Yugoslav Journal of Operations Research
10 (2000), Number 2, 235-256

ON THE IMPLEMENTATION OF STOCHASTIC QUASIGRADIENT METHODS TO SOME FACILITY LOCATION PROBLEMS

Stefan M. STEFANOV

Department of Mathematics
Neofit Rilski University
Blagoevgrad, Bulgaria

Abstract: In this paper we consider the facility location problem in a stochastic environment. After a brief description of stochastic quasigradient methods (SQM) for solving stochastic programming problems, algorithms of polynomial complexity are suggested for projecting the current approximation, generated by the SQM, onto feasible sets of two important facility location problems. The convergence of suggested algorithms is proved, and some examples are given.

Keywords: Stochastic programming, stochastic quasigradient methods, facility location, projection, algorithms.

1. INTRODUCTION

Consider the following simple facility location model in a stochastic environment ([3]). Determine the amounts x_j of facilities at points j, $j = 1,\dots,n$ in order to meet the demand w_j. Since the demand $\mathbf{w} = (w_1,\dots,w_n)$ is random, we know only its distribution function $H(\overline{\mathbf{w}}) = P\{w_1 \le \overline{w_1},\dots,w_n \le \overline{w_n}\}$. At the moment of decision making concerning $\mathbf{x} = (x_1,\dots,x_n)$, the actual value of the demand $\mathbf{w} = (w_1,\dots,w_n)$ is not known.

Suppose that we have made a decision $\mathbf{x}$ about the quantities of materials, facilities, etc., and that the actual demand turned out to be $\mathbf{w}$. We have to pay for both oversupply and shortfalls. The penalty charged at the j-th location is $y_1^j(w_j - x_j)$, if

$w_j \geq x_j$, and $y_2^j(x_j - w_j)$ if $w_j < x_j$, where the functions y_1^j and y_2^j are nondecreasing. In the simplest case these functions are linear and the total penalty is

$$\sum_{j=1}^{n} \max\{ p_j(w_j - x_j), q_j(x_j - w_j)\} \tag{1}$$

where $p_j \geq 0$ and $q_j \geq 0$ are the expenses for storage and losses because of deficit for unit of the j-th facility, $j = 1, \dots, n$.

In most cases $\mathbf{x}$ should be determined such that the average penalty is minimal, that is, to minimize the following function

$$F(\mathbf{x}) = \mathbf{E_w} f(\mathbf{x}, \mathbf{w}) = \mathbf{E_w}\{\sum_{j=1}^{n} (p_j \int_0^{x_j} (x_j - w_j)\mathbf{P}(dw_j) + q_j \int_{x_j}^{\infty} (w_j - x_j)\mathbf{P}(dw_j))\},$$

where $\mathbf{E_w}$ denotes the mathematical expectation with respect to $\mathbf{w}$. Often there are some constraints on $\mathbf{x}$.

If the volume of the store we have to use to keep the facilities is a and we have to order a quantity of the j-th product which is at least a_j and at most b_j, $j = 1, \dots, n$, we obtain the following minimization problem:

Find $x = (x_1, \dots, x_n)$ such that

$$F(\mathbf{x}) = \mathbf{E_w} f(\mathbf{x}, \mathbf{w}) \to \min_{\mathbf{x}} \tag{2}$$

subject to

$$x \in X \tag{3}$$

where X is defined through

$$\sum_{j=1}^{n} d_j x_j \leq a, \quad d_j > 0, \ j = 1, \dots, n \tag{4}$$

$$a_j \leq x_j \leq b_j, \quad j = 1, \dots, n \tag{5}$$

or

$$\sum_{j=1}^{n} d_j x_j = a, \quad d_j > 0, \ j = 1, \dots, n \tag{6}$$

$$a_j \leq x_j \leq b_j, \quad j = 1, \dots, n. \tag{7}$$

Relations "≤" and "=" in (4) and (6) mean "store may not be completely filled" and "store must be completely filled", respectively.

Here w_j are random variables in closed segments $[R1_j, R2_j]$, $j = 1, \dots, n$, respectively. The function $F(x)$ in (2) can be written in the following form

$$F(\mathbf{x}) = \mathbf{E_w} f(\mathbf{x}, \mathbf{w}) = \mathbf{E_w} \sum_{j=1}^{n} f_j(x_j, w_j) \tag{2'}$$

where

$$f_j(x_j, w_j) = \begin{cases} p_j(x_j - w_j), & \text{if } x_j \geq w_j \\ q_j(w_j - x_j), & \text{if } x_j < w_j. \end{cases}$$

Since $f(\mathbf{x}, \mathbf{w})$ is nondifferentiable at $\mathbf{x} = \mathbf{w}$ then $F(x)$ is also a nondifferentiable function.

Problem (2) - (3) is known as the multi-commodity facility location problem or as the inventory control problem and it is a special case of a (perspective) stochastic programming problem, that is, a problem of the form:

Find $x = (x_1, \dots, x_n)$ such that

$$F^0(\mathbf{x}) \equiv \mathbf{E_w} f^0(\mathbf{x}, \mathbf{w}) = \int f^0(\mathbf{x}, \mathbf{w}) \mathbf{P}(d\mathbf{w}) \to \min_{\mathbf{x}}$$

subject to

$$F^i(\mathbf{x}) \equiv \mathbf{E_w} f^i(\mathbf{x}, \mathbf{w}) = \int f^i(\mathbf{x}, \mathbf{w}) \mathbf{P}(d\mathbf{w}) \leq 0, \quad i = 1, \dots, m$$

$$x \in X \subset R^n.$$

The functions $F^i(\mathbf{x})$, $i = 0, 1, \dots, m$ are called regression functions.

Stochastic quasigradient methods (SQM) for solving stochastic optimization problems were suggested by Yu. Ermoliev ([1], [2], [3]).

Given the problem

$$\min F(\mathbf{x})$$

subject to

$$\mathbf{x} \in X$$

where X is a "deterministic" set.

SQM are defined through

$$\mathbf{x}^{k+1} = \prod_X (\mathbf{x}^k - r_k \boldsymbol{x}^k), \quad k = 0,1,\ldots, \tag{8}$$

where

- $\mathbf{x}^0$ is an arbitrary initial guess (initial approximation);
- $\prod_X(\mathbf{y})$ is a projection operation of y onto the feasible region X;
- r_k is a step size;
- $\boldsymbol{x}^k = \boldsymbol{x}^k(\mathbf{w})$ is a step direction, $\boldsymbol{x}^k(\mathbf{w})$ is a random vector such that

$$\mathbf{E}_{\mathbf{w}}(\boldsymbol{x}^k / \mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^k) = a_k \hat{\mathbf{F}}_{\mathbf{x}}(\mathbf{x}^k) + \mathbf{b}^k, \quad k = 0,1,\ldots, \tag{9}$$

where $a_k > 0$ is a random variable; $\mathbf{b}^k = (b_1^k, \ldots, b_n^k)$ is a random vector, measurable with respect to the s-algebra $\boldsymbol{B}^k$ induced by the family of random variables $(\mathbf{x}^0, \ldots, \mathbf{x}^k)$; $\hat{\mathbf{F}}_{\mathbf{x}}(\mathbf{x}^k)$ is a generalized gradient of $F(\mathbf{x})$ at $\mathbf{x}^k$; $\mathbf{E}_{\mathbf{w}}(\boldsymbol{x}^k / \mathbf{x}^0, \ldots, \mathbf{x}^k)$ is the conditional mathematical expectation of $\boldsymbol{x}^k$ subject to $\mathbf{x}^0, \ldots, \mathbf{x}^k$; r_k is also measurable with respect to $\boldsymbol{B}^k$.

When $a_k \equiv 1$, $\mathbf{b}^k \equiv \mathbf{0}$ then $\boldsymbol{x}^k$ is said to be a stochastic generalized gradient (or a stochastic quasigradient) of $F(x)$. Method (8) - (9) is called the stochastic quasigradient method.

SQM are direct methods. Convergence theorems have been proved under certain requirements for $\boldsymbol{x}^k$, r_k (e.g. [1]). SQM are slow methods. That is why one of the main problems concerning their implementation is the choice of the step-size sequence $\{r_k\}$. Convergence theory states that any sequence with the properties

$$r_k \geq 0\,; \quad r_k \to 0\,, \ k \to \infty\,; \quad \sum_{k=0}^{\infty} r_k = \infty\,; \quad \sum_{k=0}^{\infty} r_k^2 < \infty$$

may be used as a step-size sequence. However, this approach does not use information obtained during the iterative process. A modern method for choosing r_k's is so-called adaptive step-size regulation ([9]).

As A. Gaivoronski pointed out ([3]), due to the specificity of stochastic programming problems and stochastic quasigradient methods (slow convergence, nonmonotonicity, and sometimes oscillatory behaviour), it is advisible to average the values of variables and of the objective function during a certain number of last iterations and take these quantities as the final approximation to the solution.

The second basic problem regarding the implementation of SQM is finding the projection of a current point $\mathbf{y}^k \equiv \mathbf{x}^k - r_k \mathbf{x}^k$ onto the feasible set X. As it is known, this is equivalent to solving the quadratic optimization problem

$$\frac{1}{2} \| \mathbf{y}^k - \mathbf{x} \|^2 \to \min$$

$$\mathbf{x} \in X .$$

This problem has to be solved at each iteration of the algorithm. That is why projection is the most onerous and time-consuming part of the SQM (and of any gradient type projection method for constrained optimization) and we need efficient algorithms for solving this problem.

The third important question concerning implementation of SQM is calculation of the stochastic quasigradient of the function to be minimized. For example, the components of the stochastic quasigradient of $F(\mathbf{x})$ (2') at iteration k are

$$x_j^k = \hat{f}_j(x_j^k, w_j^k) = \begin{cases} p_j, \text{ if } x_j^k \geq w_j^k \\ -q_j, \text{ if } x_j^k < w_j^k, \end{cases} \quad j = 1, \ldots, n \tag{10}$$

where x_j^k is the j-th component of $\mathbf{x}$ at iteration k and w_j^k is the j-th component of the observation of $\mathbf{w}$ at iteration k.

Algorithms for finding a projection onto a set defined by an inequality/equality constraint and bounds on the variables are suggested in [5], [6], [7], [8], etc. Stochastic programming is discussed, e.g., in [1], [4], etc. This paper is devoted to an efficient polynomial algorithm for finding a projection onto the set X (4) - (5) and (6) - (7).

2. ON PROJECTION IN THE IMPLEMENTATION OF SQM TO FACILITY LOCATION PROBLEMS

As pointed out in the Introduction, we need an efficient algorithm for finding a projection of a point onto certain feasible regions.

Consider the problem of finding the projection of an arbitrary point $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_n) \in \mathbf{R}^n$ onto the set X defined by (4) - (5) and (6) - (7). This problem is equivalent to

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

$$\mathbf{x} \in X .$$

Denote this problem by $(P^{\leq})$ in the first case and by $(P^{=})$ in the second case. Since $c(\mathbf{x})$ is a strictly convex function and X is a convex closed set, then this problem always has a unique solution when $X \neq \emptyset$.

First consider the case when X is defined by (4) - (5), under the assumptions:

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

1.b) $\sum_{j=1}^{n} d_j a_j \leq \alpha$. Otherwise the constraints (4) - (5) are inconsistent and $X \neq \emptyset$. In addition to this assumption we suppose that $\alpha \leq \sum_{j=1}^{n} d_j b_j$ in some cases which are specified below.

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

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

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

Theorem 1. A feasible solution $\mathbf{x}^* = (x_1^*,\ldots,x_n^*) \in X$ (defined by (4) - (5)) is an optimal solution to problem $(P^{\leq})$ if and only if there exists some $l \in \mathbf{R}^1_+$ such that

$$x_j^* = a_j, \quad j \in J_a^l \overset{\text{def}}{=} \{ j : l \geq \frac{\hat{x}_j - a_j}{d_j} \} \tag{11}$$

$$x_j^* = b_j, \quad j \in J_b^l \overset{\text{def}}{=} \{ j : l \leq \frac{\hat{x}_j - b_j}{d_j} \} \tag{12}$$

$$x_j^* = \hat{x}_j - l d_j, \quad j \in J^l \overset{\text{def}}{=} \{ j : \frac{\hat{x}_j - b_j}{d_j} < l < \frac{\hat{x}_j - a_j}{d_j} \}. \tag{13}$$

Proof: The Kuhn-Tucker necessary and sufficient optimality conditions for minimum $\mathbf{x}^*$ are:

$$x_j^* - \hat{x}_j + l d_j - u_j + v_j = 0, \quad j = 1,\ldots,n \tag{14}$$

$$u_j(a_j - x_j^*) = 0, \quad j = 1,\ldots,n \tag{15}$$

$$v_j(x_j^* - b_j) = 0, \quad j = 1, \ldots, n \tag{16}$$

$$l\left(\sum_{j=1}^{n} d_j x_j^* - a\right) = 0, \quad l \in \mathbf{R}_+^1 \tag{17}$$

$$\sum_{j=1}^{n} d_j x_j^* \le a \qquad (18) \equiv (4)$$

$$a_j \le x_j^* \le b_j, \quad j = 1, \ldots, n \qquad (19) \equiv (5)$$

$$u_j \in \mathbf{R}_+^1, \ v_j \in \mathbf{R}_+^1, \quad j = 1, \ldots, n. \tag{20}$$

Here l, u_j, v_j, $j = 1, \ldots, n$ are the Lagrange multipliers associated with the constraints (4), $a_j \le x_j$, $x_j \le b_j$, $j = 1, \ldots, n$, respectively. If $a_j = -\infty$ or $b_j = +\infty$ for some j, we do not consider the corresponding condition (15) [(16)] and Lagrange multiplier u_j $[v_j]$.

i) Let $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ be an optimal solution to $(P^{\le})$. Then there exist constants l, u_j, v_j, $j = 1, \ldots, n$ such that Kuhn-Tucker conditions (14) - (20) are satisfied. Consider both possible cases for l:

1) Let $l > 0$. Then system (14) - (20) becomes (14), (15), (16), (19), (20) and

$$\sum_{j=1}^{n} d_j x_j^* = a \ ,$$

that is, the inequality constraint (4) is satisfied with an equality for x_j^*, $j = 1, \ldots, n$ in this case.

a) If $x_j^* = a_j$ then $u_j \ge 0, v_j = 0$. Therefore $x_j^* - \hat{x}_j = u_j - l d_j \ge -l d_j$. Since $d_j > 0$ then

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

b) If $x_j^* = b_j$ then $u_j = 0, v_j \ge 0$. Therefore $x_j^* - \hat{x}_j = -v_j - l d_j \le -l d_j$. Hence

$$l \le \frac{\hat{x}_j - x_j^*}{d_j} \equiv \frac{\hat{x}_j - b_j}{d_j}.$$

c) If $a_j < x^*_j < b_j$ then $u_j = v_j = 0$. Therefore $x^*_j = \hat{x}_j - l d_j$.

Since $d_j > 0$, $j = 1, \ldots, n$, $l > 0$ by the assumptions then $\hat{x}_j - x^*_j \geq 0$. From $b_j > x^*_j$, $x^*_j > a_j$ it follows that

$$b_j - \hat{x}_j \geq x^*_j - \hat{x}_j, \; x^*_j - \hat{x}_j \geq a_j - \hat{x}_j .$$

Using $\hat{x}_j - x^*_j \geq 0$, $d_j > 0$, we obtain $l = \frac{\hat{x}_j - x^*_j}{d_j} \leq \frac{\hat{x}_j - a_j}{d_j}$, $l = \frac{\hat{x}_j - x^*_j}{d_j} \geq \frac{\hat{x}_j - b_j}{d_j}$, that is,

$$\frac{\hat{x}_j - b_j}{d_j} \leq l \leq \frac{\hat{x}_j - a_j}{d_j} .$$

Since we are not in cases a), b), these inequalities are strict. Hence

$$\frac{\hat{x}_j - b_j}{d_j} < l < \frac{\hat{x}_j - a_j}{d_j} .$$

2) Let $l = 0$. Then system (14) - (20) becomes

$$x^*_j - \hat{x}_j - u_j + v_j = 0, \; j = 1, \ldots, n, \qquad (15), (16), (18), (19), (20).$$

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

$$\frac{\hat{x}_j - a_j}{d_j} \leq 0 \equiv l .$$

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

$$\frac{\hat{x}_j - b_j}{d_j} \geq 0 \equiv l .$$

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

$$b_j - \hat{x}_j \geq x^*_j - \hat{x}_j = 0, \; 0 = x^*_j - \hat{x}_j \geq a_j - \hat{x}_j .$$

Multiplying both inequalities by $-\frac{1}{d_j}<0$ we obtain

$$\frac{\hat{x}_j-b_j}{d_j}\le 0\equiv l\ ,\quad l\equiv 0\le\frac{\hat{x}_j-a_j}{d_j}\ .$$

Since we are not in cases a), b), these inequalities are strict, that is, in case c) we have

$$\frac{\hat{x}_j-b_j}{d_j}<l<\frac{\hat{x}_j-a_j}{d_j}.$$

In order to describe cases a), b), c) for both 1) and 2), it is convenient to introduce the following index sets: J_a^l, J_b^l, J^l defined by (11), (12) and (13), respectively. Obviously $J_a^l\cup J_b^l\cup J^l=\{1,\dots,n\}$. Thus, the "necessity" part is proved.

ii) Conversely, let $\mathbf{x}^*\in X$ and components of $\mathbf{x}^*$ satisfy (11), (12) and (13), where $l\in\mathbf{R}_+^1$.

1) If $l>0$ then $x_j^*-\hat{x}_j<0,\ j\in J^l$ according to (13) and $d_j>0$. Set:

$$l=\frac{\hat{x}_j-x_j^*}{d_j}\ (>0)\text{ from (13): }\sum_{j\in J_a^l}d_ja_j+\sum_{j\in J_b^l}d_jb_j+\sum_{j\in J^l}d_j(\hat{x}_j-ld_j)=a$$

$u_j=v_j=0$ for $j\in J^l$;

$u_j=a_j-\hat{x}_j+ld_j$ (≥ 0 according to the definition of J_a^l), $v_j=0$ for $j\in J_a^l$;

$u_j=0$, $v_j=\hat{x}_j-b_j-ld_j$ (≥ 0 according to the definition of J_b^l) for $j\in J_b^l$.

By using these expressions, it is easy to check that conditions (14), (15), (16), (17), (20) are satisfied; conditions (18) and (19) are satisfied according to the assumption $\mathbf{x}^*\in X$.

2) If $l=0$ then $x_j^*-\hat{x}_j=0,\ j\in J^l$ according to (13) and

$$J^{l=0}=\{j:\frac{\hat{x}_j-b_j}{d_j}<0<\frac{\hat{x}_j-a_j}{d_j}\}.$$

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

Set:

$$l = \frac{\hat{x}_j - x_j^*}{d_j} \ (= 0); \ u_j = v_j = 0 \ \text{ for } \ j \in J^{l=0};$$

$$u_j = a_j - \hat{x}_j + l d_j = a_j - \hat{x}_j \ (\geq 0), \ v_j = 0 \ \text{ for } \ j \in J_a^{l=0};$$

$$u_j = 0, \ v_j = \hat{x}_j - b_j - l d_j = \hat{x}_j - b_j \ (\geq 0) \ \text{ for } \ j \in J_b^{l=0}.$$

Obviously conditions (14), (15), (16), (20) are satisfied; conditions (18), (19) are also satisfied according to the assumption $\mathbf{x}^* \in X$ and condition (17) is obviously satisfied for $l = 0$.

In both cases 1) and 2) of part ii), x_j^*, l, u_j, v_j, $j = 1, \ldots, n$ satisfy Kuhn-Tucker conditions (14) - (20) 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 $(P^{\leq})$. ❑

Theorem 1 is important because it describes components of the optimal solution to $(P^{\leq})$ only through the Lagrange multiplier l associated with the inequality constraint (4). Since we do not know the value of l we define an iterative process with respect to the Lagrange multiplier l and prove its convergence in Section 3.

From $d_j > 0$ and $a_j \leq b_j$, $j = 1, \ldots, n$ it follows that

$$ub_j \overset{\text{def}}{=} \frac{\hat{x}_j - b_j}{d_j} \leq \frac{\hat{x}_j - a_j}{d_j} \overset{\text{def}}{=} la_j, \ \ j = 1, \ldots, n$$

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

Consider problem $(P^{=})$ of finding a projection onto a set X of the form (6) - (7):

$(P^{=})$

$$c(\mathbf{x}) = \sum_{j=1}^{n} c_j(x_j) = \frac{1}{2} \sum_{j=1}^{n} (x_j - \hat{x}_j)^2 \to \min$$

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

and assume the following:

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

2.b) $\sum_{j=1}^{n} d_j a_j \le a \le \sum_{j=1}^{n} d_j b_j$. Otherwise the constraints (6) - (7) are inconsistent and the feasible region (6) - (7) is empty.

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

For the sake of simplicity throughout Theorem 2 we will use the same notations l, u_j, v_j, $j = 1,\dots,n$ for the Lagrange multipliers associated with (6), $a_j \le x_j$, $x_j \le b_j$, as we have used them in Theorem 1.

Theorem 2. A feasible solution $\mathbf{x}^* = (x_1^*,\dots,x_n^*) \in X$ (defined by (6) - (7)) is an optimal solution to problem $(P^=)$ if and only if there exists some $l \in \mathbf{R}^1$ such that

$$x_j^* = a_j, \quad j \in J_a^l \overset{\text{def}}{=} \{ j : l \ge \frac{\hat{x}_j - a_j}{d_j} \} \tag{21}$$

$$x_j^* = b_j, \quad j \in J_b^l \overset{\text{def}}{=} \{ j : l \le \frac{\hat{x}_j - b_j}{d_j} \} \tag{22}$$

$$x_j^* = \hat{x}_j - l d_j, \quad j \in J^l \overset{\text{def}}{=} \{ j : \frac{\hat{x}_j - b_j}{d_j} < l < \frac{\hat{x}_j - a_j}{d_j} \} . \tag{23}$$

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

3. THE ALGORITHMS

3.1. Analysis of the solution to problem $(P^{\le})$

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

Using (11), (12) and (13), condition (17) can be written as follows

$$l \left(\sum_{j \in J_a^l} d_j a_j + \sum_{j \in J_b^l} d_j b_j + \sum_{j \in J^l} d_j (\hat{x}_j - l d_j) - a \right) = 0, \quad l \ge 0 \tag{17'}$$

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

$$x_j^* = x_j(l) = \begin{cases} a_j, & j \in J_a^l \\ b_j, & j \in J_b^l \\ \hat{x}_j - l d_j, & j \in J^l. \end{cases} \tag{24}$$

Obviously $x_j(l)$, $j = 1, \dots, n$ are piecewise linear, monotonically nondecreasing, piecewise differentiable functions of l with two breakpoints at $l = \dfrac{\hat{x}_j - a_j}{d_j}$ and $l = \dfrac{\hat{x}_j - b_j}{d_j}$.

Let

$$d(l) \overset{\text{def}}{=} \sum_{j \in J_a^l} d_j a_j + \sum_{j \in J_b^l} d_j b_j + \sum_{j \in J^l} d_j (\hat{x}_j - l d_j) - a\,. \tag{25}$$

If we differentiate (25) with respect to l we get

$$d'(l) \equiv - \sum_{j \in J^l} d_j^2 < 0\,, \tag{26}$$

when $J^l \neq \emptyset$, and $d'(l) = 0$ when $J^l = \emptyset$. Hence $d(l)$ is a monotonic nonincreasing function of l, $l \in R_+^1$, and $\max_{l \ge 0} d(l)$ is attained at the minimum admissible value of l, that is, at $l = 0$.

Case 1. If $d(0) > 0$, in order that (17') and (18) ≡ (4) be satisfied, there exists some $l^* > 0$ such that $d(l^*) = 0$, that is,

$$\sum_{j=1}^{n} d_j x_j^* = a\ , \tag{27}$$

which means that the inequality constraint (4) is satisfied with an equality for l^* in this case.

Case 2. If $d(0) < 0$, then $d(l) < 0$ for all $l \ge 0$, and the maximum of $d(l)$ with $l \ge 0$ is $d(0) = \max_{l \ge 0} d(l)$ and it is attained at $l = 0$ in this case. In order that (17') be satisfied, l must be equal to 0. Therefore $x_j^* = \hat{x}_j$, $j \in J^{l=0}$ according to (13).

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

As we have seen, for the optimal value l we have $l \geq 0$ in all possible cases, as the Kuhn-Tucker condition (17) requires. We have shown that in Case 1 we need an algorithm for finding l^* which satisfies the Kuhn-Tucker conditions (14) - (20) but such that l^* satisfies (18) with an equality. In order that this be fulfilled, the set

$$X^{=} \overset{\mathrm{def}}{=} \{\mathbf{x} \in \mathbf{R}^n : \sum_{j=1}^{n} d_j x_j = a,\ a_j \leq x_j \leq b_j,\ j = 1,\dots,n\}$$

must be nonempty. That is why we have required $a \leq \sum_{j=1}^{n} d_j b_j$ in some cases in addition to the assumption $\sum_{j=1}^{n} d_j a_j \leq a$ (see assumption 1.b)). We have also used this in the proof of Theorem 1, part ii), when $l > 0$.

From the equation $d(l) = 0$, where $d(l)$ is defined by (25), we are able to obtain a closed form expression for l:

$$l = \left(\sum_{j \in J^l} d_j^2 \right)^{-1} \left(\sum_{j \in J_a^l} d_j a_j + \sum_{j \in J_b^l} d_j b_j + \sum_{j \in J^l} d_j \hat{x}_j - a \right), \tag{28}$$

because $d'(l) < 0$ according to (26) when $J^l \neq \emptyset$ (it is important that $d'(l) \neq 0$). This expression of l is used in the algorithm suggested for problem $(P^{\leq})$. It turns out that for our purposes without loss of generality we can assume that $d'(l) \neq 0$, that is, $d(l)$ depends on l, which means that $J^l \neq \emptyset$.

At iteration k of the algorithm let us denote by $l^{(k)}$ the value of the Lagrange multiplier associated with constraint (4) [(6)], by $a^{(k)}$ - the right-hand side of (4) [(6)]; by $J^{(k)}, J_a^{l(k)}, J_b^{l(k)}, J^{l(k)}$ - the current sets $J = \{1,\dots,n\}, J_a^l, J_b^l, J^l$, respectively.

3.2. Algorithm 1

The following algorithm for solving problem $(P^{\leq})$ is based on Theorem 1.

Algorithm 1 (for problem $(P^{\leq})$)

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

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

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

If $d(0) \le 0$ then $l := 0$, go to 8

else if $d(0) > 0$ then

if $a \le \sum_{j=1}^{n} d_j b_j$ go to 2

else if $a > \sum_{j=1}^{n} d_j b_j$ go to 9 (there does not exist $l^* > 0$ such that $d(l^*) = 0$).

2. $J^{l(k)} := J^{(k)}$. Calculate $l^{(k)}$ by using the explicit expression of l (see (28)). Go to 3.

3. Construct the sets $J_a^{l(k)}, J_b^{l(k)}, J^{l(k)}$ through (11), (12), (13) (with $j \in J^{(k)}$ instead of $j \in J$) and find their cardinalities $|J_a^{l(k)}|, |J_b^{l(k)}|, |J^{l(k)}|$, respectively. Go to 4.

4. Calculate

$$d(l^{(k)}) := \sum_{j \in J_a^{l(k)}} d_j a_j + \sum_{j \in J_b^{l(k)}} d_j b_j + \sum_{j \in J^{l(k)}} d_j (\hat{x}_j - l d_j) - a^{(k)} .$$

Go to 5.

5. If $d(l^{(k)}) = 0$ or $J^{l(k)} = \emptyset$ then $l := l^{(k)}$, $J_a^l := J_a^l \cup J_a^{l(k)}$, $J_b^l := J_b^l \cup J_b^{l(k)}$, $J^l := J^{l(k)}$, go to 8

else if $d(l^{(k)}) > 0$ go to 6

else if $d(l^{(k)}) < 0$ go to 7.

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

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

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

9. The problem has no solution because $X = \emptyset$ or there does not exist some $l^* > 0$ satisfying Theorem 1.

10. End.

Remark 1. To avoid a possible "endless loop" in programming the algorithm, the criterion of Step 5 to go to Step 8 at iteration k usually is $d(l^{(k)}) \in [-e, e]$ instead of $d(l^{(k)}) = 0$, where $e > 0$ is some given tolerance value up to which the equality $d(l^*) = 0$ may (for problem $(P^{\leq})$) or must (for problem $(P^{=})$) be satisfied.

3.3. Convergence and complexity of Algorithm 1

Theorem 3. Let $l^{(k)}$ be the sequence generated by Algorithm 1.

i) If $d(l^{(k)}) > 0$ then $l^{(k)} \leq l^{(k+1)}$;

ii) If $d(l^{(k)}) < 0$ then $l^{(k)} \geq l^{(k+1)}$;

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

Taking into consideration (26), Case 1, Case 2, Case 3 and Step 1 (the sign of $d(0)$) and Step 2 of Algorithm 1, it follows that $l^{(k)} \geq 0$ for each k. Since $x_j^{(k)}$ are determined from (13): $x_j^{(k)} = \hat{x}_j - l^{(k)} d_j$, $j \in J^{l(k)}$, substituted in $\sum_{j \in J^{l(k)}} d_j x_j^{(k)} = a^{(k)}$ at Step 2 of Algorithm 1 and since $l^{(k)} \geq 0$, $d_j > 0$ then $x_j^{(k)} - \hat{x}_j \leq 0$, that is, $\hat{x}_j \geq x_j^{(k)}$.

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

$$\sum_{j \in J^{l(k+1)}} d_j x_j^{(k)} \equiv \sum_{j \in J^{(k+1)}} d_j x_j^{(k)} = \sum_{j \in J^{(k)} \setminus J_a^{l(k)}} d_j x_j^{(k)} = a^{(k)} - \sum_{j \in J_a^{l(k)}} d_j x_j^{(k)} . \qquad (29)$$

Let $j \in J_a^{l(k)}$. According to definition (11) of $J_a^{l(k)}$ we have

$$\frac{\hat{x}_j - a_j}{d_j} \leq l^{(k)} = \frac{\hat{x}_j - x_j^{(k)}}{d_j} .$$

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

From (29), using that $d_j > 0$ and Step 6 we get

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

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

$$l^{(k)} = \frac{\hat{x}_j - x_{j_0}^{(k)}}{d_{j_0}} \leq \frac{\hat{x}_j - x_{j_0}^{(k+1)}}{d_{j_0}} = l^{(k+1)} .$$

We have used that the relationship between $l^{(k)}$ and $x_j^{(k)}$ is given by (13) for $j \in J^{l(k)}$ according to Step 2 of Algorithm 1 and that $\hat{x}_j \geq x_j^{(k)}$, $j \in J^{l(k)}$ according to (13) with $l^{(k)} \geq 0$ and $d_j > 0$.

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

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

Components $x_j^* = a_j$, $j \in J_a^l$ and $x_j^* = b_j$, $j \in J_b^l$ obviously satisfy (5). From

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

and $d_j > 0$ it follows that $a_j - \hat{x}_j < x_j^* - \hat{x}_j < b_j - \hat{x}_j$, $j \in J^l$. Therefore $a_j \leq x_j^* \leq b_j$ for $j \in J^l$. Hence all x_j^*, $j = 1, \dots, n$ satisfy (5).

We have proved that if $d(l)_{|l=0} \geq 0$ and $X^{=} \neq \emptyset$ then there exists some $l^* \geq 0$ such that $d(l^*) = 0$. Since at Step 2 we determine $l^{(k)}$ from the equality $\sum_{j \in J^{l(k)}} d_j x_j^{(k)} = \alpha^{(k)}$ for each k, then (4) is satisfied with an equality in this case. Otherwise, if $d(0) < 0$ then we set $l = 0$ (Step 2) and we have $\sum_{j \in J} d_j x_j(0) - \alpha \equiv d(0) < 0$, that is, (4) is satisfied as a strict inequality in this case.

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

Remark 2. Theorem 3, definitions of J_a^l (11), J_b^l (12) and J^l (13) and Steps 6, 7, 8 of Algorithm 1 allow us to assert that the calculation of l, operations $x_j^* := a_j$, $j \in J_a^{l(k)}$ (Step 6), $x_j^* := b_j$, $j \in J_b^{l(k)}$ (Step 7) and the construction of J_a^l, J_b^l, J^l are in accordance with Theorem 1.

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

Step 0 takes time $O(n)$. Step 1 (construction of sets J_a^0, J_b^0, J^0; calculation of $d(0)$ and checking whether X is empty) also takes time $O(n)$. The calculation of $x_j^{(k)}$, $j = 1, \ldots, n$ and $l^{(k)}$ requires $O(n)$ time (Step 2). Step 3 takes $O(n)$ time because of the construction of $J_a^{l(k)}$, $J_b^{l(k)}$, $J^{l(k)}$. Step 4 also requires $O(n)$ time and Step 5 requires constant time. Each of Steps 6, 7 and 8 takes time which is bounded by $O(n)$: 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 $O(n)$. Hence the algorithm has $O(n^2)$ 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 great n. In fact, this number does not depend on n but only on the three index sets defined by (11), (12), (13). In practice, the algorithm has $O(n)$ running time.

3.4. Algorithm 2 (for problem $(P^{=})$) and its convergence

After analysis of the optimal solution to problem $(P^{=})$, similar to that of problem $(P^{\leq})$, we suggest the following algorithm for solving problem $(P^{=})$.

Algorithm 2 (for problem $(P^{=})$)

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

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

3. Construct the sets $J_a^{l(k)}, J_b^{l(k)}, J^{l(k)}$ through (21), (22), (23) (with $j \in J^{(k)}$ instead of $j \in J$) and find their cardinalities $| J_a^{l(k)} |, | J_b^{l(k)} |, | J^{l(k)} |$. Go to 4.

Steps 4 - 8 are the same as steps 4 - 8 of Algorithm 1, respectively.

9. Problem $(P^{=})$ has no solution because the corresponding feasible set X (6) - (7) is empty.

10. End.

A theorem analogous to Theorem 3 holds for Algorithm 2 which guarantees the "convergence" of $l^{(k)}, J^{l(k)}, J_a^{l(k)}, J_b^{l(k)}$ to the optimal $l, J^{l}, J_a^{l}, J_b^{l}$, respectively.

Theorem 4. Let $l^{(k)}$ be the sequence generated by Algorithm 2.

i) If $d(l^{(k)}) > 0$ then $l^{(k)} \leq l^{(k+1)}$;

ii) If $d(l^{(k)}) < 0$ then $l^{(k)} \geq l^{(k+1)}$.

The proof of Theorem 4 is similar to that of Theorem 3.

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

4. EXTENSIONS

If it is allowed that $d_j = 0$ for some j in problems $(P^{\leq})$ and $(P^{=})$ then for such indices j we cannot construct the expressions $\frac{\hat{x}_j - a_j}{d_j}$ and $\frac{\hat{x}_j - b_j}{d_j}$ by means of which we define sets J_a^{l}, J_b^{l}, J^{l} for the corresponding problem. In such cases x_j's are not involved in (4) [in (6), respectively] for such indices j.

Let us denote

$$J = \{1, \ldots, n\}, \quad Z0 = \{ j \in J : d_j = 0\}.$$

Here "0" means the "computer zero". In particular, when $J = Z0$ and $a = 0$ then X is defined only by (5) [by (7), respectively].

Theorem 5. Problem $(P^{\leq})$ can be decomposed into two subproblems:

$(P1^{\leq})$ - for $j \in Z0$, $(P2^{\leq})$ - for $j \in J \setminus Z0$.

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

$$x_j^* = \begin{cases} a_j, & j \in Z0 \text{ and } \hat{x}_j \leq a_j \\ b_j, & j \in Z0 \text{ and } \hat{x}_j \geq b_j \\ \hat{x}_j, & j \in Z0 \text{ and } a_j < \hat{x}_j < b_j, \end{cases} \tag{30}$$

that is, the subproblem $(P1^{\leq})$ itself is decomposed into $n_0 =| Z0 |$ independent problems.

The optimal solution to $(P2^{\leq})$ is given by (11), (12), (13) with $J := J \setminus Z0$.

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

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

Theorem 6. Problem $(P^{=})$ can be decomposed into two subproblems:

$(P1^{=})$ - for $j \in Z0 := \{ j \in J : d_j = 0\}$ and $(P2^{=})$ - for $j \in J \setminus Z0$.

The optimal solution to $(P1^{=})$ is also given by (30). The optimal solution to $(P2^{=})$ is given by (21), (22), (23) with $J := J \setminus Z0$.

Thus, with the use of Theorem 5 and Theorem 6 we can express components of the optimal solutions to problems $(P^{\leq})$ and $(P^{=})$ without the necessity of constructing the expressions $\frac{\hat{x}_j - a_j}{d_j}$ and $\frac{\hat{x}_j - b_j}{d_j}$ with $d_j = 0$.

5. COMPUTATIONAL EXPERIMENTS

In this section we present the results of some numerical experiments obtained by applying a SQM with adaptive step-size regulation to multi-commodity facility location problems. The projection of the current approximation onto the feasible region was found by using the polynomial algorithms suggested in this paper.

Example 1.

$$\min\{F(\mathbf{x}) = E_{\mathbf{w}} \sum_{j=1}^{5} \max\{ p_j(x_j - w_j), q_j(w_j - x_j)\}\}$$

subject to

$$x_1 + x_2 + 2x_3 + 3x_4 + x_5 = 200$$
$$0 \le x_1 \le 50$$
$$0 \le x_2 \le 7$$
$$0 \le x_3 \le 7$$
$$0 \le x_4 \le 80$$
$$0 \le x_5 \le 25.$$

Here

$$\mathbf{p} = (p_1, p_2, p_3, p_4, p_5) = (1,0,3,1,2)\,,\ \mathbf{q} = (q_1, q_2, q_3, q_4, q_5) = (3,4,1,2,3)\,,$$

and w_j, $j = 1,2,3,4,5$ are random variables uniformly distributed on the closed segments

$$[0,60], [0,15], [0,17], [0,90], [0,40]\,,$$

respectively.

Obviously this is a problem of the form (2) - (3) with $n = 5$ and feasible set X of the form (6) - (7).

Optimal solution (by using SQM)
(Quantities of the last 10 iterations have been averaged):

$$\mathbf{x}^* = (42.08259, 6.98305, 3.76966, 41.86273, 17.80680)\,.$$

Optimal value of $F(\mathbf{x})$: $F(\mathbf{x}^*) = 96.63854$

The equality constraint is satisfied with tolerance: 0.00000000419.

This test example can also be solved analytically by using a nonlinear programming approach.

The analytical expression of the objective function is

$$F(\mathbf{x}) = \frac{1}{30}x_1^2 + \frac{2}{15}x_2^2 + \frac{2}{17}x_3^2 + \frac{1}{60}x_4^2 + \frac{1}{16}x_5^2 - 3x_1 - 4x_2 - x_3 - 2x_4 - 3x_5 + 278.5$$

Analytic solution (using a nonlinear programming approach, [9]):

$$\mathbf{x}^* = (41.88057, 7.00000, 2.48092, 41.27456, 22.33456) .$$

Optimal value of $F(\mathbf{x}) : F(\mathbf{x}^*) = 98.10089$

Example 2.

$$\min\{F(\mathbf{x}) = E_{\mathbf{w}} \sum_{j=1}^{6} \max\{ p_j(x_j - w_j), q_j(w_j - x_j)\}\}$$

subject to

$$\begin{aligned}
&3x_1 + 4x_2 + 7x_3 + 5x_5 + x_6 \le 872\\
&0 \le x_1 \le 45\\
&0 \le x_2 \le 56\\
&0 \le x_3 \le 32\\
&0 \le x_4 \le 27\\
&0 \le x_5 \le 456\\
&0 \le x_6 \le 45.
\end{aligned}$$

Here

$\mathbf{p} = (p_1, p_2, p_3, p_4, p_5, p_6) = (5,7,8,3,5,1)$, $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6) = (4,56,7,8,4,4)$,

and w_j , $j = 1,2,3,4,5,6$ are uniformly distributed random variables on closed segments

$$[0,56], [0,57], [0,36], [0,34], [0,468], [0,65] ,$$

respectively.

Obviously this is a problem of the form (2) - (3) with $n = 6$ and feasible set X of the form (4) - (5).

Optimal solution (by SQM)
(Quantities of the last 10 iterations have been averaged):

$$\mathbf{x}^* = (25.13417, 48.01999, 15.62868, 25.32763, 49.51728, 33.14936) .$$

Optimal value of $F(\mathbf{x}) : F(\mathbf{x}^*) = 972.59309$.

The effectiveness of algorithms for problems $(P^{\le})$ and $(P^{=})$ has been tested by many other examples.

REFERENCES

[1] Ermoliev, Yu., Methods for Stochastic Programming, Nauka, Moscow, 1976 (In Russian).

[2] Ermoliev, Yu., and Gaivoronski, A., "Stochastic quasigradient methods and their implementation", Working Paper A-2361, IIASA, Laxenburg, Austria, 1984.

[3] Gaivoronski, A., "Stochastic quasigradient methods and their implementation", in: Yu. Ermoliev, R.J.-B. Wets (eds.), Numerical Techniques for Stochastic Optimization, Springer Verlag, Berlin, 1988, 313-351.

[4] Kall, P., and Wallace, S.W., Stochastic Programming, Wiley, New York, 1994.

[5] Rockafellar, R.T., and Wets, R.J.-B., "A note about projections in the implementation of stochastic quasigradient methods", in: Yu. Ermoliev, R.J.-B. Wets (eds.), Numerical Techniques for Stochastic Optimization, Springer Verlag, Berlin, 1988, 385-392.

[6] Stefanov, S.M., "Facility location problems and methods for solving them", M.S. Thesis, Sofia University, 1989, 1-84 (In Bulgarian).

[7] Stefanov, S.M., "Nondifferentiable and stochastic programming", PhD Dissertation, 1991, 1-86.

[8] Stefanov, S.M., "Convex separable minimization problem with a linear inequality constraint and bounds on the variables", Computational Optimization and Applications. An International Journal, Kluwer Academic Publishers, to appear.

[9] Urasiev, S., "Adaptive stochastic quasigradient procedures", in: Yu. Ermoliev, R.J.-B. Wets (eds.), Numerical Techniques for Stochastic Optimization, Springer Verlag, Berlin, 1988, 373-384.