar to that to problem (CSE), we suggest the following algorithm for solving problem (CESP).

Algorithm 2 (*for problem* (*CESP*))

1. (*Initialization*) $J := \{1, \ldots, n\}$, $k := 0$, $\alpha^{(0)} := \alpha$, $n^{(0)} := n$, $J^{(0)} := J$, $J_a^\lambda := \emptyset$, $J_b^\lambda := \emptyset$. If $\sum_{j \in J} d_j a_j \le \alpha \le \sum_{j \in J} d_j b_j$, go to 2 else go to 9.
2. $J^{\lambda(k)} := J^{(k)}$. Calculate $\lambda^{(k)}$ by using the explicit expression
$$\lambda^{(k)} = -\exp\left\{ \left[\sum_{j \in J^{\lambda(k)}} \frac{d_j}{k_j} \right]^{-1} \left[\alpha^{(k)} - \sum_{j \in J^{\lambda(k)}} \frac{d_j}{k_j} \ln \frac{d_j}{k_j} \right] \right\} \quad (< 0)\,.$$
Go to 3.
3. Construct the sets $J_a^{\lambda(k)}$, $J_b^{\lambda(k)}$, $J^{\lambda(k)}$ through (16), (17), (18) (with $j \in J^{(k)}$ instead of $j \in J$) and find their cardinal numbers $|J_a^{\lambda(k)}|$, $|J_b^{\lambda(k)}|$, $|J^{\lambda(k)}|$. Go to 4.

4. Calculate

$$\delta(\lambda^{(k)}) := \sum_{j\in J_a^{\lambda(k)}} d_j a_j + \sum_{j\in J_b^{\lambda(k)}} d_j b_j + \ln(-\lambda^{(k)}) \sum_{j\in J^{\lambda(k)}} \frac{d_j}{k_j} + \sum_{j\in J^{\lambda(k)}} \frac{d_j(\ln d_j - \ln k_j)}{k_j} - \alpha^{(k)} .$$

   Go to 5.

5. If $\delta(\lambda^{(k)}) = 0$ or $J^{\lambda(k)} = \emptyset$ 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 $j \in J_b^\lambda$; $x_j^* := \dfrac{1}{k_j} \ln\left(-\dfrac{\lambda d_j}{k_j}\right)$ for $j \in J^\lambda$. Go to 10.

9. Problem (CESP) has no optimal solution because the feasible set X (5)-(6) is empty.

10. End.

To avoid a possible 'endless loop' in programming Algorithms 1 and 2, 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 (given or chosen) tolerance value up to which the equality $\delta(\lambda) = 0$ must be satisfied.

A theorem analogous to Theorem 3 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 $\lambda, J^\lambda, J_a^\lambda, J_b^\lambda$, respectively.

Theorem 4. *Let $\{\lambda^{(k)}\}$ be the sequence generated by Algorithm 2. Then*

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

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

The proof of Theorem 4 is omitted because it is similar to that of Theorem 3.

It can be proved that Algorithm 2 has $\mathcal{O}(n^2)$ running time, and point $\mathbf{x}^* = (x_j^*)_{j \in J}$ generated by this algorithm is feasible for problem (CESP), which is an assumption of Theorem 2.

4. Extensions

4.1 *Theoretical aspects*

Up to now we required $d_j > 0,\ j \in J$, in (2) and (5) of problems (CSE) and (CESP), respectively. However, if it is allowed $d_j = 0$ for some j in problems (CSE) and (CESP) then for such indices j we cannot construct the expressions $\dfrac{s_j m_j e^{-m_j a_j}}{d_j}$ and $\dfrac{s_j m_j e^{-m_j b_j}}{d_j}$ for problem (CSE), and $-\dfrac{k_j e^{k_j a_j}}{d_j}$ and $-\dfrac{k_j e^{k_j b_j}}{d_j}$ for problem (CESP), 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 (2) [in (5), respectively] for such indices j. It turns out that we can cope with this difficulty and solve problems (CSE) and (CESP) with $d_j = 0$ for some j's.

Denote

$$Z0 = \{ j \in J : d_j = 0 \} .$$

Here '0' means the 'computer zero'. In particular, when $J = Z0$ and $\alpha = 0$ then the set X is defined only by (3) (by (6), respectively).

Theorem 5 (Characterization of the optimal solution to problem (CSE): an extended version). *Problem* (CSE) *can be decomposed into two subproblems:* (CSE1) *for* $j \in Z0$ *and* (CSE2) *for* $j \in J \setminus Z0$.

The optimal solution to (CSE1) *is*

$$x_j^* = b_j , \quad j \in Z0 , \tag{24}$$

that is, subproblem (CSE1) *itself is decomposed into* $n_0 \equiv |Z0|$ *independent problems. The optimal solution to* (CSE2) *is given by* (13), (14), (15) *with* $J := J \setminus Z0$.

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

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

$$-s_j m_j e^{-m_j x_j^*} - u_j + v_j = 0\,, \quad j \in Z0 \text{ from (7)} \tag{7'}$$

and (8)-(12).

(a) If $x_j^* = a_j$, then $u_j \geq 0$, $v_j = 0$. From $(7')$ it follows that $-s_j m_j e^{-m_j x_j^*} = u_j \geq 0$, which is impossible because $s_j > 0$, $m_j > 0$ and $e^{-m_j x_j^*} > 0$.

(b) If $x_j^* = b_j$, then $u_j = 0$, $v_j \geq 0$. Therefore $-s_j m_j e^{-m_j x_j^*} = -v_j \leq 0$, which is always satisfied for $s_j > 0$, $m_j > 0$.

(c) If $a_j < x_j^* < b_j$, then $u_j = v_j = 0$. Therefore $-s_j m_j e^{-m_j x_j^*} = 0$, that is, $s_j m_j = 0$, which is impossible according to the assumption $s_j > 0$, $m_j > 0$.

As we have observed, only case (b) is possible for $j \in Z0$, and $x_j^* = b_j$, $j \in Z0$.

(2) Components of the optimal solution to (CSE2) are obtained by using the same approach as that of the proof of 'necessity' part of Theorem 1, but with the reduced index set $J := J \setminus Z0$.

Sufficiency. Conversely, let $\mathbf{x}^* \in X$ and components of $\mathbf{x}^*$ satisfy: (24) for $j \in Z0$, and (13), (14), (15) with $J := J \setminus Z0$. Set:

$$u_j = 0,\ v_j = s_j m_j e^{-m_j b_j} \quad (> 0) \quad \text{for } j \in Z0\,.$$

If $\lambda \neq 0$, set:

$$\lambda = \frac{s_j m_j e^{-m_j x_j^*}}{d_j} = \lambda(\mathbf{x}^*) \qquad \text{from (15);}$$

$$u_j = v_j = 0 \qquad \text{for } a_j < x_j^* < b_j,\ j \in J \setminus Z0;$$

$$u_j = -s_j m_j e^{-m_j a_j} + \lambda d_j \ (\geq 0),\ v_j = 0 \qquad \text{for } x_j^* = a_j, j \in J \setminus Z0;$$

$$u_j = 0\,,\ v_j = s_j m_j e^{-m_j b_j} - \lambda d_j \quad (\geq 0) \qquad \text{for } x_j^* = b_j, j \in J \setminus Z0\,.$$

As in the proof of Theorem 1, $J_a^{\lambda=0} = J^{\lambda=0} = \emptyset$.

It can be verified that $\mathbf{x}^*$, λ, u_j, v_j, $j \in J$, satisfy the KKT conditions $(7')$-(12). Then $\mathbf{x}^*$ with components: (24) for $j \in Z0$, and (13), (14), (15) for $J := J \setminus Z0$ is the optimal solution to problem $(\text{CSE}) = (\text{CSE1}) \cup (\text{CSE2})$. □

An analogous result holds for problem (CESP).

Theorem 6 (Characterization of the optimal solution to problem (CESP): an extended version). *Problem* (CESP) *can be decomposed into two subproblems:* (CESP1) *for* $j \in Z0$ *and* (CESP2) *for* $j \in J \setminus Z0$.

The optimal solution to (CESP1) *is* $x_j^* = a_j,\ j \in Z0$. *The optimal solution to* (CESP2) *is given by* (16), (17), (18) *with* $J := J \setminus Z0$.

The proof of Theorem 6 is omitted because it repeats in part the proofs of Theorem 1 and Theorem 5.

Thus, with the use of Theorem 5 and Theorem 6 we can express components of the optimal solutions to problems (CSE) and (CESP) without the necessity of constructing the expressions $\dfrac{s_j m_j e^{-m_j a_j}}{d_j}$, $\dfrac{s_j m_j e^{-m_j b_j}}{d_j}$, $-\dfrac{k_j e^{k_j a_j}}{d_j}$ and $-\dfrac{k_j e^{k_j b_j}}{d_j}$ with $d_j = 0$.

4.2 *Computational aspects*

Algorithms 1 and 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 the first step of the algorithms to check whether the corresponding feasible region is empty or nonempty and at Step 3 in the expressions $\dfrac{s_j m_j e^{-m_j x_j}}{d_j}$ and $-\dfrac{k_j e^{k_j x_j}}{d_j}$ 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 above discussion, it is convenient to do the following.

Construct the sets of indices:

$$\begin{aligned} SVN &= \{j \in J \setminus Z0 : a_j > -\infty,\ b_j < +\infty\}, \\ SV1 &= \{j \in J \setminus Z0 : a_j > -\infty,\ b_j = +\infty\}, \\ SV2 &= \{j \in J \setminus Z0 : a_j = -\infty,\ b_j < +\infty\}, \\ SV &= \{j \in J \setminus Z0 : a_j = -\infty,\ b_j = +\infty\}. \end{aligned} \tag{25}$$

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 $J_a^\lambda, J_b^\lambda, J^\lambda$ without the necessity of calculating the values $\dfrac{s_j m_j e^{-m_j x_j}}{d_j}$ (for problem (CSE)) 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:

$$\begin{aligned} J^{\lambda\, SVN} &= \left\{ j \in SVN : \frac{s_j m_j e^{-m_j b_j}}{d_j} < \lambda < \frac{s_j m_j e^{-m_j a_j}}{d_j} \right\}, \\ J_a^{\lambda\, SVN} &= \left\{ j \in SVN : \lambda \geq \frac{s_j m_j e^{-m_j a_j}}{d_j} \right\}, \\ J_b^{\lambda\, SVN} &= \left\{ j \in SVN : \lambda \leq \frac{s_j m_j e^{-m_j b_j}}{d_j} \right\}; \end{aligned}$$

for $SV1$:

$$\begin{aligned} J^{\lambda\, SV1} &= \left\{ j \in SV1 : \lambda < \frac{s_j m_j e^{-m_j a_j}}{d_j} \right\}, \\ J_a^{\lambda\, SV1} &= \left\{ j \in SV1 : \lambda \geq \frac{s_j m_j e^{-m_j a_j}}{d_j} \right\}; \end{aligned} \tag{26}$$

for $SV2$:

$$\begin{aligned} J^{\lambda\, SV2} &= \left\{ j \in SV2 : \lambda > \frac{s_j m_j e^{-m_j b_j}}{d_j} \right\}, \\ J_b^{\lambda\, SV2} &= \left\{ j \in SV2 : \lambda \leq \frac{s_j m_j e^{-m_j b_j}}{d_j} \right\}; \end{aligned}$$

for SV:

$$J^{\lambda\, SV} = SV\,.$$

Then:

$$\begin{aligned} J^\lambda &:= J^{\lambda\, SVN} \cup J^{\lambda\, SV1} \cup J^{\lambda\, SV2} \cup J^{\lambda\, SV}, \\ J_a^\lambda &:= J_a^{\lambda\, SVN} \cup J_a^{\lambda\, SV1}, \\ J_b^\lambda &:= J_b^{\lambda\, SVN} \cup J_b^{\lambda\, SV2}. \end{aligned} \tag{27}$$

We use the sets $J^\lambda, J_a^\lambda, J_b^\lambda$ (27) as the corresponding sets with the same names in Algorithms 1 and 2.

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

About Algorithm 1.

Step 1^1. (*Initialization*) $J := \{1, \dots, n\}$, $k := 0$, $\alpha^{(0)} := \alpha$, $n^{(0)} := n$, $J^{(0)} := J$, $J_a^\lambda := \emptyset$, $J_b^\lambda := \emptyset$.

Construct the set $Z0$. If $j \in Z0$ then $x_j^* := b_j$.

Set $J := J \setminus Z0$, $J^{(0)} := J$, $n^{(0)} := n - |Z0|$.

Construct the sets SVN, $SV1$, $SV2$, SV.

If $SVN = J$ then

if $\sum_{j \in J} d_j a_j \le \alpha \le \sum_{j \in J} d_j b_j$ then go to Step 2

else go to Step 9 (feasible region X is empty)

else if $SV1 \cup SVN = J$ then

if $\sum_{j \in J} d_j a_j \le \alpha$ then go to Step 2

else go to Step 9 (feasible region X is empty)

else if $SV2 \cup SVN = J$ then

if $\alpha \le \sum_{j \in J} d_j b_j$ then go to Step 2

else go to Step 9 (feasible region X is empty)

else if $SV \neq \emptyset$ then go to Step 2 (feasible region is always nonempty).

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 (27) and find their cardinal numbers $|J_a^{\lambda(k)}|$, $|J_b^{\lambda(k)}|$, $|J^{\lambda(k)}|$, respectively. Go to Step 4.

Similarly, we can define subsidiary index sets of the form (26) for problem (CESP) as well and modify Steps 1 and 3 of Algorithm 2.

Modifications of the algorithms connected with theoretical and computational aspects do not influence upon their computational complexity, discussed in section 3, because these modifications do not affect the 'iterative' steps of algorithms.

5. Computational experiments

In this section we present results 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/128MB SDRAM IBM PC compatible. Each problem was run 30 times. Coefficients $s_j > 0$, $m_j > 0$, $d_j > 0$, $j \in J$, for problem (CSE) and $k_j > 0$, $d_j > 0$, $j \in J$, for problem (CESP) were randomly generated.

Problem	(CSE)		(CESP)	
Number of variables	$n = 1200$	$n = 1500$	$n = 1200$	$n = 1500$
Average number of iterations	3.03	3.13	2.07	5.10
Average run time (in seconds)	0.000096	0.00011	0.00001	0.000101

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.

The effectiveness of algorithms for problems (CSE) and (CESP) 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] G. R. Bitran and A. C. Hax, Disaggregation and resource allocation using convex knapsack problems with bounded variables, *Management Science*, Vol. 27 (1981), pp. 431–441.

[2] J.-P. Dussault, J. Ferland and B. Lemaire, Convex quadratic programming with one constraint and bounded variables, *Mathematical Programming*, Vol. 36 (1986), pp. 90–104.

[3] R. Helgason, J. Kennington and H. Lall, A polynomially bounded algorithm for a singly constrained quadratic program, *Mathematical Programming*, Vol. 18 (1980), pp. 338–343.

[4] N. Katoh, T. Ibaraki and H. Mine, A polynomial time algorithm for the resource allocation problem with a convex objective function, *Journal of the Operations Research Society*, Vol. 30 (1979), pp. 449–455.

[5] H. Luss and S. K. Gupta, Allocation of effort resources among competing activities, *Operations Research*, Vol. 23 (1975), pp. 360–366.

[6] S. M. Stefanov, On the implementation of stochastic quasigradient methods to some facility location problems, *Yugoslav Journal of Operations Research*, Vol. 10 (2) (2000), pp. 235–256.

[7] S. M. Stefanov, Convex separable minimization subject to bounded variables, *Computational Optimization and Applications. An International Journal*, Vol. 18 (1) (2001), pp. 27–48.

[8] S. M. Stefanov, *Separable Programming. Theory and Methods*, Kluwer Academic Publishers, Dordrecht - Boston - London, 2001.

[9] S. M. Stefanov, Convex separable minimization problems with a linear constraint and bounds on the variables, in *Applications of Mathematics in Engineering and Economics*, Vol. 27, D. Ivanchev and M. D. Todorov (eds.), Heron Press, Sofia, 2002, 392–402.

[10] P.H. Zipkin, Simple ranking methods for allocation of one resource, *Management Science*, Vol. 26 (1980), pp. 34–43.

Received April, 2005