Computational Optimization and Applications, 1, (1992), 93–111

Application of the Alternating Direction Method of Multipliers to Separable Convex Programming Problems

MASAO FUKUSHIMA
Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, Kyoto 606, Japan

Received July 31, 1991, Revised March 24, 1992.

Abstract. This paper presents a decomposition algorithm for solving convex programming problems with separable structure. The algorithm is obtained through application of the alternating direction method of multipliers to the dual of the convex programming problem to be solved. In particular, the algorithm reduces to the ordinary method of multipliers when the problem is regarded as nonseparable. Under the assumption that both primal and dual problems have at least one solution and the solution set of the primal problem is bounded, global convergence of the algorithm is established.

Keywords: Convex programming, separable problems, decomposition, alternating direction method of multipliers, parallel algorithm.

1. Introduction

Decomposition of large-scale problems is a classical topic of optimization ([5]) that still attracts serious attention of many researchers. In particular, recent advances of parallel computers allow algorithms to take full advantage of problem structure, e.g., separability (see [2] and the references cited therein).

The purpose of this paper is to present a new decomposition algorithm for solving the separable convex programming problem

$$\begin{aligned} \text{(P)} \quad \text{minimize} \quad & \sum_{j=1}^{n} f_j(x_j) \\ \text{subject to} \quad & \sum_{j=1}^{n} c_{ij}(x_j) \leq 0, \quad i = 1, \ldots, m, \\ & x_j \in X_j \subset R^{d_j}, \quad j = 1, \ldots, n \end{aligned}$$

where $f_j : R^{d_j} \to R$ and $c_{ij} : R^{d_j} \to R$ are convex functions and X_j are nonempty closed convex subsets of R^{d_j} for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$. In the

following, all vectors are understood to be column vectors, but we shall often write $x = (x_1, \ldots, x_n)$ instead of $x = (x_1, \ldots, x_n)^T$ or $x = (x_1^T, \ldots, x_n^T)^T$ in order to simplify the notation.

There are a number of approaches for solving problem (P), but dual methods seem to be most popular. Let $y \in R^m$ be a vector of Lagrange multipliers and define the Lagrangian $L : R^d \times R^m \to R$ by

$$\begin{aligned} L(x, y) &= \sum_{j=1}^{n} f_j(x_j) + \left\langle y, \sum_{j=1}^{n} c_j(x_j) \right\rangle \\ &= \sum_{j=1}^{n} \{f_j(x_j) + \langle y, c_j(x_j)\rangle\} \end{aligned} \tag{1}$$

where $d = \Sigma_{j=1}^{n} d_j$, $x = (x_1, \ldots, x_n)$, $c_j(x_j) = (c_{1j}(x_j), \ldots, c_{mj}(x_j))$, $j = 1, \ldots, n$, and $\langle \cdot, \cdot \rangle$ denotes the inner product. Then the Lagrangian dual of (P) is the problem

$$\text{maximize } g(y) \text{ subject to } y \geq 0$$

where the function $g : R^m \to [-\infty, +\infty)$ is defined by

$$g(y) = \inf_{x \in X} L(x, y) \tag{2}$$

with $X = X_1 \times \cdots \times X_n \subset R^d$. Using (1) and (2) we may rewrite the dual problem as

$$(D) \text{ maximize } \sum_{j=1}^{n} g_j(y) \text{ subject to } y \geq 0$$

where $g_j : R^m \to [-\infty, +\infty)$ are defined by

$$g_j(y) = \inf_{x_j \in X_j} \{f_j(x_j) + \langle y, c_j(x_j)\rangle\}, \quad j = 1, \ldots, n \tag{3}$$

Thus, the evaluation of the function g decomposes into evaluations of the n functions g_j, which can be done in parallel by solving n independent minimization problems involving each individual variable x_j only.

The dual problem (D) is a concave maximization problem. Moreover, if the infimum on the right-hand side of (3) is always attained uniquely for each j, which is the case particularly if the functions f_j are strictly convex and cofinite in the sense of [7] (p. 116), then the dual functions g_j are not only finite-valued everywhere but also continuously differentiable so that various descent methods can be applied to problem (D). Detailed discussions on dual descent methods for the case of linear constraints may be found in Tseng ([12]). In the general case, however, the functions g_j are not necessarily differentiable, and, further, it is quite likely that $g_j(y)$ may take the value $-\infty$ for some y. A natural approach to

such a problem would therefore be to use a carefully designed nondifferentiable optimization technique ([6]).

Another interesting way of dealing with problem (P) under the general setting is to modify the problem by adding a quadratic term to the objective function, thereby obtaining a strictly convex objective function. A typical example is the proximal point method ([8, 9]), which iteratively solves a subproblem of the form (P) with objective function replaced by $\Sigma_{j=1}^{n}\{f_j(x_j) + (r/2)||x_j - x_j^{(k)}||^2\}$, where $r > 0$ is a given constant and $x_j^{(k)}$ are components of the current iterate $x^{(k)}$. Since this problem has a strongly convex objective function, its dual is a differentiable concave maximization problem, to which various descent-type algorithms can be applied.

A different but closely related approach is to utilize the method of multipliers ([1]). Though straightforward application of the latter method to problem (P) generally loses the separable structure of the problem, careful reformulation of the problem may still lead to implementation that preserves the inherent separability for some special classes of problems. Specifically, Bertsekas and Tsitsiklis [2] (pp. 249–251), consider the separable problem with linear constraints and show how the method of multipliers can be applied without destroying the separable structure of the given problem. Moreover, in [2] (p. 254), it is shown that the same class of problems can also be dealt with effectively by the alternating direction method of multipliers ([3, 4]), which may be viewed as a variant of the method of multipliers (see also [13] and [14] for related methods). Note that these methods do not require the strict convexity of the functions f_j, but assume that the coupling constraints are all linear.

In this paper, we consider applying the alternating direction method of multipliers to the dual problem (D), rather than the primal problem (P) as is done in [2]. The objective functions f_j are not assumed strictly convex and the constraint functions c_j are not assumed affine. Note that none of the functions are required to be differentiable. Interestingly, the resulting algorithm resembles the method of Spingarn ([11]) that is derived from a variant of the proximal point method ([10]). The difference between Spingarn's algorithm and ours is analogous to that between the proximal method of multipliers ([9]) and the method of multipliers ([1]).

2. Preliminaries

In this section, we briefly review the alternating direction method of multipliers. For more detail, the reader may refer to [2], [3] and [4].

The method is designed to solve a problem of the form

$$\begin{aligned} &\text{minimize} && G_1(y) + G_2(z) \\ &\text{subject to} && Ay - z = 0,\ y \in C_1,\ z \in C_2 \end{aligned} \tag{4}$$

where $G_1 : R^s \to (-\infty, +\infty]$ and $G_2 : R^t \to (-\infty, +\infty]$ are closed proper convex functions, $\mathbf{A}$ is a $t \times s$ matrix, and C_1 and C_2 are nonempty closed convex subsets of R^s and R^t, respectively. The iteration of the alternating direction method of multipliers may be written as

$$y^{(k+1)} := \arg\min_{y \in C_1} \left\{ G_1(y) + \langle p^{(k)}, Ay \rangle + \frac{r}{2} ||Ay - z^{(k)}||^2 \right\} \tag{5}$$

$$z^{(k+1)} := \arg\min_{z \in C_2} \left\{ G_2(z) - \langle p^{(k)}, z \rangle + \frac{r}{2} ||Ay^{(k+1)} - z||^2 \right\} \tag{6}$$

$$p^{(k+1)} := p^{(k)} + r[Ay^{(k+1)} - z^{(k+1)}] \tag{7}$$

where r is a positive constant and the initial vectors $p^{(0)}$ and $z^{(0)}$ may be chosen arbitrarily. Note that (5) and (6) correspond to a single cycle of the (block) Gauss-Seidel method to minimize the augmented Lagrangian

$$\Lambda_r(y, z, p^{(k)}) = G_1(y) + G_2(z) + \langle p^{(k)}, Ay - z \rangle + \frac{r}{2} ||Ay - z||^2$$

for problem (4), while (7) is the ordinary multiplier update in the method of multipliers. The minimum on the right-hand side of (5) is uniquely attained whenever $\text{rank}(A) = s$, while the minimum on the right-hand side of (6) is always attained uniquely. Therefore, the above method is well defined under the assumption $\text{rank}(A) = s$. Moreover, under the same assumption, it can be shown that the sequence $\{(y^{(k)}, z^{(k)}, p^{(k)})\}$ generated by (5)–(7) is bounded and every limit point of $\{(y^{(k)}, z^{(k)})\}$ is a solution of problem (4), whenever the solution set of the latter problem is nonempty. In addition, the sequence $\{(z^{(k)}, p^{(k)})\}$ has a unique limit point (for a proof of these results, see Propositions 4.2 and its proof in [2], Chapter 3).

3. Algorithm

In this section, we obtain a decomposition algorithm for problem (P) by applying the alternating direction method of multipliers to problem (D). Throughout this section, we make the following assumption.

ASSUMPTION. *Problems (P) and (D) have nonempty solution sets. Moreover, the solution set of (P) is bounded.*

In applying the alternating direction method of multipliers to (D), we adopt a technique used in Bertsekas and Tsitsiklis ([2], p. 246 and p. 256). First we rewrite (D) in the following equivalent form:

$$(\hat{D}) \quad \begin{aligned} &\text{maximize} && \sum_{j=1}^{n} g_j(z_j) \\ &\text{subject to} && y - z_j = 0,\ j = 1, \ldots, n \\ & && z_j \geq 0,\ j = 1, \ldots, n \end{aligned}$$

where $z_j \in R^m$, $j = 1, \ldots, n$, are artificial variables. We then apply the alternating direction method of multipliers (5)–(7) to problem $(\hat{D})$ with the following identifications:

$$\begin{aligned} &G_1(y) = 0, \quad C_1 = R^m \\ &A = \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} \in R^{nm \times m}, \quad z = (z_1, z_2, \ldots, z_n) \in R^{nm} \\ &G_2(z) = -\Sigma_{j=1}^{n} g_j(z_j), \quad C_2 = \{z \in R^{nm} | z_j \geq 0,\ j = 1, \ldots, n\} \end{aligned} \tag{8}$$

where I is the $n \times n$ identity matrix.

Partitioning the multiplier vector $p \in R^{nm}$ as

$$p = (p_1, p_2, \ldots, p_n)$$

where $p_j \in R^m$, $j = 1, \ldots, n$, we may write the alternating direction method of multipliers for $(\hat{D})$ as follows:

$$y^{(k+1)} := \arg\min_{y \in R^m} \left\{ \left\langle \sum_{j=1}^{n} p_j^{(k)}, y \right\rangle + \frac{r}{2} \sum_{j=1}^{n} \| y - z_j^{(k)} \|^2 \right\} \tag{9}$$

$$z_j^{(k+1)} := \arg\max_{z_j \geq 0} \left\{ g_j(z_j) + \langle p_j^{(k)}, z_j \rangle - \frac{r}{2} \| y^{(k+1)} - z_j \|^2 \right\}, \quad j = 1, \ldots, n \tag{10}$$

$$p_j^{(k+1)} := p_j^{(k)} + r(y^{(k+1)} - z_j^{(k+1)}),\ j = 1, \ldots, n \tag{11}$$

where r is a positive constant and the initial vectors $p_j^{(0)}$, $j = 1, \ldots, n$, and $z_j^{(0)}$, $j = 1, \ldots, n$, may be chosen arbitrarily. Note that, by the separability of $(\hat{D})$, the updates (10) and (11) of variables $z = (z_1, \ldots, z_n)$ and $p = (p_1, \ldots, p_n)$ can be performed in parallel for $j = 1, \ldots, n$.

To further simplify the computation, observe that, by carrying out the minimization, $y^{(k+1)}$ can be written as

$$y^{(k+1)} = \frac{1}{n} \sum_{j=1}^{n} z_j^{(k)} - \frac{1}{nr} \sum_{j=1}^{n} p_j^{(k)}$$

The next lemma shows that, for each j, $z_j^{(k+1)}$ can be computed via solving an optimization problem that involves the component variable x_j.

LEMMA 1. *Under the assumption that (P) has a nonempty bounded solution set,* $z_j^{(k+1)} = (z_{1j}^{(k+1)}, \ldots, z_{mj}^{(k+1)})$ *is given by*

$$z_{ij}^{(k+1)} = \max\left\{0,\ y_i^{(k+1)} + \frac{1}{r}(p_{ij}^{(k)} + c_{ij}(x_j^{(k+1)}))\right\},\quad i = 1, \ldots, m \tag{12}$$

where $x_j^{(k+1)}$ *is a solution of the minimization problem*

$$\begin{aligned} &\text{minimize} \quad f_j(x_j) + \frac{r}{2}\sum_{i=1}^{m}\left[\max\left\{0,\ y_i^{(k+1)} + \frac{1}{r}(p_{ij}^{(k)} + c_{ij}(x_j))\right\}\right]^2 \\ &\text{subject to} \quad x_j \in X_j \end{aligned} \tag{13}$$

Proof. By the definition (3) of g_j we have

$$\begin{aligned} &\max_{z_j \geq 0}\left\{g_j(z_j) + \langle p_j^{(k)}, z_j\rangle - \frac{r}{2}\|y^{(k+1)} - z_j\|^2\right\} \\ &= \max_{z_j \geq 0}\inf_{x_j \in X_j}\left\{f_j(x_j) + \langle z_j, p_j^{(k)} + c_j(x_j)\rangle - \frac{r}{2}\|y^{(k+1)} - z_j\|^2\right\} \end{aligned} \tag{14}$$

Note that if the solution set of problem (P) is nonempty and bounded, then the functions $\Sigma_{j=1}^n f_j$, $\Sigma_{j=1}^n c_{ij}$, $i = 1, \ldots, m$, and the set $X = X_1 \times \cdots \times X_n$ have no direction of recession in common in the sense of [7] (p. 61 and p. 69). By the separability of the problem, this implies that the same is true for the functions f_j, c_{ij}, $i = 1, \ldots, m$, and the set X_j for each j.

Let us consider the saddle function $K : R^{d_j} \times R^m \to [-\infty, +\infty]$ defined by

$$K(x_j, z_j) = \begin{cases} f_j(x_j) + \langle z_j, p_j^{(k)} + c_j(x_j)\rangle - \frac{r}{2}\|y^{(k+1)} - z_j\|^2, & \text{if } x_j \in X_j,\ z_j \geq 0 \\ +\infty, & \text{if } x_j \notin X_j,\ z_j \geq 0 \\ -\infty, & \text{if } z_j \ngeq 0. \end{cases}$$

Then, by the preceding arguments, it is seen that the convex function $K(\cdot, z_j)$ has no direction of recession for any $z_j > 0$, while the convex function $-K(x_j, \cdot)$ trivially has no direction of recession for any $x_j \in X_j$. Therefore it follows from [7] (Theorems 37.3 and 37.6) that

$$\sup_{z_j \geq 0}\inf_{x_j \in X_j} K(x_j, z_j) = \inf_{x_j \in X_j}\sup_{z_j \geq 0} K(x_j, z_j) < \infty$$

and that the function K actually has a saddle point $(\bar{x}_j, \bar{z}_j) \in X_j \times \{z_j | z_j \geq 0\}$ such that

$$K(\bar{x}_j, \bar{z}_j) = \max_{z_j \geq 0}\min_{x_j \in X_j} K(x_j, z_j) = \min_{x_j \in X_j}\max_{z_j \geq 0} K(x_j, z_j) \tag{15}$$

Consequently, it follows from (14) that

$$\max_{z_j \geq 0} \left\{ g_j(z_j) + \langle p_j^{(k)}, z_j \rangle - \frac{r}{2} \|y^{(k+1)} - z_j\|^2 \right\}$$
$$= \min_{x_j \in X_j} \max_{z_j \geq 0} \left\{ f_j(x_j) + \langle z_j, p_j^{(k)} + c_j(x_j) \rangle - \frac{r}{2} \|y^{(k+1)} - z_j\|^2 \right\} \tag{16}$$

But, for any fixed x_j, the maximum on the right-hand side of (16) is uniquely attained by

$$z_j = \left[y^{(k+1)} + \frac{1}{r}(p_j^{(k)} + c_j(x_j)) \right]_+$$

where $[\cdot]_+$ denotes the orthogonal projection of a vector onto the nonnegative orthant, i.e.

$$z_{ij} = \max \left\{ 0, y_i^{(k+1)} + \frac{1}{r}(p_{ij}^{(k)} + c_{ij}(x_j)) \right\}, \quad i = 1, \ldots, m \tag{17}$$

We may thus substitute (17) into the function on the right-hand side of (16) to eliminate the variables z_j. As a result, we obtain the objective function of problem (13). Therefore, if a solution $x_j^{(k+1)}$ of problem (13) is found, we can determine $z_j^{(k+1)}$ by (12). Clearly such $(x_j^{(k+1)}, z_j^{(k+1)})$ is a saddle point $(\bar{x}_j, \bar{z}_j)$ of K satisfying (15). (Note that the existence of a solution of problem (13) is guaranteed, since the functions f_j, c_{ij}, $j = 1, \ldots, m$, and the set X_j have no direction of recession in common.) □

To summarize, we state the algorithm as follows.

3.1. *Algorithm 1*

Step 1: Choose a constant $r > 0$ and initial vectors $(p_j^{(0)}, z_j^{(0)})$, $j = 1, \ldots, n$, arbitrarily. Set $k := 0$.

Step 2: Compute

$$y^{(k+1)} = \frac{1}{n} \sum_{j=1}^{n} z_j^{(k)} - \frac{1}{nr} \sum_{j=1}^{n} p_j^{(k)} \tag{18}$$

Step 3: For each $j = 1, \ldots, n$, find a solution $x_j^{(k+1)}$ of the minimization problem

$$\text{minimize} \quad f_j(x_j) + \frac{r}{2} \sum_{i=1}^{m} \left[\max \left\{ 0, y_i^{(k+1)} + \frac{1}{r}(p_{ij}^{(k)} + c_{ij}(x_j)) \right\} \right]^2$$
$$\text{subject to} \quad x_j \in X_j,$$

and determine $z_j^{(k+1)} = (z_{1j}^{(k+1)}, \ldots, z_{mj}^{(k+1)})$ by

$$z_{ij}^{(k+1)} = \max\left\{0,\ y_i^{(k+1)} + \frac{1}{r}(p_{ij}^{(k)} + c_{ij}(x_j^{(k+1)}))\right\},\ i = 1, \ldots, m \tag{19}$$

Step 4: For each $j = 1, \ldots, n$, compute

$$p_j^{(k+1)} = p_j^{(k)} + r(y^{(k+1)} - z_j^{(k+1)})$$

Set $k := k + 1$ and go to step 2. □

Since the matrix A defined by (8) has full column rank, it follows from the fact mentioned at the end of the previous section that the sequence $\{(y^{(k)}, z^{(k)}, p^{(k)})\}$ generated by Algorithm 1 is bounded. Moreover, every limit point of $\{(y^{(k)}, z^{(k)})\}$ is a solution of problem $(\hat{D})$, and the sequence $\{(z^{(k)}, p^{(k)})\}$ has a unique limit point. It now remains to establish convergence of the sequence $\{x^{(k)}\}$.

THEOREM 1. *Suppose that problems (P) and (D) have nonempty solution sets and that the solution set of (P) is bounded. Then the sequence $\{x^{(k)}\}$ generated by Algorithm 1 is bounded and every limit point of $\{x^{(k)}\}$ is a solution of (P).*

Proof. Since $\{(z^{(k)}, p^{(k)})\}$ has a unique limit point, and since any limit point of $\{(y^{(k)}, z^{(k)})\}$ is a solution of problem $(\hat{D})$, the sequence $\{y^{(k)}\}$ also has a unique limit point, which is equal to that of $\{z_j^{(k)}\}$ for any j. That is, we have

$$y^{(k)} \to y^* \tag{20}$$

$$z_j^{(k)} \to y^*,\ j = 1, \ldots, n \tag{21}$$

$$p_j^{(k)} \to p_j^*,\ j = 1, \ldots, n \tag{22}$$

for some $y^* \in R^m$ and $p_j^* \in R^m$, $j = 1, \ldots, n$, where in particular y^* is a solution of problem (D).

For each j, let us define the functions $F_j^{(k)} : R^{d_j} \to (-\infty, +\infty]$, $k = 1, 2, \ldots$, and $F_j^* : R^{d_j} \to (-\infty, +\infty]$ by

$$F_j^{(k)}(x_j) = f_j(x_j) + \frac{r}{2}\sum_{i=1}^{m}\left[\max\left\{0,\ y_i^{(k)} + \frac{1}{r}(p_{ij}^{(k-1)} + c_{ij}(x_j))\right\}\right]^2 + \delta(x_j|X_j)$$

and

$$F_j^*(x_j) = f_j(x_j) + \frac{r}{2}\sum_{i=1}^{m}\left[\max\left\{0,\ y_i^* + \frac{1}{r}(p_{ij}^* + c_{ij}(x_j))\right\}\right]^2 + \delta(x_j|X_j)$$

respectively, where $\delta(\cdot|X_j)$ is the indicator function of the set X_j. Note that the sequence $\{F_j^{(k)}\}$ of closed convex functions e-converges (epi-converges) to the closed convex functions F_j^* in the sense of [15]. Moreover, since (P) has a nonempty bounded solution set, the functions f_j, c_{ij}, $i = 1, \ldots, m$, and the set

X_j have no direction of recession in common, which in turn implies that the function F_j^* has no direction of recession and, hence, has a compact solution set. Since $x_j^{(k)}$ is a minimum of the function $F_j^{(k)}$ for each k, it follows from [15] (Theorem 9) that the sequence $\{x_j^{(k)}\}$ is bounded and every limit point belongs to the set of minima of F_j^*.

Now let x_j^* denote an arbitrary limit point of $\{x_j^{(k)}\}$ for each j. Since x_j^* minimizes the function F_j^* we have

$$\begin{aligned} 0 \quad &\in \quad \partial F_j^*(x_j^*) \\ &= \quad \partial f_j(x_j^*) + \sum_{i=1}^{m} \max\left\{0,\, y_i^* + \frac{1}{r}(p_{ij}^* + c_{ij}(x_j^*))\right\} \partial c_{ij}(x_j^*) + \partial \delta(x_j^* | X_j) \end{aligned} \tag{23}$$

where ∂ denotes the subdifferential operator.

Incidentally it follows from (18) and (20)–(22) that

$$\sum_{j=1}^{n} p_j^* = 0 \tag{24}$$

Moreover, (19) together with (20)–(22) implies

$$y_i^* = \max\left\{0,\, y_i^* + \frac{1}{r}(p_{ij}^* + c_{ij}(x_j^*)\right\},\ i = 1, \ldots, m \tag{25}$$

which, in turn, implies that

$$\begin{aligned} y_i^* = 0 \quad &\Rightarrow \quad p_{ij}^* + c_{ij}(x_j^*) \leq 0 \\ y_i^* > 0 \quad &\Rightarrow \quad p_{ij}^* + c_{ij}(x_j^*) = 0 \end{aligned} \tag{26}$$

Then it follows from (23) and (25) that

$$0 \in \partial f_j(x_j^*) + \sum_{i=1}^{m} y_i^* \partial c_{ij}(x_j^*) + \partial \delta(x_j^* | X_j) \tag{27}$$

Since (27) holds for each j, we have

$$\begin{aligned} 0 \quad &\in \quad \sum_{j=1}^{n} \partial f_j(x_j^*) + \sum_{i=1}^{m} y_i^* \sum_{j=1}^{n} \partial c_{ij}(x_j^*) + \sum_{j=1}^{n} \partial \delta(x_j^* | X_j) \\ &= \quad \partial\left(\sum_{j=1}^{n} f_j(x_j^*)\right) + \sum_{i=1}^{m} y_i^* \partial\left(\sum_{j=1}^{n} c_{ij}(x_j^*)\right) + \partial\left(\sum_{j=1}^{n} \delta(x_j^* | X_j)\right) \end{aligned} \tag{28}$$

where the last equality follows from [7] (Theorem 23.8). On the other hand, the relation (26) implies that the inequalities

$$p_{ij}^* + c_{ij}(x_j^*) \leq 0$$

hold for all i and j, so that

$$\sum_{j=1}^{n} p_{ij}^* + \sum_{j=1}^{n} c_{ij}(x_j^*) \leq 0, \quad i = 1, \ldots, m$$

Therefore, by (24), we have

$$\sum_{j=1}^{n} c_{ij}(x_j^*) \leq 0, \quad i = 1, \ldots, m \tag{29}$$

Moreover, by (26) and (24), we have

$$y_i^* > 0 \quad \Rightarrow \quad \sum_{j=1}^{n} c_{ij}(x_j^*) = 0 \tag{30}$$

Since (28)–(30) imply that $x^* = (x_1^*, \ldots, x_n^*)$ together with the Lagrange multiplier vector $p^* = (p_1^*, \ldots, p_m^*)$ satisfies the Karush-Kuhn-Tucker optimality conditions for problem (P), the proof is complete. □

4. Discussion

The algorithm presented in the previous section solves at each iteration the following n independent subproblems:

$$\begin{aligned} &\text{minimize} && f_j(x_j) + \frac{r}{2}\sum_{i=1}^{m}\left[\max\left\{0, y_i^{(k+1)} + \frac{1}{r}(p_{ij}^{(k)} + c_{ij}(x_j))\right\}\right]^2 \\ &\text{subject to} && x_j \in X_j \end{aligned} \tag{31}$$

However, problem (31) resembles an augmented Lagrangian for the problem

$$\begin{aligned} &\text{minimize} && f_j(x_j) \\ &\text{subject to} && c_{ij}(x_j) \leq -p_{ij}^{(k)}, \quad i = 1, \ldots, m \\ & && x_j \in X_j \end{aligned}$$

with the Lagrange multiplier vector $y^{(k+1)} \in R^m$ that is common to all $j = 1, \ldots, n$. Therefore, the vector $p_j^{(k)} = (p_{1j}^{(k)}, \ldots, p_{mj}^{(k)})$ may be thought of as the (negative) amount of the resources assigned to the jth subsystem. After solving subproblems (31), the algorithm updates the Lagrange multiplier vector separately for each j based on the respective solutions of (31). At this stage, there are n different estimates $z_j^{(k+1)}$ of Lagrange multipliers for $j = 1, \ldots, n$. Using this information, the algorithm then reallocates the resources by updating $p_j^{(k+1)}$, and proceeds to the

next iteration. At the beginning of the new iteration, the different values $z_j^{(k+1)}$ of Lagrange multiplier estimates are integrated to the single Lagrange multiplier vector $y^{(k+1)}$, which is again common to all subsystems. In this manner, the algorithm successively updates not only Lagrange multipliers (prices) for the coupling constraints, but also the amount of resources to be assigned to each subsystem.

A decomposition method with similar nature has been proposed by Spingarn [11] for the same class of separable problems (P). Though the original notation and formulation of [11] are somewhat different from ours, suitable transformation of variables and rearrangement reveal that Spingarn's algorithm, which is called Algorithm 4 in [11], may be restated as follows.

Step 1: Choose a constant $r > 0$ and initial vectors $x^{(0)} \in R^d$ and $(z_j^{(0)}, p_j^{(0)})$, $j = 1, \ldots, n$, such that $\Sigma_{j=1}^n p_j^{(0)} = 0$. Set $k := 0$.

Step 2: Compute

$$y^{(k+1)} = \frac{1}{n}\sum_{j=1}^{n} z_j^{(k)}$$

Step 3: For each $j = 1, \ldots, n$, find the unique solution $x_j^{(k+1)}$ of the minimization problem

$$\begin{aligned} \text{minimize} \quad & f_j(x_j) + \frac{1}{2n^2 r}\|x_j - x_j^{(k)}\|^2 \\ & + \frac{r}{2}\sum_{i=1}^{m}\left[\max\left\{0,\, y_i^{(k+1)} + \frac{1}{r}(p_{ij}^{(k)} + c_{ij}(x_j))\right\}\right]^2 \\ \text{subject to} \quad & x_j \in X_j, \end{aligned}$$

and determine $z_j^{(k+1)} = (z_{1j}^{(k+1)}, \ldots, z_{mj}^{(k+1)})$ by

$$z_{ij}^{(k+1)} = \max\left\{0,\, y_i^{(k+1)} + \frac{1}{r}(p_{ij}^{(k)} + c_{ij}(x_j^{(k+1)}))\right\},\quad i = 1, \ldots, m$$

Step 4: For each $j = 1, \ldots, n$, compute

$$\hat{p}_j^{(k+1)} = p_j^{(k)} + r(y^{(k+1)} - z_j^{(k+1)})$$

and

$$p_j^{(k+1)} = \hat{p}_j^{(k+1)} - \frac{1}{n}\sum_{l=1}^{n}\hat{p}_l^{(k+1)}$$

Set $k := k + 1$ and go to step 2. □

This algorithm differs from Algorithm 1 in two respects. First, each subproblem solved at step 3 contains the extra quadratic term $\frac{1}{2n^2 r}\|x_j - x_j^{(k)}\|^2$, which is

peculiar to methods of the proximal point type. Second, step 4 contains an additional update of the multiplier vectors $p_j^{(k)}$ in order to maintain the condition $\Sigma_{j=1}^n p_j^{(k)} = 0$ throughout the iterations. In the special case where $n = 1$, Spingarn's algorithm reduces to the proximal method of multipliers [9], while Algorithm 1 is nothing but the ordinary method of multipliers [1].

5. Numerical results

In this section, we report some computational results with Algorithm 1 presented in Section 3. The test problems used in our numerical experiments are of the form (P), in which f_j and c_{ij} are all strictly convex quadratic functions, and $X_j = R^{d_j}$. The termination criterion for the algorithm is

$$||y^{(k+1)} - y^{(k)}||_\infty < \text{TOL} \tag{32}$$

where $||\cdot||_\infty$ denotes the l_∞-norm of a vector and TOL is a positive tolerance parameter. For the experiments below, TOL equals 10^{-5}.

5.1. *Effect of parameter r*

First we examine how the value of parameter r affects the performance of the algorithm. In this experiment, we randomly generated five problems of the size $n = 4$, $d_j = 4$, $j = 1, \ldots, n$, and $m = 15$. Table 1 reports the numbers of iterations to achieve the above termination criterion for each of the five problems with four different values for r. Observe that the choice of r significantly affects the performance of the algorithm. In particular, if r is chosen too small or too large, then convergence of the algorithm can be very slow.

Table 1. Value of parameter r versus number of iterations

Problem number	$r = 5$	$r = 10$	$r = 20$	$r = 30$
1	149	81	83	116
2	183	100	116	156
3	250	136	86	143
4	140	75	103	150
5	202	118	92	146

The size of test problems is $(n, d_j, m) = (4, 4, 15)$, and tolerance parameter TOL in the stopping criterion is 10^{-5}.

To further investigate the algorithm, we check how the quantity $||y^{(k+1)} - y^{(k)}||_\infty$ decreases as the iteration proceeds. Figure 1 displays the graphs of the average values of $||y^{(k+1)} - y^{(k)}||_\infty$ versus the number of iterations for each value of r. From this figure, $||y^{(k+1)} - y^{(k)}||_\infty$ decreases almost geometrically for each value of r.

5.2. *Effect of problem size*

Next we examine the performance of the algorithm for various problem sizes. There are three factors that determine the problem size, i.e., m = the number of constraints, n = the number of blocks, and d_j = the number of variables in each block j. Figures 2–4 show results of the experiments where one of the three factors is varied while the rest are kept fixed. The numbers in each figure represent the numbers of iterations to obtain convergence for five randomly generated problems for each problem size. Throughout the experiments, the values of parameters r and TOL are fixed.

Figure 2 shows that the number of iterations increases as the number of constraints m increases. This is due to the fact that the algorithm addresses the dual problem (D) and there are as many dual variables as there are constraints. On the other hand, Figures 3 and 4 show that the other two factors n and d_j, which are unrelated to the size of problem (D), do not seem to seriously affect the number of iterations to obtain convergence. Note that we are concerned here only with the number of iterations. Computational time depends, of course, on the number of blocks and the number of variables in the primal problem (P).

5.3. *Comparison with the method of partial inverses*

We have also tested the method of partial inverses [11] as described in Section 4. It was remarkable that the method of partial inverses behaved almost identically with Algorithm 1 for the test problems used in the aforementioned numerical experiments. More precisely, the number of iterations of the former method differs only slightly from the latter for almost all test problems. To explain, note that the subproblem of the method of partial inverse contains an extra proximal term $(2n^2r)^{-1}||x_j - x_j^{(k)}||^2$ in the objective function of the subproblems. For a proper value of r, e.g., $r = 5 \sim 30$ in the above experiments, this proximal term can be very small. For instance, when $n = 4$ and $r = 10$, the coefficient of the proximal term becomes $(2n^2r)^{-1} = 1/320$. In such a case, we may regard the two algorithms as solving almost the same subproblems, and, hence, there is little difference in the performance of the two algorithms.

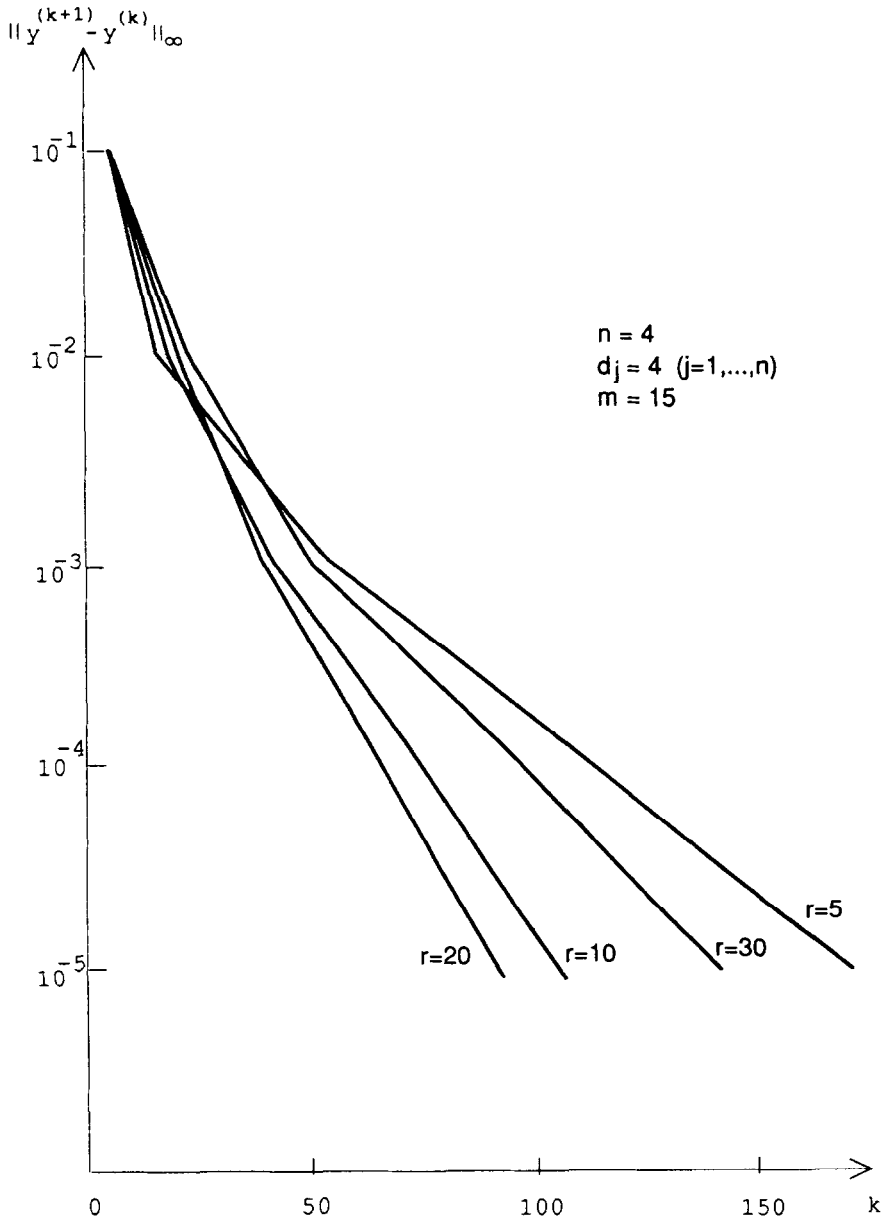

Figure 1. Behavior of the alternating direction methods of multipliers.

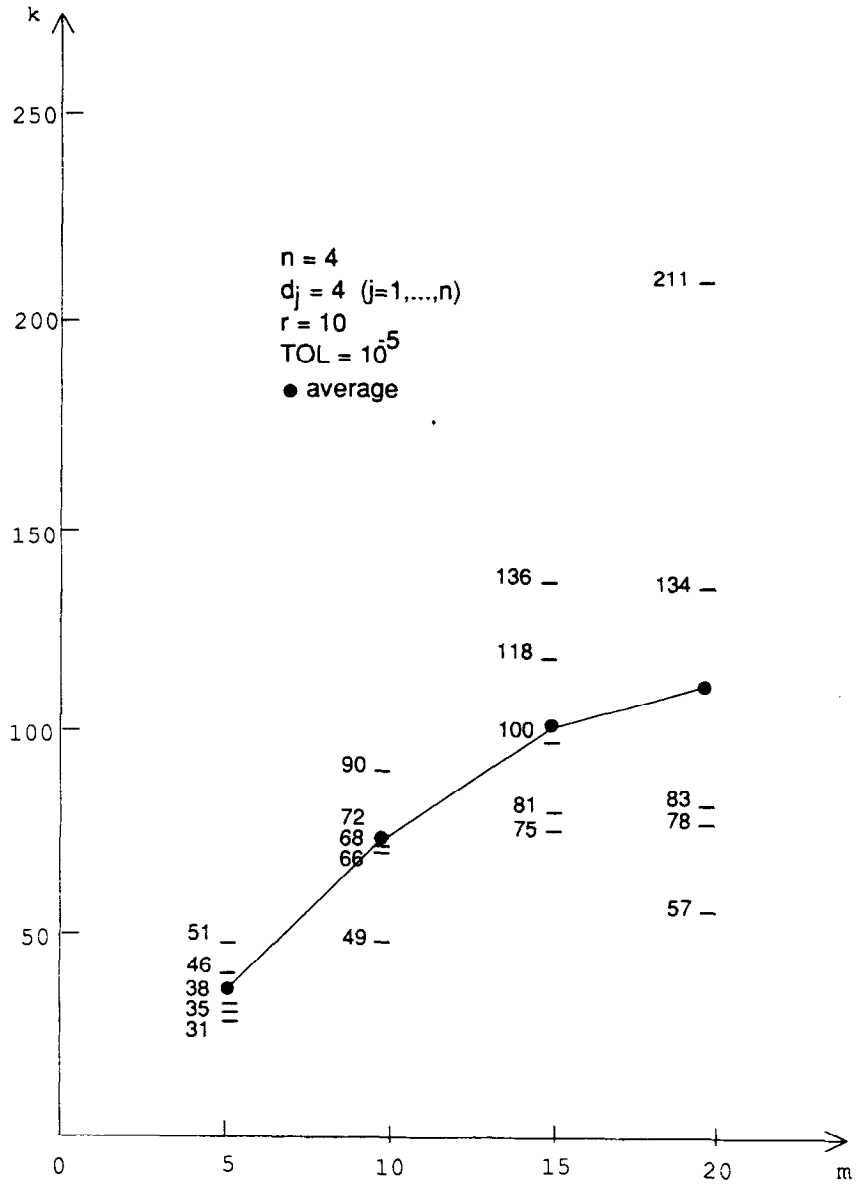

Figure 2. Number of constraints versus number of iterations.

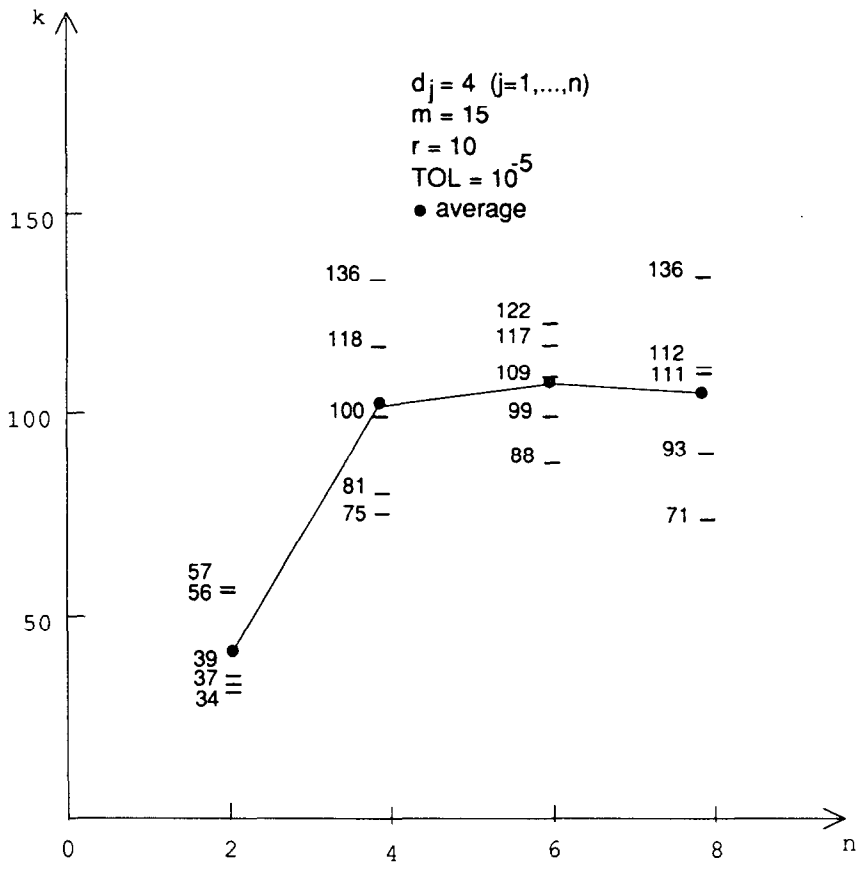

Figure 3. Number of blocks versus number of iterations.

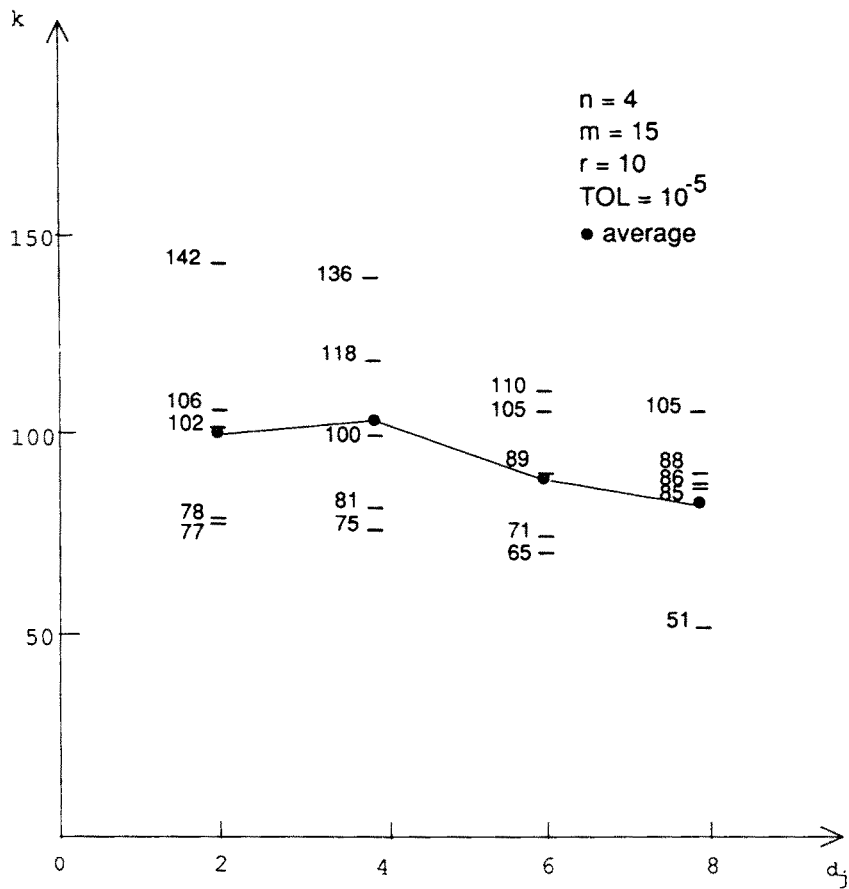

Figure 4. Number of variables in each block versus number of iterations.

Table 2. Comparison with the method of partial inverses

Problem number	Method	$r = 1$	$r = 0.1$	$r = 0.05$
1	ADMM[1]	65	504	939
	MPI[2]	67	506	941
2	ADMM[1]	222	1180	1778
	MPI[2]	223	1304	1927
3	ADMM[1]	49	442	814
	MPI[2]	48	444	815
4	ADMM[1]	102	758	1402
	MPI[2]	103	690	1279
5	ADMM[1]	76	669	1125
	MPI[2]	71	654	1127

The size of test problems is $(n, d_j, m) = (2, 2, 8)$ and the tolerance parameter TOL in the stopping criterion is 10^{-5}.

[1] ADMM stands for the alternating direction method of multipliers.

[2] MPI stands for the method of partial inverses.

To examine this issue further, we experimented with using very small values of r, so that the proximal term $(2n^2r)^{-1}||x_j - x_j^{(k)}||^2$ brought a significant difference between the two algorithms. We generated five small test problems randomly, and each problem was solved by both algorithms with $r = 1$, 0.1 and 0.05. The numbers of iterations to achieve the criterion in (32) are summarized in Table 2. From this table, the performance of the two algorithms differs from one problem to another and it is not possible to conclude that one is superior to the other. It is noted that, as mentioned in Subsection 5.1, too small values of r generally cause slow convergence and the values of r shown in Table 2 are precisely those values for which convergence of both algorithms is extremely slow. From a practical viewpoint, we may therefore consider that the two algorithms have equal efficiency at least for the class of test problems treated here.

Acknowledgment

I am grateful to one of the referees for careful reading of the manuscript. This work was supported in part by the Scientific Research Grant-in-Aid from the

Ministry of Education, Science, and Culture, Japan (number 02680025).

References

1. D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press: New York, 1982.
2. D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall: Englewood Cliffs, NJ, 1989.
3. M. Fortin and R. Glowinski, "On decomposition-coordination methods using an augmented Lagrangian," in M. Fortin and R. Glowinski (eds.), *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, North-Holland: Amsterdam, 1983, pp. 97–146.
4. D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. App.* vol. 2, pp. 17–40, 1976.
5. L.S. Lasdon, *Optimization Theory for Large Systems*, Macmillan: New York, 1970.
6. C. Lemaréchal, "Nondifferentiable optimization," in G.L. Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd (eds.), *Handbooks in Operations Research and Management Science, Vol. 1, Optimization*, North-Holland: Amsterdam, 1989, pp. 529–572.
7. R.T. Rockafellar, *Convex Analysis*, Princeton University Press: Princeton, NJ, 1970.
8. R.T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. on Control and Optimization*, vol. 14, pp. 877–898, 1976.
9. R.T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Math. of Oper. Res.*, vol. 1, pp. 97–116, 1976.
10. J.E. Spingarn, "Partial inverse of a monotone operator," *Appl. Math. and Optimization*, vol. 10, pp. 247–265, 1983.
11. J.E. Spingarn, "Applications of the method of partial inverses to convex programming: Decomposition," *Math. Programming*, vol. 32, pp. 199–223, 1985.
12. P. Tseng, "Dual ascent methods for problems with strictly convex costs and linear constraints: A unified approach," *SIAM J. on Control and Optimization*, vol. 28, pp. 214–242, 1990.
13. P. Tseng, "Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming," *Math. Programming*, vol. 48, pp. 249–263, 1990.
14. P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM J. on Control and Optimization*, vol. 29, pp. 119–138, 1991.
15. R.J.B. Wets, "Convergence of convex functions, variational inequalities and convex optimization problems," in R.W. Cottle, F. Giannessi and J.-L. Lions (eds.), *Variational Inequalities and Complementarity Problems*, John Wiley: Chichester, U.K., 1980, pp. 375–403.