An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 41 (2001) 51–62
www.elsevier.nl/locate/camwa

Space-Decomposition Multiplier Method for Constrained Minimization Problems

CHIN-SUNG LIU
Industrial Technology Research Institute
Mechanical Industry Research Laboratories, Hsinchu, Taiwan, R.O.C.

CHING-HUAN TSENG
Applied Optimum Design Laboratory, Department of Mechanical Engineering
National Chiao Tung University, Hsinchu 30050, Taiwan, R.O.C.
chtseng@cc.nctu.edu.tw

(Received November 1998; revised and accepted April 2000)

Abstract—In this paper, a new multiplier method that decomposes variable space into decomposed spaces is introduced. This method allows constrained minimization problems to be decomposed into subproblems. A potential constraint strategy that uses only part of the constraint set in the decomposed-space subproblems is also presented to increase the efficiency of this new space-decomposition multiplier method. Three examples are given to demonstrate this method and the potential constraint strategy. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords—Constrained minimization, Decomposition method, Multiplier method.

1. INTRODUCTION

In this paper, a space-decomposition multiplier (SDMP) method is proposed for solving the constrained minimization problem

$$\min_{x \in \Re^n} f(x), \tag{1}$$

subject to

$$g_j(x) \leq 0, \qquad j = 1, \dots, m, \tag{2}$$

$$h_j'(x) = 0, \qquad j' = 1, \dots, m', \tag{3}$$

where $f : \Re^n \to \Re$, $g_j : \Re^n \to \Re$, and $h_{j'} : \Re^n \to \Re$ are lower-bounded, continuous functions. The (augmented Lagrangian) multiplier methods introduced by Hestenes [1] and Powell [2] were popular for such constrained minimization problems in the 1980s. Powell showed that the multiplier method can be superior to the penalty function method [2]. The multiplier methods have continuously found their applications in neural networks for constrained problems [3], in neural networks learning rules [4,5], and in mixed-integer, discrete, and continuous optimization [6]. Recently, new

The research reported in this paper was supported under a project sponsored by the National Science Council Grant, Taiwan, R.O.C., NSC 85-2732-E-009-010.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
PII: S0898-1221(00)00255-8

penalty and multiplier methods have continuously been developed [7]. Although the multiplier method can significantly improve the efficiency of classical penalty function method, it has been shown that other higher-order algorithms, such as the sequential quadratic programming (SQP) method [8], are more efficient than the classical multiplier method [9,10]. However, despite being less efficient, the multiplier method is still superior in some ways that are listed below.

1. Constrained minimization problems can be transformed into unconstrained minimization problems using the multiplier method. Therefore, the unconstrained minimization technique, including decomposition and parallel processing techniques, can be used directly to solve constrained minimization problems.
2. In general, the multiplier method requires less computer storage space, especially when used for large-scale minimization problems.
3. Exact boundary values for $g(x)$ and $h(x)$ can be found using the multiplier method.

Combining all of these advantages and overcoming the inefficiency of the multiplier methods, the space-decomposition multiplier (SDMP) method is proposed in this paper. The SDMP method extends the classical multiplier method using the space-decomposition minimization (SDM) algorithm proposed by Liu and Tseng [11]. The SDM algorithm decomposes the original variable space $S \in \Re^n$ into subspaces, and allows the minimization problem (1) to be decomposed into subproblems that can be solved either singly on a single processor [11] or simultaneously on parallel processors [12]. The SDM algorithm is based on decomposition methods that provide a systematic approach to decompose minimization problems into small-scaled and coupled subproblems. Multilevel optimization methods, whose applications can be found in structure optimizations [13,14] and in mixed-discrete optimization problems [15], are typical hierarchic decomposition methods. It has been shown that multilevel optimization methods can decompose problems into a set of hierarchically related subproblems, while preserving the coupling among the decomposed subproblems [16]. The applications of multilevel optimization methods can be also found in neural networks learning rules [17]. Other decomposition methods that decompose minimization problems into subproblems directly have also been proposed by Kibardin [18], Mouallif, Nguyen and Strodiot [19]. Recently, Bouaricha and Moré [20] introduced partial separability for large-scale minimization problems. In these studies, the computing efficiency was shown to increase when the original minimization problems could be decomposed into subproblems.

The SDM algorithm is also based on parallel variable distribution techniques. The parallel variable distribution (PVD) algorithm, proposed by Ferris and Mangasarian [21], and further extended to inexact PVD algorithms by Solodov [22], was a method that distributes q blocks $x_1, \dots, x_q$ of variable x among q processors. These q variable blocks are communicated among processors either synchronously or asynchronously. The PVD algorithm provides a mechanism for updating coupled variables among decomposed subproblems. More recently, Fukushima [23] proposed a more general framework that was called the parallel variable transformation (PVT) algorithm. In this algorithm, the variables are transformed into spaces of smaller dimension, which altogether span the space of the original variables. Fukushima [23] showed that the PVD algorithms can be a special case of the general PVT framework.

The notation and terminology used in this paper are described as follows: $S \in \Re^n$ denotes n-dimensional Euclidean design space with ordinary inner product and associated two-norm $\|\cdot\|$, italic characters denoting variables and vectors. For a differentiable function $f : \Re^n \to \Re$, ∇f denotes the n-dimensional vector of partial derivatives with respect to x and $\nabla f_{S_i}(x_{S_i})$ denotes the n_i-dimensional vector of partial derivatives with respective to $x_{S_i} \in \Re^{n_i}$. For simplicity of notation, changes in the ordering of variables are allowed throughout this paper. Therefore, the variable vector $x \in S$ can be decomposed into subvectors. That is, $x = [x_{S_1}, \dots, x_{S_q}]$, where x_{S_i}, $i = 1, \dots, q$, are subvector or subcomponent of x.

This paper is organized as follows. In Section 2, basic definitions of the decomposed-space set, decomposed-space minimization function, and decomposed-space constraint set are given.

In Section 3, the SDM algorithm for unconstrained minimization problems is introduced. In Section 4, convergence criteria are derived. The classical multiplier method and the space-decomposition multiplier method are introduced in Sections 5 and 6, respectively. In Section 7, the potential constraint strategy that uses only a subset of the constraints is proven. Numerical results are presented in Section 8.

2. DECOMPOSED-SPACE SET

The space-decomposition minimization (SDM) algorithm [11] is a sequential algorithm that can solve minimization problems (1). The decomposed-space set and decomposed-space minimization function defined in [11] are modified as follows for this paper.

DEFINITION 2.1. NONOVERLAPPING DECOMPOSED-SPACE SET. *The original variable space S is spanned by $\{x \mid x \in \Re^n\}$. If the variable x is decomposed into $x = [x_{S_1}, \dots, x_{S_q}]$, then the decomposed space S_i, spanned by the subvector $\{x_{S_i} \mid x_{S_i} \in \Re^{n_i}$, where $\sum_{i=1}^{q} n_i = n\}$, forms a nonoverlapping decomposed-space set $\{S_1, \dots, S_q\}$. That is, $\bigcup_{i=1}^{q} S_i = S$ and $S_i \cap S_j = \emptyset$, if $i \neq j$.*

From Definition 2.1, the minimization function (1) can be decomposed as

$$f(x) = f_{S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) + f_{\bar{S}_i}\left(x_{\bar{S}_i}\right), \tag{4}$$

where $x_{\bar{S}_i}$ is the complement vector of x_{S_i}, $f_{S_i}(x_{S_i}, x_{\bar{S}_i})$ is the decomposed-space minimization function in the decomposed space S_i and $f_{\bar{S}_i}(x_{\bar{S}_i})$ is the complement decomposed-space function.

COROLLARY 2.2. *From (4), $f_{\bar{S}_i}(x_{\bar{S}_i})$ is only a function of $x_{\bar{S}_i}$. Therefore, it can be removed from the minimization subproblem if $x_{\bar{S}_i}$ is invariant in the decomposed space S_i.*

DEFINITION 2.3. DECOMPOSED-SPACE CONSTRAINT SET. *The decomposed-space constraint sets $g_{j,S_i}(x_{S_i}, x_{\bar{S}_i})$ and $h_{j',S_i}(x_{S_i}, x_{\bar{S}_i})$ are defined as*

$$g_j(x) = g_{j,S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) + g_{j,\bar{S}_i}\left(x_{\bar{S}_i}\right) \le 0, \qquad j = 1, \dots, m, \quad i = 1, \dots, q, \tag{5}$$

$$h_{j'}(x) = h_{j',S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) + h_{j',\bar{S}_i}\left(x_{\bar{S}_i}\right) = 0, \qquad j' = 1, \dots, m', \quad i = 1, \dots, q. \tag{6}$$

COROLLARY 2.4. *From Definition 2.3, the complement decomposed-space constraints $g_{j,\bar{S}_i}(x_{\bar{S}_i})$ and $h_{j',\bar{S}_i}(x_{\bar{S}_i})$ are only functions of $x_{\bar{S}_i}$; that is, if $x_{\bar{S}_i}$ is a constant vector, $g_{j,\bar{S}_i}(x_{\bar{S}_i})$ and $h_{j',\bar{S}_i}(x_{\bar{S}_i})$ will be constant values. Therefore, they need be calculated only once during the minimization process in decomposed space S_i.*

3. SPACE-DECOMPOSITION MINIMIZATION (SDM) ALGORITHM

The SDM algorithm is a sequential algorithm that can efficiently solve unconstrained minimization problems on a single processor [11]. In this section, the SDM algorithm is presented in brief without proof. In the following sections, it will be further expanded to the constrained minimization problem (1)–(3).

ALGORITHM 3.1. SPACE-DECOMPOSITION MINIMIZATION (SDM) ALGORITHM.

Step 1. Decompose the variable space into a nonoverlapping decomposed-space set $\{S_1, \dots, S_q\}$ and derive the decomposed-space minimization functions $f_{S_i}(x_{S_i}, x_{\bar{S}_i})$, for $i = 1, \dots, q$.

Step 2. Choose the starting point $x^{(1)}$, where $x^{(1)} = [x_{S_1}^{(1)}, \dots, x_{S_q}^{(1)}]$ and set $k = 1$.

Step 3. For $i = 1$ to q, solve one or more steps of the minimization subproblems $f_{S_i}(x_{S_i}^{(k)}, x_{\bar{S}_i})$ using any convergent descent algorithm that satisfies

$$-\nabla f_{S_i}\left(x_{S_i}^{(k)}, x_{\bar{S}_i}\right)^{\mathrm{T}} d_{S_i}^{(k)} \ge \sigma_1\left(\left\|\nabla f_{S_i}\left(x_{S_i}^{(k)}, x_{\bar{S}_i}\right)\right\|\right) \ge 0 \tag{7}$$

and

$$f_{S_i}\left(x_{S_i}^{(k)}, x_{\bar{S}_i}\right) - f_{S_i}\left(x_{S_i}^{(k+1)}, x_{\bar{S}_i}\right) \geq \sigma_2\left(-\nabla f_{S_i}\left(x_{S_i}^{(k)}, x_{\bar{S}_i}\right)^{\mathrm{T}} d_{S_i}^{(k)}\right) \geq 0, \tag{8}$$

where $\sigma_1(\cdot)$, $\sigma_2(\cdot)$ are forcing functions [21,23], and $d_{S_i}^{(k)}$ is the search direction in the decomposed space S_i.

Step 4. Apply convergence criterion, such as $\|\nabla f_{S_i}(x_{S_i}^{(k)}, x_{\bar{S}_i})\| \leq \varepsilon$, to all decomposed spaces S_i. If the convergence criterion has been satisfied for all decomposed spaces, the minimum solution has been found as $x^* = [x_{S_1}^*, \dots, x_{S_q}^*]$; otherwise, set $k = k + 1$ and go to Step 3.

4. CONSTRAINT VIOLATION AND CONVERGENCE CRITERIA

As shown in [24], two convergence criteria are required for the inner loop and the outer loop of multiplier methods. In the inner loop, the unconstrained minimization convergence criterion is called Lagrangian condition. In the outer loop, the constraint condition is required. For the algorithm presented in this paper, the Lagrangian condition is satisfied first, and then the constraint condition is satisfied.

If the gradient method is used for the decomposed-space subproblem in the inner loop, the Lagrangian condition can be

$$\|\nabla A\| = \sqrt{\sum_{i=1}^{q} \|\nabla A_{S_i}\|^2} \leq \varepsilon_1. \tag{9}$$

In the outer loop, the constraint violation is monitored using the violation criteria for all constraints. Therefore, a maximum constraint violation $V^{(k)}$ is defined as [25]

$$V^{(k)} = \max\left\{0; g_1, \dots, g_m; |h_1|, \dots, |h_{m'}|\right\}. \tag{10}$$

Therefore, the constraint condition for the outer loop is satisfied when

$$V^{(k)} \leq \varepsilon_2. \tag{11}$$

If only inequality constraints are presented in the constraint set, the cumulative constraint measure that permits representation of large numbers of inequality constraints by a single cumulative measure can be used. This approach is generally applied to structure optimization and is defined as [26]

$$V^{(k)} = \left(\frac{1}{\rho}\right) \ln\left[\sum_{j=1}^{m} \exp\left(\rho g_j\right)\right], \tag{12}$$

where ρ is an arbitrarily large number taken between 25 and 50.

5. AUGMENTED LAGRANGE MULTIPLIER METHOD

Multiplier methods that combine the penalty function and the Lagrange multiplier λ are briefly reviewed before describing the space-decomposition multiplier (SDMP) method. The decomposed-space augmented Lagrangian function is also illustrated in this section.

To solve the constrained minimization problems (1)–(3), the augmented Lagrangian function is defined as [27]

$$\begin{aligned} A(x) = f(x) &+ \sum_{j=1}^{m}\left[\lambda_j^{(k)}\left(g_j(x) + s_j^2\right)\right] + r^{(k)}\sum_{j=1}^{m}\left[g_j(x) + s_j^2\right]^2 \\ &+ \sum_{j'=1}^{m'}\left[\lambda_{m+j'}^{(k)} h_{j'}(x)\right] + r^{(k)}\sum_{j'=1}^{m'}\left[h_{j'}(x)\right]^2, \end{aligned} \tag{13}$$

where $\lambda_j^{(k)}$ are the Lagrange multipliers and $r^{(k)}$ is the penalty parameter in the k^{th} iteration. To avoid the necessity of having m additional slack variables s_j, the augmented Lagrangian function is shown to be equivalent to [28]

$$A(x) = f(x) + \sum_{j=1}^{m} \left[\lambda_j^{(k)} \alpha_j(x)\right] + r^{(k)} \sum_{j=1}^{m} [\alpha_j(x)]^2 + \sum_{j'=1}^{m'} \left[\lambda_{m+j'}^{(k)} h_{j'}(x)\right] + r^{(k)} \sum_{j'=1}^{m'} [h_{j'}(x)]^2, \quad (14)$$

where

$$\alpha_j(x) = \max\left\{ g_j(x), -\frac{\lambda_j^{(k)}}{2r^{(k)}} \right\}. \quad (15)$$

Therefore, the minimization solution can be found by minimizing the augmented Lagrangian function $A(x)$, with the Lagrange multipliers updated using

$$\lambda_j^{(k+1)} = \lambda_j^{(k)} + 2r^{(k)} \alpha_j(x), \qquad j = 1, \ldots, m, \quad (16)$$

and

$$\lambda_{m+j'}^{(k+1)} = \lambda_{m+j'}^{(k)} + 2r^{(k)} h_{j'}(x), \qquad j' = 1, \ldots, m'. \quad (17)$$

Then, the penalty parameter $r^{(k)}$ is updated using

$$r^{(k+1)} = cr^{(k)}, \quad (18)$$

where $c > 1$ is a constant value. The iterative process in (14)–(18) continues until convergence has been achieved.

From Definitions 2.1 and 2.3, the decomposed-space augmented Lagrangian function that will be used in the following sections can be defined as

$$\begin{aligned} A_{S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) = f_{S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) &+ \sum_{j=1}^{m} \left[\lambda_j^{(k)} \alpha_j(x_{S_i}, x_{\bar{S}_i})\right] + r^{(k)} \sum_{j=1}^{m} \left[\alpha_j(x_{S_i}, x_{\bar{S}_i})\right]^2 \\ &+ \sum_{j'=1}^{m'} \left[\lambda_{m+j'}^{(k)} h_{j'}(x_{S_i}, x_{\bar{S}_i})\right] + r^{(k)} \sum_{j'=1}^{m'} \left[h_{j'}(x_{S_i}, x_{\bar{S}_i})\right]^2, \end{aligned} \quad (19)$$

where

$$\alpha_j\left(x_{S_i}, x_{\bar{S}_i}\right) = \max\left\{ g_{j,S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) + g_{j,\bar{S}_i}\left(x_{\bar{S}_i}\right), -\frac{\lambda_j^{(k)}}{2r^{(k)}} \right\} \quad (20)$$

and

$$h_{j'}\left(x_{S_i}, x_{\bar{S}_i}\right) = h_{j',S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) + h_{j',\bar{S}_i}\left(x_{\bar{S}_i}\right). \quad (21)$$

In addition, equations (16),(17) can be modified to

$$\lambda_j^{(k+1)} = \lambda_j^{(k)} + 2r^{(k)} \alpha_j\left(x_{S_i}, x_{\bar{S}_i}\right), \qquad j = 1, \ldots, m, \quad (22)$$

$$\lambda_{m+j'}^{(k+1)} = \lambda_{m+j'}^{(k)} + 2r^{(k)} \left[h_{j',S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) + h_{j',\bar{S}_i}\left(x_{\bar{S}_i}\right)\right], \qquad j' = 1, \ldots, m'. \quad (23)$$

Therefore, the augmented Lagrangian function (14) can be rewritten as

$$A(x) = A_{S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) + f_{\bar{S}_i}\left(x_{\bar{S}_i}\right). \quad (24)$$

Since $f_{\bar{S}_i}(x_{\bar{S}_i})$ is a constant value in decomposed space S_i, it can be omitted during the minimization process. In addition, $g_{j,\bar{S}_i}(x_{\bar{S}_i})$ and $h_{j',\bar{S}_i}(x_{\bar{S}_i})$ are also constant values in (20)–(23). Therefore, they need be calculated only once during the minimization process in the decomposed space S_i.

6. SPACE-DECOMPOSITION MULTIPLIER (SDMP) METHOD

As shown in the previous sections, the space-decomposition minimization (SDM) algorithm can be applied to the multiplier method and is summarized as the space-decomposition multiplier (SDMP) method. The SDMP method for the constrained minimization problems (1)–(3) includes an outer loop and an inner loop. The outer loop provides a framework for the classical multiplier method, while the inner loop minimizes the augmented Lagrange function (19) using the SDM algorithm. This is summarized as follows.

ALGORITHM 6.1. SPACE-DECOMPOSITION MULTIPLIER (SDMP) METHOD.

OUTER LOOP.

Step 1. Decompose the variable space $S \in \Re^n$ into q nonoverlapping decomposed spaces $\{S_1, \dots, S_q\}$. Then, derive the q decomposed-space minimization function f_{S_i}, the q decomposed-space constraint sets $g_{j,S_i}(x_{S_i}, x_{\bar{S}_i}), h_{j',S_i}(x_{S_i}, x_{\bar{S}_i})$, and the q complement decomposed-space constraint sets $g_{j,\bar{S}_i}(x_{\bar{S}_i}), h_{j',\bar{S}_i}(x_{\bar{S}_i})$, where $i = 1, \dots, q$, $j = 1, \dots, m$, and $j' = 1, \dots, m'$.

Step 2. Choose the starting point $x^{(1)}$, where $x^{(1)} = [x_{S_1}^{(1)}, \dots, x_{S_q}^{(1)}]$ and set $k = 1$.

Step 3. Initialize the Lagrange multiplier $\lambda_j^{(1)}$, the penalty parameter $r^{(1)}$ and the constant value $c > 1$. In general, set $\lambda_j^{(1)} = 0$, where $j = 1, \dots, (m + m')$.

INNER LOOP.

Step 4. For $i = 1$ to q, solve the minimization solution of the augmented Lagrange function (19) using any zero or one-order convergent descent algorithm. Then, update $x_{S_i}^{(k,i)}$, which is a subvector of $x^{(k)}$; that is, $x^{(k)} = [x_{S_1}^{(k,1)}, \dots, x_{S_q}^{(k,q)}]$.

Step 5. If the Lagrangian condition (9) has been satisfied, go to Step 6 of the outer loop; otherwise, go to Step 4 of the inner loop.

Step 6. Update the penalty parameter $r^{(k)}$ by (18) and the Lagrange multiplier $\lambda_j^{(k)}$ by (22),(23).

Step 7. If the constraint condition (11) has been satisfied, the minimization solution is found as $x^* = [x_{S_1}^*, \dots, x_{S_q}^*]$; otherwise, set $k = k + 1$ and go to Step 4 of the inner loop.

7. POTENTIAL CONSTRAINT STRATEGY

Numerical algorithms that use only subsets of the constraints are said to use a potential constraint strategy [25]. The main effect of using such a strategy is on the efficiency of the iterative process. This is especially true for large and complex minimization problems that may have hundreds of constraints, but only a few constraints may be in the potential set. In this section, it will be shown that any constraint not in the potential set can be temporarily removed from the constraint set during the iterative process. This elimination of constraints can reduce the dimensions of decomposed-space subproblems and increase the efficiency of the entire algorithm. Therefore, the potential constraint strategy is highly beneficial and should be used in practical optimization applications [25].

To apply the potential constraint strategy to the SDMP method, the constraint set (2),(3) is divided into potential constraint set ζ_{S_i} and nonpotential constraint set $\bar{\zeta}_{S_i}$ in decomposed space S_i as follows:

$$\bar{\zeta}_{S_i} = \{g_j \mid g_j(x_{\bar{S}_i}), \forall j \in \{1, \dots, m\}\} \qquad \text{and} \qquad \{h_{j'} \mid h_{j'}(x_{\bar{S}_i}), \forall j' \in \{1, \dots, m'\}\} \tag{25}$$

and

$$\zeta_{S_i} = \{g_j \mid g_j(x_{\bar{S}_i}) \notin \bar{\zeta}_{S_i}, \forall j \in \{1, \dots, m\}\} \text{ and } \{h_{j'} \mid h_{j'}(x_{\bar{S}_i}) \notin \bar{\zeta}_{S_i}, \forall j' \in \{1, \dots, m'\}\}. \tag{26}$$

That is, the potential constraint set ζ_{S_i} is constructed by the constraints that are functions of x_{S_i} and/or $x_{\bar{S}_i}$. By contrast, the nonpotential constraint set $\bar{\zeta}_{S_i}$ is constructed by the constraints that are functions of $x_{\bar{S}_i}$. The potential constraint strategy can be applied to the decomposed-space augmented Lagrangian function (19), and is formulized in the following theorem.

THEOREM 7.1. *For the constrained minimization problems (1)–(3), if any constraint is not in the potential constraint set ζ_{S_i}, it can be temporarily removed from the constraint set in the decomposed space S_i.*

PROOF. From (5), if $g_j(x)$ is a function of only $x_{\bar{S}_i}$ in decomposed space S_i, then

$$g_j(x) = g_{j,\bar{S}_i}\left(x_{\bar{S}_i}\right) = \text{constant value in } S_i. \tag{27}$$

Therefore, (15) gives

$$\alpha_j(x) = \text{constant value in } S_i. \tag{28}$$

Similarly, if $h_{j'}(x)$ is a function of only $x_{\bar{S}_i}$ in the decomposed space S_i, from (6), we can also have

$$h_{j'}(x) = h_{j',\bar{S}_i}\left(x_{\bar{S}_i}\right) = \text{constant value in } S_i. \tag{29}$$

Therefore, the decomposed-space augmented Lagrangian function (19) can be rewritten as

$$\begin{aligned} A_{S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) = f_{S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) &+ \left\{ \sum_{g_j \in \zeta_{S_i}} \left[\lambda_j^{(k)} \alpha_j\left(x_{S_i}, x_{\bar{S}_i}\right)\right] + \sum_{g_j \in \bar{\zeta}_{S_i}} \left[\lambda_j^{(k)} \alpha_j\left(x_{\bar{S}_i}\right)\right] \right\} \\ &+ \left\{ \sum_{h_{j'} \in \zeta_{S_i}} \left[\lambda_{m+j'}^{(k)} h_{j'}\left(x_{S_i}, x_{\bar{S}_i}\right)\right] + \sum_{h_{j'} \in \bar{\zeta}_{S_i}} \left[\lambda_{m+j'}^{(k)} h_{j'}\left(x_{\bar{S}_i}\right)\right] \right\} \\ &+ r^{(k)} \left\{ \sum_{g_j \in \zeta_{S_i}} \left[\alpha_j\left(x_{S_i}, x_{\bar{S}_i}\right)\right]^2 + \sum_{g_j \in \bar{\zeta}_{S_i}} \left[\alpha_j\left(x_{\bar{S}_i}\right)\right]^2 \right\} \\ &+ r^{(k)} \left\{ \sum_{h_{j'} \in \zeta_{S_i}} \left[h_{j'}\left(x_{S_i}, x_{\bar{S}_i}\right)\right]^2 + \sum_{h_{j'} \in \bar{\zeta}_{S_i}} \left[h_{j'}\left(x_{\bar{S}_i}\right)\right]^2 \right\} \end{aligned} \tag{30}$$

$$\begin{aligned} = f_{S_i}\left(x_{S_i}, x_{\bar{S}_i}\right) &+ \sum_{g_j \in \zeta_{S_i}} \left[\lambda_j^{(k)} \alpha_j\left(x_{S_i}, x_{\bar{S}_i}\right)\right] + r^{(k)} \sum_{g_j \in \zeta_{S_i}} \left[\alpha_j\left(x_{S_i}, x_{\bar{S}_i}\right)\right]^2 \\ &+ \sum_{h_{j'} \in \zeta_{S_i}} \left[\lambda_{m+j'}^{(k)} h_{j'}\left(x_{S_i}, x_{\bar{S}_i}\right)\right] + r^{(k)} \sum_{h_{j'} \in \zeta_{S_i}} \left[h_{j'}\left(x_{S_i}, x_{\bar{S}_i}\right)\right]^2 + \Phi\left(x_{\bar{S}_i}\right), \end{aligned} \tag{31}$$

where

$$\begin{aligned} \Phi\left(x_{\bar{S}_i}\right) = \sum_{g_j \in \bar{\zeta}_{S_i}} \left[\lambda_j^{(k)} \alpha_j\left(x_{\bar{S}_i}\right)\right] + r^{(k)} \sum_{g_j \in \bar{\zeta}_{S_i}} \left[\alpha_j\left(x_{\bar{S}_i}\right)\right]^2 \\ + \sum_{h_{j'} \in \bar{\zeta}_{S_i}} \left[\lambda_{m+j'}^{(k)} h_{j'}\left(x_{\bar{S}_i}\right)\right] + r^{(k)} \sum_{h_{j'} \in \bar{\zeta}_{S_i}} \left[h_{j'}\left(x_{\bar{S}_i}\right)\right]^2. \end{aligned}$$

Since $x_{\bar{S}_i}$ is a constant vector in the decomposed space S_i, $\Phi(x_{\bar{S}_i})$ is also a constant value. Therefore, $\Phi(x_{\bar{S}_i})$ can be removed from (31) without affecting the minimization solutions of the decomposed-space subproblems. That is, the nonpotential constraint set $\bar{\zeta}_{S_i}$ can be temporarily eliminated from the constraint set in the decomposed space S_i. Since only a subset of the constraint set is required to evaluate the minimization solution, the efficiency of the SDMP method can be improved especially for large-scale constraint sets. ∎

From Theorem 7.1, the decomposed-space augmented Lagrangian function (19) can be rewritten as

$$\begin{aligned} A_{S_i}\left(x_{S_i},x_{\bar{S}_i}\right) = f_{S_i}\left(x_{S_i},x_{\bar{S}_i}\right) &+ \sum_{g_j\in\zeta_{S_i}}\left[\lambda_j^{(k)}\alpha_j\left(x_{S_i},x_{\bar{S}_i}\right)\right] + r^{(k)}\sum_{g_j\in\zeta_{S_i}}\left[\alpha_j\left(x_{S_i},x_{\bar{S}_i}\right)\right]^2 \\ &+ \sum_{h_{j'}\in\zeta_{S_i}}\left[\lambda_{m+j'}^{(k)}h_{j'}\left(x_{S_i},x_{\bar{S}_i}\right)\right] + r^{(k)}\sum_{h_{j'}\in\zeta_{S_i}}\left[h_{j'}\left(x_{S_i},x_{\bar{S}_i}\right)\right]^2 \end{aligned} \tag{32}$$

that can be applied to Step 4 of Algorithm 6.1.

8. IMPLICIT CONSTRAINED MINIMIZATION PROBLEM

In many mechanical and structural engineering applications, the minimization problem is the weight, mass, or material volume of the designed system. This is usually an explicit function of variables x. Implicit minimization problem, such as stress, displacement, vibration frequencies, etc., can also be treated by introducing artificial variables [25]. Therefore, a general minimization problem can be formulated as an explicit minimization function $f(x)$ satisfying the implicit constraints

$$\begin{aligned} g_j(x,U) &\le 0, \qquad j=1,\dots,m, \\ h'_j(x,U) &= 0, \qquad j'=1,\dots,m', \end{aligned} \tag{33}$$

or the explicit constraints

$$\begin{aligned} g_j(x) &\le 0, \qquad j=1,\dots,m, \\ h'_j(x) &= 0, \qquad j'=1,\dots,m', \end{aligned} \tag{34}$$

where U is an implicit function of x. In some engineering applications, such as the structural systems, the implicit variables U can be expressed as the equilibrium equation

$$K(x)U = F(x), \tag{35}$$

where $K(x)$ is a $\ell \times \ell$ stiffness matrix and $F(x)$ is an effective load vector having ℓ components. The stiffness matrix and effective load vector generally depend explicitly on the variables x.

If the one-order descent method is used to minimize the constrained minimization problem, it is necessary to evaluate the gradients of constraint functions. When the constraint functions are implicit in variables x, the special procedures for gradient evaluation are required [25]. From (33) and the chain rule of differentiation, the total derivative of g_j with respect to the variables in the decomposed-space S_i is given as

$$\frac{dg_j}{dx_{S_i}} = \frac{\partial g_j}{\partial x_{S_i}} + \frac{\partial g_j^{\mathrm{T}}}{\partial U}\frac{dU}{dx_{S_i}}, \tag{36}$$

where

$$\frac{\partial g_j}{\partial U} = \left[\frac{\partial g_j}{\partial U_1}, \frac{\partial g_j}{\partial U_2}, \dots, \frac{\partial g_j}{\partial U_\ell}\right]^{\mathrm{T}}$$

and

$$\frac{dU}{dx_{S_i}} = \left[\frac{dU_1}{dx_{S_i}}, \frac{dU_2}{dx_{S_i}}, \dots, \frac{dU_\ell}{dx_{S_i}}\right]^{\mathrm{T}}.$$

The partial derivatives $\frac{\partial g_j}{\partial x_{S_i}}$ and $\frac{\partial g_j}{\partial U}$ are easy to calculate. The $\frac{dU}{dx_{S_i}}$ can be obtained by differentiating (35). That is,

$$K(x)\frac{dU}{dx_{S_i}} = \frac{\partial F(x)}{\partial x_{S_i}} - \frac{\partial K(x)}{\partial x_{S_i}}U. \tag{37}$$

Therefore, the $\frac{dU}{dx_{S_i}}$ can be calculated by (37), because the explicit form of $K(x)$ and $F(x)$ is generally available. Then, the gradient of constraint can be calculated from (36). The gradients of the equality constraints can also be calculated by the similar procedure discussed above.

Other efficient procedures for calculating derivatives of implicit function with respect to the variables x were generally known as design sensitivity analysis [29]. Further research on the design sensitivity analysis for the SDMP method is warranted.

9. EXAMPLES AND NUMERICAL RESULTS

In this section, three constrained minimization problems are used to demonstrate the decomposed-space minimization function and the potential constraint strategy.

EXAMPLE 1. (See [30].)

$$\begin{aligned} \text{Minimize:} \quad & f(x) = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2, \\ \text{subject to:} \quad & x_1 - 2 = 0, \\ & x_3^2 + x_4^2 - 2 = 0. \end{aligned}$$

In this problem, $\{x_1, x_2, x_3, x_4\}$ spans the original design space $S \in \Re^4$. When the design space is divided into four one-dimensional decomposed spaces $S_i \in \Re^1$, $i = 1, \dots, 4$, the decomposed-space subproblems using the potential constraint strategy are as follows.

Subproblem 1. $f_{S_1}(x) = (x_1 - 1)^2$ with constraint $x_1 - 2 = 0$.
Subproblem 2. $f_{S_2}(x) = (x_2 - 2)^2$ with no constraint.
Subproblem 3. $f_{S_3}(x) = (x_3 - 3)^2$ with constraint $x_3^2 + \varphi$, where $\varphi = x_4^2 - 2$ is a constant value in S_3.
Subproblem 4. $f_{S_4}(x) = (x_4 - 4)^2$ with constraint $x_4^2 + \varphi$, where $\varphi = x_3^2 - 2$ is a constant value in S_4.

The numerical results of the SDMP method and the SQP method [31] are shown in Table 1.

Table 1. Numerical results of Examples 1 as compared to the SQP method.

Method	x_1^*	x_2^*	x_3^*	x_4^*	V^*	$f(x^*)$	T (s)
Exact Solution	2.00000	2.00000	0.84853	1.13137	—	13.8579	—
SDMP Method	1.99999	2.00000	0.84853	1.13138	1.19E − 10	13.8578	0.015
SQP Method	2.00000	1.99989	0.84852	1.13138	2.11E − 08	13.8579	0.020

EXAMPLE 2. (See [30].)

$$\begin{aligned} \text{Minimize:} \quad & f(x) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 + 90\left(x_4 - x_3^2\right)^2 + (1 - x_3)^2 \\ & \qquad + 10.1\,(x_2 - 1)^2 + 10.1\,(x_4 - 1)^2 + 19.8\,(x_2 - 1)\,(x_4 - 1), \\ \text{subject to:} \quad & -10 \le x_i \le 10, \qquad i = 1, \dots, 4. \end{aligned}$$

The original design space is decomposed as in numerical Example 1, and the decomposed-space subproblems are as follows.

Subproblem 1. $f_{S_1}(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ with constraints $-10 \le x_1 \le 10$.
Subproblem 2. $f_{S_2}(x) = 100(x_2 - x_1^2)^2 + 10.1(x_2 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$ with constraints $-10 \le x_2 \le 10$.
Subproblem 3. $f_{S_3}(x) = 90(x_4 - x_3^2)^2 + (1 - x_3)^2$ with constraints $-10 \le x_3 \le 10$.
Subproblem 4. $f_{S_4}(x) = 90(x_4 - x_3^2)^2 + 10.1(x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$ with constraints $-10 \le x_4 \le 10$.

Table 2. Numerical results of Examples 2 as compared to the SQP method.

Method	x_1^*	x_2^*	x_3^*	x_4^*	V^*	$f(x^*)$	T (s)
Exact Solution	1.00000	1.00000	1.00000	1.00000	—	0.000000	—
SDMP Method	1.00117	1.00233	0.99882	0.99767	0.00000	0.000005	0.080
SQP Method	0.99888	0.99787	1.00040	1.00083	0.00000	0.010000	0.110

The numerical results of the SDMP method and the SQP method [31] are shown in Table 2.

EXAMPLE 3. (See [30].)

$$\begin{aligned}
\text{Minimize:}\quad & f(x) = x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4,\\
\text{subject to:}\quad & 8 - x_1 - 2x_2 \geq 0,\\
& 12 - 4x_1 - x_2 \geq 0,\\
& 12 - 3x_1 - 4x_2 \geq 0,\\
& 8 - 2x_3 - x_4 \geq 0,\\
& 8 - x_3 - 2x_4 \geq 0,\\
& 5 - x_3 - x_4 \geq 0,\\
& x_i \geq 0, \qquad i = 1,\dots,4.
\end{aligned}$$

The original design space can be decomposed into two decomposed spaces that are spanned by $\{x_1, x_2\}$ and $\{x_3, x_4\}$, respectively. The decomposed-space subproblems then become the following.

$$\begin{aligned}
\text{Subproblem 1.}\quad & f(x) = x_1 - x_2 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4,\\
\text{subject to:}\quad & 8 - x_1 - 2x_2 \geq 0,\\
& 12 - 4x_1 - x_2 \geq 0,\\
& 12 - 3x_1 - 4x_2 \geq 0,\\
& x_i \geq 0, \qquad i = 1,2.\\
\text{Subproblem 2.}\quad & f(x) = -x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4,\\
\text{subject to:}\quad & 8 - 2x_3 - x_4 \geq 0,\\
& 8 - x_3 - 2x_4 \geq 0,\\
& 5 - x_3 - x_4 \geq 0,\\
& x_i \geq 0, \qquad i = 3,4.
\end{aligned}$$

Further decomposing the spaces into one-dimensional decomposed spaces, the decomposed-space subproblems become

$$\begin{aligned}
\text{Subproblem 1.}\quad & f(x) = x_1 - x_1x_3 + x_1x_4,\\
\text{subject to:}\quad & 8 - x_1 - 2x_2 \geq 0,\\
& 12 - 4x_1 - x_2 \geq 0,\\
& 12 - 3x_1 - 4x_2 \geq 0,\\
& x_1 \geq 0.
\end{aligned}$$

Subproblems 2–4 have forms similar to Subproblem 1. Since only inequality constraints are presented in this problem, the cumulative constraint measure (12) can also be applied to this problem. The numerical results as compared to the SQP method [31] are shown in Table 3.

Table 3. Numerical results of Examples 3 as compared to the SQP method.

Method	x_1^*	x_2^*	x_3^*	x_4^*	V^*	$f(x^*)$	T (s)
Exact Solution	0.0000	3.0000	0.0000	4.0000	—	−15.0000	—
SDMP Method	2.24E − 5	2.9999	−9.89E − 7	4.0000	3.99E − 09	−14.9999	0.015
SQP Method	0.0000	3.0000	0.0000	4.0000	2.45E − 10	−15.0000	0.010

The numerical results for all example problems were obtained on a Pentium 150 Mhz machine with 48 MB of RAM memory. The Lagrangian condition for the inner loop and the constraint condition for the outer loop were set as $\varepsilon_1 = 10^{-4}$ and $\varepsilon_2 = 10^{-8}$, respectively. As shown in Tables 1–3, the numerical minimization solutions from the SDMP method are similar to the solutions from the SQP method. In addition, the constraint violations and the efficiency are also equivalent to that of the SQP method.

10. CONCLUSIONS

A fundamental convergent space-decomposition multiplier method is presented in this paper for constrained minimization problems. This method allows minimization problems to be decomposed into subproblems that can be solved using zero- or one-order convergent algorithms. The constrain set can also be decomposed into potential constraint set and nonpotential constraint set. It is shown that any constraint not in the potential constraint set can be temporarily removed from the constraint set in the decomposed space. Numerical results show that the new multiplier method can perform well with the potential constraint strategy.

REFERENCES

1. M.R. Hestenes, Multiplier and gradient methods, *Journal of Optimization Theory and Applications* **4** (5), 303–320, (1969).
2. M.J.D. Powell, A method for nonlinear constraints in minimization problems, In *Optimization*, (Edited by R. Fletcher), Academic Press, New York, (1972).
3. S. Zhang and A.G. Constantinides, Lagrange programming neural networks, *IEEE Transactions on Circuits and Systems, II: Analog and Digital Signal Processing* **39** (7), 441–452, (1992).
4. C.Y. Maa and M.A. Shanblatt, A two-phase optimization neural network, *IEEE Transactions on Neural Networks* **3** (6), 1003–1009, (1992).
5. Z.L. Stan, Improving convergence and solution quality of Hopfield-type neural networks with augmented Lagrange multipliers, *IEEE Transactions on Neural Networks* **7** (6), 1507–1516, (1996).
6. B.K. Kannan and S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, *ASME Journal of Mechanical Design* **116**, 405–411, (1994).
7. A. Ben-Tal and M. Zibulevsky, Penalty/barrier multiplier methods for convex programming problems, *SIAM Journal on Optimization* **7** (2), 347–366, (1997).
8. C.H. Tseng and J.S. Arora, On implementation of computational algorithms for optimal design, 1: Preliminary investigation, 2: Extensive numerical investigation, *International Journal for Numerical Methods in Engineering* **26**, 1365–1402, (1988).
9. V.H. Nguyen and J.J. Strodiot, On the convergence rate for a penalty function method of exponential type, *Journal of Optimization Theory and Applications* **27** (4), 495–508, (1979).
10. P. Tseng and D.P. Bertsekas, On the convergence of the exponential multiplier method for convex programming, *Mathematical Programming* **60**, 1–19, (1993).
11. C.S. Liu and C.H. Tseng, Space-decomposition minimization method for large-scale minimization problems, *Computers Math. Applic.* **37** (7), 73–88, (1999).
12. C.S. Liu and C.H. Tseng, Parallel synchronous and asynchronous space-decomposition algorithms for large-scale minimization problems, *Computational Optimization and Applications*, (1998).
13. W. Xicheng, D. Kennedy and F.W. Williams, A two-level decomposition method for shape optimization of structures, *International Journal for Numerical Methods in Engineering* **40**, 75–88, (1997).
14. U. Kirsch, Two-level optimization of prestressed structures, *Engineering Structures* **19** (4), 309–317, (1997).
15. G. Thierauf and J. Cai, Parallel evolution strategy for solving structural optimization, *Engineering Structures* **19** (4), 318–324, (1997).
16. J. Sobieszczanski-Sobieski, B.B. James and M.F. Riley, Structural sizing by generalized, multilevel optimization, *AIAA Journal* **25** (1), 139–145, (1987).

17. C.S. Liu and C.H. Tseng, Two-level learning algorithm for multilayer neural networks, In *10th IEEE International Conference on Tools with Artificial Intelligence*, Taipei, Taiwan, November 10–12, 1998, pp. 97–102, (1998).
18. V.M. Kibardin, Decomposition into functions in the minimization problem, *Automation and Remote Control* **40** (1), 1311–1323, (1980).
19. K. Mouallif, V.H. Nguyen and J.-J. Strodiot, A perturbed parallel decomposition method for a class of nonsmooth convex minimization problems, *SIAM Journal on Control and Optimization* **29** (4), 829–847, (1991).
20. A. Bouaricha and J.J. Moré, Impact of partial separability on large-scale optimization, *Computational Optimization and Applications* **7** (1), 27–40, (1997).
21. M.C. Ferris and O.L. Mangasarian, Parallel variable distribution, *SIAM Journal on Optimization* **4** (4), 815–832, (1994).
22. M.V. Solodov, New inexact parallel variable distribution algorithms, *Computational Optimization and Applications* **7**, 165–182, (1997).
23. M. Fukushima, Parallel variable transformation in unconstrained optimization, *SIAM Journal on Optimization* **8**, 658–672, (1998).
24. J.T. Betts, A gradient projection-multiplier method for nonlinear programming, *Journal of Optimization Theory and Applications* **24** (4), 523–548, (1978).
25. J.S. Arora, *Introduction to Optimum Design*, pp. 355–356, pp. 387–392, McGraw-Hill, New York, (1989).
26. P. Hajela, C.L. Bloebaum and J. Sobieszczanski-Sobieski, Application of global sensitivity equations in multidisciplinary aircraft synthesis, *AIAA Journal* **27** (12), 1002–1010, (1990).
27. S.S. Rao, *Engineering Optimization: Theory and Practice*, 3rd Edition, pp. 521–530, pp. 630–635, John Wiley & Sons, New York, (1996).
28. R.T. Rockafellar, The multiplier method of Hestenes and Powell applied to convex programming, *Journal of Optimization Theory and Applications* **12** (6), 555–562, (1973).
29. H.M. Adelman and R.T. Haftka, Sensitivity anslysis of discrete structural systems, *AIAA Journal* **24** (5), 823–832, (1986).
30. W. Hock and K. Schittkowski, *Lecture Notes in Economics and Mathematical Systems 187*, Springer, Berlin, (1980).
31. C.H. Tseng, W.C. Liao and T.C. Yang, MOST 1.1 manual, Tech. Report AODL-9-96-01, Department of Mechanical Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C, (1996).