Filomat 31:3 (2017), 799–808
DOI 10.2298/FIL1703799B

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Parallel Implementation of Augmented Lagrangian Method within L-Shaped Method for Stochastic Linear Programs

Malihe Behboodi-Kahoo[a], Saeed Ketabchi[a]

[a] *Department of Applied Mathematics, Faculty of Mathematical Sciences,University of Guilan, P. O. Box 416351914, Rasht, Iran*

Abstract. In this paper, we study two-stage stochastic linear programming (SLP) problems with fixed recourse. The problem is often large scale as the objective function involves an expectation over a discrete set of scenarios. This paper presents a parallel implementation of the augmented Lagrangian method for solving SLPs. Our parallel method is based on a modified version of the L-shaped method and reducing linear master and recourse programs to unconstrained maximization of concave differentiable piecewise quadratic functions. The maximization problem is solved using the generalized Newton method. The parallel method is implemented in Matlab. Large scale SLP with several millions of variables and several hundreds of thousands of constraints are solved. The results of uniprocessor and multiprocessor computations are presented which show that the parallel algorithm is effective.

1. Introduction

Before presenting the mathematical formula of the two-stage stochastic linear program (SLP) model, we introduce some notation. Let $(\Omega; \vartheta; P)$ be a discrete probability space and consider $\omega \in \Omega$ which $\Omega = \{1, 2, \ldots, N\}$ is the set of scenarios with associated probabilities $\{p(\omega_1), p(\omega_2), \ldots, p(\omega_N)\}$ such that $\sum_{i=1}^{N} p(\omega_i) = 1$.
In this paper, consider the following two-stage stochastic linear program (SLP) with fixed recourse and a finite number of scenarios

$$\min_{x \in X} \quad f(x) = c^T x + \phi(x), \quad X = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}, \tag{1}$$

where

$$\phi(x) = E(Q(x, \omega)) = \sum_{i=1}^{N} Q(x, \omega_i) p(\omega_i),$$

and

$$Q(x, \omega) = \min_{y \in \mathbb{R}^{n_2}} \{q(\omega)^T y \mid Wy = h(\omega) - T(\omega)x \ , \ y \geq 0\}. \tag{2}$$

2010 *Mathematics Subject Classification*. Primary 90C15; Secondary 90C05, 90C20
Keywords. Two-stage stochastic linear programming, Recourse problem, L-Shaped method, Augmented Lagrangian method, Parallel computing
Received: 07 January 2015; Accepted: 09 January 2015
Communicated by Predrag Stanimirović
This author's work was supported by Faculty of Mathematical Science Grant in the date 10 July 2011 in University of Guilan.
Email addresses: behboodi.m@webmail.guilan.ac.ir (Malihe Behboodi-Kahoo), sketabchi@guilan.ac.ir (Saeed Ketabchi)

Here E represents the expectation with respect to $\omega \in \Omega$. In the second stage $q(.) \in \mathbb{R}^{n_2}$, $h(.) \in \mathbb{R}^{m_2}$ and matrix $T(.) \in \mathbb{R}^{m_2 \times n}$ for each realization ω and $W \in \mathbb{R}^{m_2 \times n_2}$ is the recourse matrix which we are taking here as fixed. Also, in the first stage, $A \in \mathbb{R}^{m_1 \times n_1}, c \in \mathbb{R}^{n_1}$ and $b \in \mathbb{R}^{m_1}$. In this paper, matrices A and W are assumed to have full row rank and $m_1 << n_1$ and $m_2 << n_2$.
Assuming that the uncertainties are represented by a finite scenario set, the stochastic program (1)-(2) can be reformulated [13] as the following deterministic equivalent program

$$\begin{aligned} \min \quad & c^T x + \sum_{i=1}^{N} \hat{q}_i y_i \\ s.t. \quad & Ax = b, \\ & T_i x + W y_i = h_i, \quad i = 1, \dots, N, \\ & x \geq 0,\ y_i \geq 0, \quad i = 1, \dots, N, \end{aligned} \tag{3}$$

where $T_i := T(\omega_i), h_i := h(\omega_i), y_i := y(\omega_i), q_i := q(\omega_i)$ and $\hat{q}_i := \rho(\omega_i) q_i$ for each realization ω_i of the random variable ω. This problem consists of $n_1 + N.n_2$ variables and $m_1 + N.m_2$ equality constraints. Usually N is a very large number. Hence, the stochastic linear programs (1)-(2) or (3) can become huge and very difficult to solve. The challenge of solving such problems has led to many interesting computational and theoretical developments and has provided a motivation for more study.

The most common algorithm used for solving this type of problem is the L-shaped method of Van Slyke and Wets [23], a variant of Benders' decomposition that decomposes the problem into first-stage (master) variables and second-stage (subproblem) variables. The L-shaped method generates optimality and feasibility cuts. In the L-shaped method, master and subproblems are linear. Thus, this can be a disadvantage associate with the L-shaped method in the cases that master and subproblems are large scale. Here, we propose a modified version of L-shaped method to reduce master and subproblems into quadratic unconstrained problems and by parallel implementation of algorithm accelerate the speed.
In fact, the main difficulty in solving SLP is the size. The evaluation of the objective function f in (1) is very expensive and the L-shaped method involves at least N optimization problems. To eliminate practically this drawback of the method, we suggest i) augmented Lagrangian method, which reduces master and subproblems to unconstrained quadratic problems and paves the way for using quasi-Newton algorithm and finding exact solutions of problems; ii) generalized Newton method, to minimize the number of iterations in solving unconstrained master and subproblems; iii) parallel processing techniques, to solved subproblems at each iteration of the algorithm in parallel environment. Then, results of some computational tests are described, which show that the method is capable of solving SLP of considerable size.

The remaining part of this paper is organized as follows. In Section 2, the L-shaped method is summarized. In Section 3 and 4, the new methods for feasibility and optimality cuts are discussed respectively. The modified L-shaped method is presented in Section 5. In Section 6, we discuss a parallel implementation of our method and give the numerical results. Also, concluding remarks are given in Section 7.

We now describe our notation. Let $a = [a_i]$ be a vector in $\mathbb{R}^n$. By a_+ we mean a vector in $\mathbb{R}^n$ whose *ith* entry is 0 if $a_i < 0$ and equals a_i if $a_i \geq 0$. By A^T we mean the transpose of matrix A, and $\nabla f(x_0)$ is the gradient of f at x_0. For $x \in R^n$, $\|x\|$ and $\|x\|_\infty$ denote 2−norm and infinity norm respectively. Also, $\vec{1}$ is a vector of ones and I is the identity matrix.

2. L-Shaped Decomposition

We can see L-shaped decomposition method in most of references such as [5, 13].
This method consists of three main steps: generating feasibility cuts, optimality cuts and solving the master problem. In this paper, we introduce a new method for solving two-stage stochastic linear programs with fixed recourse. Using quadratic program and an augmented Lagrangian method for generating feasibility and optimality cuts respectively, we can reduce the number of iterations and the time of solving two-stage stochastic linear program with fixed recourse in comparison traditional methods. In this algorithm, two types of constraints are sequentially appended: feasibility cuts and optimality cuts.
In the next section, the method for feasibility cuts is presented.

3. Feasibility Cut

In this section, we discuss the method for computing feasibility cut. This type of cuts is generated in L-shaped method in which some linear subproblems are solved. This cut tests whether the recourse problem is feasible for the current vector x^ν for all $i = 1, \dots, N$ or not. If not, this means that for some i, there is a hyperplane separating $h_i - T_i x^\nu$ and set $\{t \mid t = Wy, y \geq 0\}$. If we name the hyperplane $\{x \mid \sigma x = 0\}$, this hyperplane must satisfy $\sigma^T t \leq 0$ for all $t \in \{t \mid t = Wy, y \geq 0\}$ and $\sigma^T(h_i - T_i x^\nu) > 0$. In [23], Van Slyke and Wets introduced a method for computing the separating hyperplane and it was used. This hyperplane is obtained by taking σ for the value of the dual multipliers of a linear problem [13, 23].

In this paper, instead of the Slyke and Wets method, we introduce the following constrained quadratic program to obtain the separating hyperplane

$$\min_{y \in \mathbb{R}^{n_2}_+} \eta' = \frac{1}{2}\|Wy - (h_i - T_i x^\nu)\|^2. \tag{4}$$

Problem (4) is a convex quadratic program. Thus, it always has an optimal solution with $\eta' \geq 0$. The current vector satisfies the feasibility criterion if and only if the optimal value $\eta' = 0$ for all $i = 1, \dots, N$. If for some i, the optimal value $\eta' > 0$, then there exists a vector $y \in \mathbb{R}^{n_2}$ such that

$$\begin{cases} W^T(Wy - (h_i - T_i x^\nu)) \geq 0, \;\; y \geq 0, \\ y^T W^T(Wy - (h_i - T_i x^\nu)) = 0. \end{cases} \tag{5}$$

Let $\sigma = (h_i - T_i x^\nu) - Wy$. Therefore, from the above relation, we have

$$W^T \sigma \leq 0, \tag{6}$$

and

$$\begin{aligned} 0 < \eta' \quad &= \frac{1}{2}\|Wy - (h_i - T_i x^\nu)\|^2 \\ &= \frac{1}{2}\left(y^T W^T(Wy - (h_i - T_i x^\nu)) - (h_i - T_i x^\nu)^T(Wy - (h_i - T_i x^\nu))\right) \\ &= -\frac{1}{2}(h_i - T_i x^\nu)^T(Wy - (h_i - T_i x^\nu)) = \frac{1}{2}(h_i - T_i x^\nu)^T \sigma. \end{aligned}$$

Thus σ has the desired properties. For normality, we use $\sigma = \frac{(h_i - T_i x^\nu) - Wy}{\|(h_i - T_i x^\nu) - Wy\|}$.

To solve the problem (4), one can apply the Hager-Zhang active set algorithm (HZ-ASA) [10] or penalty method.

4. Optimality Cut

In this section, we present the augmented Lagrangian method for solving the recourse subproblem and generating an optimality cut. In the augmented Lagrangian method, an unconstrained maximization problem is solved which gives the projection of a point on the solution set of the following subproblem

$$\begin{aligned} \min \quad & \eta = \hat{q}_i y \\ s.t. \quad & Wy = h_i - T_i x^\nu, \\ & y \geq 0. \end{aligned} \tag{7}$$

Assume that $\hat{y} \in \mathbb{R}^{n_2}$ is an arbitrary vector. Consider the problem of finding the least 2-norm projection $\hat{y}_*$ of $\hat{y}$ on the solution set Y_* of the subproblem (7)

$$\begin{aligned} &\frac{1}{2}\|\hat{y}_* - \hat{y}\|^2 = \min_{y \in Y_*} \frac{1}{2}\|y - \hat{y}\|^2, \\ &Y_* = \{y \in \mathbb{R}^{n_2} \mid Wy = h_i - T_i x^\nu, \; \hat{q}_i{}^T y = \eta, \;\; y \geq 0\}. \end{aligned} \tag{8}$$

Considering that the objective function of the problem (8) is strictly convex, its solution is unique. The Lagrange function of problem (8) is as follows

$$L(y,p,\beta,\hat{y}) = \frac{1}{2}\|y-\hat{y}\|^2 + p^T(h_i - T_i x^\nu - Wy) + \beta(\hat{q}_i{}^T y - \eta),$$

where $p \in \mathbb{R}^{m_2}$ and $\beta \in \mathbb{R}$ are Lagrange multipliers and $\hat{y}$ is a constant vector. The dual problem of (8) has the form

$$\max_{\beta\in\mathbb{R}} \max_{p\in\mathbb{R}^{m_2}} \min_{y\in\mathbb{R}^{n_2}_+} L(y,p,\beta,\hat{y}). \tag{9}$$

We note that the solution of the inner minimization problem in (9) is (see [14, 19])

$$y = (\hat{y} + W^T p - \beta\hat{q}_i)_+. \tag{10}$$

By substituting (10) into $L(y,p,\beta,\hat{y})$, we have

$$\begin{aligned}\hat{L}(p,\beta,\hat{y}) &:= \min_{y\in\mathbb{R}^{n_2}_+} L(y,p,\beta,\hat{y}) \\ &= (h_i - T_i x^\nu)^T p - \frac{1}{2}\|(\hat{y} + W^T p - \beta\hat{q}_i)_+\|^2 - \beta\eta + \frac{1}{2}\|\hat{y}\|^2.\end{aligned}$$

Therefore, the dual problem of (8) is given by the following formula

$$\max_{\beta\in\mathbb{R}} \max_{p\in\mathbb{R}^{m_2}} \hat{L}(p,\beta,\hat{y}). \tag{11}$$

We can show that if β is sufficiently large, solving the inner maximization problem in (11) gives the unique solution of the problem (8). In this way, we solve the following maximization problem to obtain the solutions of (8) and the dual of (7) for sufficiently large β [8, 16–18]

$$\max_{p\in\mathbb{R}^{m_2}} S(p,\beta,\hat{y}) \tag{12}$$

in which $\beta, \hat{y}$ are constant and function $S(p,\beta,\hat{y})$ is introduced as follows

$$S(p,\beta,\hat{y}) = (h_i - T_i x^\nu)^T p - \frac{1}{2}\|(\hat{y} + W^T p - \beta\hat{q}_i)_+\|^2. \tag{13}$$

According to this fact, augmented Lagrangian method presents the following iteration process for solving (8):

$$\begin{aligned} p_{k+1} &\in \arg\max_{p\in\mathbb{R}^{m_2}}\{(h_i - T_i x^\nu)^T p - \frac{1}{2}\|(y_k + W^T p - \beta\hat{q}_i)_+\|^2\}, \\ y_{k+1} &= (y_k + W^T p_{k+1} - \beta\hat{q}_i)_+, \end{aligned} \tag{14}$$

where y_0 is an arbitrary vector.
For arbitrary y_0 and $\beta > 0$, this process converges to a solution $y_* \in Y_*$ in a finite number of steps M. Also, $\frac{p_{M+1}}{\beta}$ gives an exact solution of the dual problem of (7) (see [16–18] for detailed analysis and numerical result).

5. Quadratic L-Shaped Algorithm

In this section, we present the modified version of the L-shaped method by employing seperating and augmented Lagrangian methods for feasibility and optimality cuts respectively. Also, we rewrite the standard form of the master problem and apply an augmented Lagrangian method to solve it. Therefore, the modified version is as follows:

Step 1 Set $k = l = \nu = 0$.

Step 2 Set $\nu = \nu + 1$. Solve the following quadratic unconstrained problem:

$$\max_{z\in\mathbb{R}^{m+k+l}} r^T z - \frac{1}{2}\|(\hat{v} + R^T z - \beta t)_+\|^2. \tag{15}$$

In which

$$R = \begin{pmatrix} A & 0 & 0 & 0 & 0 \\ D & 0 & 0 & -I & 0 \\ E & \vec{1} & -\vec{1} & 0 & -I \end{pmatrix}, r = \begin{pmatrix} b \\ d \\ e \end{pmatrix}, t = \begin{pmatrix} c \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, v = \begin{pmatrix} x \\ \theta_+ \\ \theta_- \\ s_1 \\ s_2 \end{pmatrix},$$

and

$$D = \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_k \end{pmatrix}, d = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{pmatrix}, E = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_l \end{pmatrix}, e = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_l \end{pmatrix}.$$

In addition, $\hat{v} \in \mathbb{R}^{n+2+k+l}$ is an arbitrary vector.

Let z^ν be the unique optimal solution, then set $v^\nu = (x^\nu, \theta_+^\nu, \theta_-^\nu, s_1^\nu, s_2^\nu) = (\hat{v} + R^T z^\nu - \beta t)_+$ and $\theta^\nu = \theta_+^\nu - \theta_-^\nu$. If l is zero, solve the problem

$$\max_{z\in\mathbb{R}^{m+k}} r^T z - \frac{1}{2}\|(\hat{v} + R^T z - \beta t)_+\|^2. \tag{16}$$

where

$$R = \begin{pmatrix} A & 0 \\ D & -I \end{pmatrix}, r = \begin{pmatrix} b \\ d \end{pmatrix}, t = \begin{pmatrix} c \\ 0 \end{pmatrix}, v = \begin{pmatrix} x \\ s_1 \end{pmatrix},$$

and

$$D = \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_k \end{pmatrix}, d = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{pmatrix},$$

and $\hat{v} \in \mathbb{R}^{n+k}$ is an arbitrary vector. Let z^ν be the optimal solution and set $v^\nu = \begin{pmatrix} x^\nu \\ s_1^\nu \end{pmatrix} = (\hat{v} + R^T z^\nu - \beta t)_+$. In this case, θ^ν is considered equal to $-\infty$.

Step 3 For $i = 1, \ldots, N$, solve the following quadratic subproblems:

$$\min_{y\in\mathbb{R}_+^{n_2}} \eta' = \frac{1}{2}\|Wy - (h_i - T_i x^\nu)\|^2.$$

until, for some i, the optimal value $\eta' > 0$. In this case, let y be the optimal solution. Then, set $\sigma^\nu = \frac{(h_i - T_i x^\nu) - Wy}{\|(h_i - T_i x^\nu) - Wy\|}$ and define

$$D_{k+1} := (\sigma^\nu)^T T_i, \quad d_{k+1} := (\sigma^\nu)^T h_i$$

and append to the matrix D and vector d respectively. Set $k := k + 1$ and return to Step 2. If for all scenario i, $\eta' = 0$, go to Step 4.

Step 4 For all $i = 1, 2, \ldots, N$, solve the unconstrained quadratic program

$$\max_{p\in\mathbb{R}^{m_2}} S(p, \beta, \hat{y}) = (h_i - T_i x^\nu)^T p - \frac{1}{2}\|(\hat{y} + W^T p - \beta\hat{q}_i)_+\|^2, \tag{17}$$

where $\hat{y} \in \mathbb{R}^{n_2}$ is an arbitrary vector and $\beta \in \mathbb{R}$ is large enough. Let $p(\beta)$ be the optimal solution of Problem i of type (17) and set $\pi_i^\nu = \frac{p(\beta)}{\beta}$. Define

$$E_{l+1} := \sum_{i=1}^{N} (\pi_i^\nu)^T T_i, \quad e_{l+1} := \sum_{i=1}^{N} (\pi_i^\nu)^T h_i.$$

Let $\eta^\nu = e_{l+1} - E_{l+1}x^\nu$. If $\theta^\nu \geq \eta^\nu$, stop; x^ν is an optimal solution. Otherwise, set $l := l + 1$, append E_{l+1} and e_{l+1} to the matrix E and vector e respectively and return to Step 2.[1)]

From a computational point of view the above iterative framework is efficient because this algorithm is suitable for a parallel environment. Each subproblem may be solved independently of others, possibly in a seperate processing node.

6. Numerical Results

In this paper, we consider SLP problem with a rectangular matrix of coefficients in which the number of constraints noticeably more than the number of variables. Since most of such problems are complete recourse; hence here, we focus on SLP problem with complete recourse. It is obvious that for such problems, it is not needed the feasibilty cut. Therefore, in the quadratic L-shaped method, we have just optimality cut.

In each iteration of the quadratic L-shaped algorithm, master problem (15) or (16) and subproblems (4) are solved. Although, if the current point is feasible, N concave, piecewise quadratic, unconstrained maximization problems (17) have to be solved as well. It is obvious that the objective functions of the problems (15), (16) and (17) are only once-differentiable. Hence, the concept of generalized Hessian is used for them. In this section, we discuss the generalized Newton method for the problem (17). Note that the conditions of these problems are similar. Therefore, in this section, we propose the generalized Newton method for the problem (17) and duplicate it for master problems (15) and (16).
The gradient and the generalized Hessian of the objective function of (17) are

$$\nabla_p S(p, \beta, \hat{y}) = (h_i - T_i x^\nu) - W(\hat{y} + W^T p_{k+1} - \beta\hat{q}_i)_+,$$
$$\partial_p^2 S(p, \beta, \hat{y}) = -WD(\kappa)W^T,$$

where $D(\kappa)$ denotes a diagonal matrix where the ith-diagonal elements κ_j equals to 1, if $(\hat{y}+W^T p_{k+1}-\beta\hat{q}_i)_j > 0$ and equals to 0, if $(\hat{y} + W^T p_{k+1} - \beta\hat{q}_i)_j \leq 0$ for $j = 1, \ldots, n_2$.
In the algorithm, the generalized Hessian may be singular, thus we use a modified Newton. In each iteration for solving (17), we need to find the direction and new point by

$$(\partial_p^2 S(p, \beta, \hat{y}) - \delta I)d_s = -\nabla_p S(p, \beta, \hat{y}), \tag{18}$$
$$p_{s+1} = p_s + \lambda_s d_s,$$

where δ is a small positive number and λ_s is the suitable step length that is determined by Armijo rule [2, 19].

[1)] The Matlab code of this paper is available from the authors upon request.

6.1. Parallel Computation

In [18], we have shown that quadratic L-shaped method is more efficient and faster than L-shaped method. If we can implement a parallel program of quadratic L-shaped method, then, we can accelerate the speed of this method. Based on the results of our experiments, it seems that the quadratic L-shaped method (which use augmentedl Lagrangian algorithm within L-shaped method) is substantially better. This method can become more efficient to use. Here, we develop a parallel implementation of the proposed method in order to increase the size of problems and decrease the time of solving problem.

Generating optimality cut involves solving N quadratic programming problems (17). N is the number of scenarios which in the real problems can be very large. In fact, it is impractical or impossible to solve the real SLP problem on a single computer, especially given limited computer memory. This provides a strong motivation for using parallel computation. Also, in the quadratic L-shaped algorithm, the primary computation cost is in the solving N quadratic unconstraint problems (17). This can be done in parallel with minimal cost so that N problems (17) is broken into the number of processors.

For solving N subproblems (17), it is used generalized Newton method. Hence, we have single algorithm and different data correspond to each subproblems. Therefore, for solving them, we can use spmd loop in Matlab. Also, for this aim, we can use parfor loop. In spmd, workers meet at synchronization points. Since, here the SLP problem is fixed recourse; thus, the size of recourse problems are the same. In the other words, there is not considerable waiting time.

At each iteration of quadratic L-shaped method, one row is appended to matrix R in (17). If the size of A and the number of iterations be large, maybe the construction of generalized Hessian matrix or solving the linear system (18) in the computation of the generalized Newton direction be hard or imposible. In this case, we can use data decomposition and distribute matrix R into processors. To distribute matrix R, there are three schemes: column scheme, cellular scheme and row scheme. In each scheme, it is formed the Hessian matrix and solved the linear system (18) using explicit formulas. The details of these computation is given in [9]. Here, the problem solved dose not require data decomposition. For solving (17) in parallelism, we can use parfor and spmd loops. Since the size of recourse problems are the same, thus the implementing parallelism in both of loops are nearly simultaneous. In Table 1, the result is obtained by using parfor loop.

The results are presented in Tables 1-2 . In these tables, T and S denote the computation time in seconds and the parallel speedup in solving SLP problems respectively. Furthermore, the number of processors is shown by n_p. Table 3 presents the characteristics of the problems are solved in Table 1. In this table, m_1 and n_1 are the dimensions of matrix A in first-stage problem and d_1 is the density of this matrix. Also, m_2 and n_2 are the dimensions of the recourse matrix W in the second-stage and d_2 is the density of this matrix. The abvious accuracy criteria for (3) are a test for feasibility and optimality such

$$\begin{cases} f(x) = c^T x + \sum_{i=1}^{N} \hat{q}_i y_i = f^* \\ Ax = b, \\ T_i x + W y_i = h_i, \quad i = 1, \ldots, N, \\ x \geq 0,\ y_i \geq 0, \quad i = 1, \ldots, N, \end{cases}$$

where f^* is the optimal value of (3). These criteria are satifised for each number of processors. Here, Table 2 presents the criteria for 14 processors.

In this part, the test problems are randomly generated in the form of (3) and the data in Tables 1-2 was generated using 16 processors, dual-core computers, each with 12 GB of memory. When using 8 workers, they were all on a single computer. We used 2 computers for 16 workers.

Table 1: Time and Speedup

		n_p					
		1	2	4	8	12	14
p_1	T	53.97	62.524	74.771	44.331	31.990	30.275
	S	1	0.863	0.722	1.218	1.687	1.783
p_2	T	120.415	185.749	181.605	100.153	71.795	61.976
	S	1	0.648	0.663	1.202	1.677	1.943
p_3	T	73.973	21.853	29.915	21.538	18.497	19.581
	S	1	3.385	2.475	3.435	3.999	3.778
p_4	T	148.148	81.007	50.291	32.158	26.942	26.264
	S	1	1.829	2.946	4.607	5.499	5.641
p_5	T	228.299	88.854	109.140	59.671	48.76	43.766
	S	1	2.569	2.092	3.826	4.682	5.216
p_6	T	345.084	170.151	161.904	81.475	63.733	57.232
	S	1	2.028	2.131	4.235	5.415	6.030
p_7	T	513.511	248.264	219.738	109.382	79.289	71.622
	S	1	1.806	2.337	4.695	6.476	7.170
p_8	T	107.106	31.891	40.636	27.083	24.440	23.755
	S	1	3.358	2.636	3.955	4.382	4.509
p_9	T	150.920	63.652	50.887	33.345	28.075	27.092
	S	1	2.37	2.966	4.526	5.376	5.571
p_{10}	T	198.977	98.711	67.451	40.327	33.520	34.221
	S	1	2.016	2.950	4.934	5.936	5.814
p_{11}	T	225.253	138.655	107.028	59.351	48.241	42.524
	S	1	1.697	2.198	3.964	4.877	5.532
p_{12}	T	494.954	302.059	212.976	112.715	84.305	72.124
	S	1	1.639	2.324	4.391	5.871	6.862
p_{13}	T	582.212	88.312	142.883	122.903	67.971	62.330
	S	1	6.615	4.089	4.753	8.598	9.373

Table 2: Optimality conditions for problems solved in Table 1 on 14 processors

Name of problem	$\|\mathbf{Ax}-\mathbf{b}\|_\infty$	$\max_{1\le i\le N}\|\mathbf{T_i x}+\mathbf{W y_i}-\mathbf{h_i}\|_\infty$	$\|\mathbf{f(x)}-\mathbf{f^*}\|$	**time**
p_1	3.3751e-14	9.9476e-14	2.9104e-11	30.275
p_2	4.2633e-14	1.6342e-13	1.4438e-11	61.976
p_3	5.8265e-13	4.1211e-13	4.1910e-9	19.581
p_4	2.3164e-12	2.737e-12	5.9605e-8	26.246
p_5	1.8048e-12	6.1391e-12	1.8626e-8	43.766
p_6	1.2665e-12	5.9401e-12	7.4506e-9	57.231
p_7	1.5774e-12	7.1054e-13	2.2352e-8	71.622
p_8	1.7906e-12	2.5011e-12	6.7055e-8	23.755
p_9	2.8137e-12	3.1974e-12	2.3352e-8	27.092
p_{10}	2.2951e-12	4.1425e-12	7.4506e-9	34.222
p_{11}	2.10179e-12	5.2616e-12	1.4901e-8	42.524
p_{12}	2.5651e-12	7.09126e-12	7.4506e-8	72.124
p_{13}	9.6634e-13	2.8422e-12	9.3132e-9	62.330

Table 3: The size of problems solved in Table 1

	N	$m_1 \times n_1 \times d_1$	$m_2 \times n_2 \times d_2$
p_1	150	$100 \times 2e3 \times 0.01$	$100 \times 5e4 \times 0.001$
p_2	100	$100 \times 2e3 \times 0.01$	$100 \times 5e4 \times 0.001$
p_3	150	$100 \times 5e4 \times 0.01$	$100 \times 1e4 \times 0.001$
p_4	150	$100 \times 5e4 \times 0.01$	$1000 \times 1e4 \times 0.001$
p_5	100	$100 \times 5e4 \times 0.01$	$100 \times 5e4 \times 0.01$
p_6	150	$100 \times 5e4 \times 0.01$	$100 \times 5e4 \times 0.01$
p_7	200	$100 \times 5e4 \times 0.01$	$100 \times 5e4 \times 0.01$
p_8	112	$100 \times 6e4 \times 0.01$	$100 \times 1e4 \times 0.01$
p_9	150	$100 \times 6e4 \times 0.01$	$100 \times 1e4 \times 0.01$
p_{10}	200	$100 \times 6e4 \times 0.01$	$100 \times 1e4 \times 0.01$
p_{11}	100	$100 \times 6e4 \times 0.01$	$100 \times 5e4 \times 0.01$
p_{12}	200	$100 \times 6e4 \times 0.01$	$100 \times 5e4 \times 0.01$
p_{13}	100	$100 \times 6e4 \times 0.01$	$100 \times 1e5 \times 0.005$

From Table 1, it is reasonable to use a parallel implementation of the proposed method. By this technique, we can solve large SLP problems. The problem p_{13} is solved with more than 20 millions of variables and 20 thousands of constraints. Also, note that all problems were solved very accurately for all processors, the norms of the criteria do not exceed 7.4506e-8.
The results in the tables demonstrate the high efficiency of the proposed method. For instance, a SLP problem with more than twenty million nonnegative variables and more than two thousand equality constraints is solved to a good accuracy in about one minute (see the last row of the table). Therefore, the parallel program implemented for 14 processors turned out to be considerably more efficient for larger SLPs on more processors. Note that only Matlab's facilities are exploited for the computer implementation of our method.

7. Conclusion

In this paper, we have modified the basic version of the L-shaped method with the augmented Lagrangian algorithm and generalized Newton method. In our method, all problems are quadratic. Hence, it is named quadratic L-shaped method. At each iteration of our algorithm, it is solved N optimization problems. Therefore, we use parallel implementation of augmented Lagrangian method on several number of processors. By this technique, we can solve large SLP problems in a few time. Here, We have illustrated the high speedup and the high solution accuracy of algorithm by the problems with very large number of nonnegative variables and moderate number of equality type constraints.

References

[1] J. Abaffy, E. Allevi, A Modified L-Shaped Method, J. Optimiz. Theory App. 123 (2004) 255–270.
[2] L. Armijoo, Minimazation of Functions Havinig Lipschitz-Continus First Partial Derivetives, Pacifice J. Math. 16 (1966) 1–3.
[3] J. F. Benders, Partitioning Procedures for Solving Mixed-Variables Programming Problems, Numer. Math. 4 (1962) 238–252.
[4] J. R. Birge, Decomposition and partitioning methods for multistage stochastic linear programs, Oper. Res. 33 (1985) 989–1007.
[5] J. R. Birge, F. Louveaux, Introduction to Stochastic Programming, Springer Series in Operation Research Springer-Verlag, New York, 1997.
[6] G. B. Dantzig, Linear Programming Under Uncertainty, Management Sci. 1 (1955) 197–206.
[7] G. B. Dantzig, P. Wolfe, Decomposition Principle for Linear Programs, Oper. Res. 8 (1960) 101–111.
[8] Yu. G. Evtushenko, A. I. Golikov, N. Mollaverdi, Augmented Lagrangian Method for Large-Scale Linear Programming Problem, Comp. Math. Math. Phys. 49 (2009) 1303–1317.
[9] V. A. Garanzha, A. I. Golikov, Yu. G. Evtushenko, M. Kh. Nguen, Parallel Implementation of Newton's Method for Solving Large-Scale Linear Programs, Optim. Method. Softw. 20 (2005) 515–524.
[10] W. W. Hager, H. Zhang, A new active set algorithm for box constrained optimization, SIAM J. Optim. 17 (2006) 526–557.
[11] J. L. Higle, S. Sen, Stochastic decomposition: an algorithm for two stage linear programs with recourse, Math. Oper. Res. 16 (1991) 650–669.
[12] J. B. Hiriart-Urruty, J. J. Strodiot, V. H. Nguyen, Generalized Hessian Matrics and Second-Order Optimality Condisions for Problems CL1 data, Appl. Math. Opt. 11 (1984) 43–56.
[13] P. Kall, S. W. Wallace, Stochastic Programming, John Wiley & Sons, 1994.
[14] S. Ketabchi, E. Ansari-Piri, On the Solution Set of Convex Problems and Its Numerical Application, J. Comput. Appl. Math. 206 (2007) 288–292.
[15] G. E. Karniadakis, R. M. Kirby, Parallel Scientific Computing in C++ and MPI: A seamless approach to parallel algorithms and their implementation, Cambridge University Press, 2003.
[16] S. Ketabchi, M. Behboodi-Kahoo, Augmented Lagrangian method for recourse problem of two-stage stochastic linear programming, Kybernetika 1 (2013) 188–198.
[17] S. Ketabchi, M. Behboodi-Kahoo, Smoothing techniques and augmented Lagrangian method for recourse problem of two-stage stochastic linear programming, J. Appl. Math. 2013, Article ID 735916, 8 pages http://dx.doi.org/10.1155/2013/735916.
[18] S. Ketabchi, M. Behboodi-Kahoo, Augmented Lagrangian method within L-shaped method for stochastic linear programs, Submitted in Appl. Math. Comput. (2013).
[19] O. L. Mangasarian, A Newton Method for Linear Programming, J. Optimiz. Theory App. 121 (2004) 1–18.
[20] S. S. Nielsen, S. A. Zenios, Scalable Parallel Benders Decomposition for Stochastic Linear Programming, Parallel Comput. 23 (1997) 1069–1088.
[21] A. Prekopa, Probabilistic programming. In: Stochastic Programming, (A. Ruszczynski and A. Shapiro,eds.), Handbook in Operations Research and Management Science, Elsevier, Amsterdam, 10 (2003) 267–352.
[22] S. S. Nielsen, S. A. Zenios, Scalable Parallel Benders Decomposition for Stochastic Linear Programming, Parallel Comput. 23 (1997) 1069–1088.
[23] R. M. Van Slyke, R. Wets, L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming, SIAM J. Appl. Math. 17 (1969) 638–663.