INT. J. CONTROL, 1989, VOL. 50, NO. 4, 1461–1486

# Mixed coordination method for non-linear programming problems with separable structures

JIANXIN TANG†, PETER B. LUH‡ and TSU-SHUAN CHANG§

Static optimization with linear equality constraints and separable structures is studied by using the mixed coordination method. The idea is to relax equality constraints via Lagrange multipliers, and create a hierarchy where the Lagrange multipliers and part of the decision variables are selected as high-level variables. The method was proposed about ten years ago with a simple high-level updating scheme. We show that the solution of the high-level problem is a saddle point, and the simple updating scheme has a linear convergence rate under appropriate conditions. To obtain faster convergence, the modified Newton method is adopted at the high level. There are two difficulties associated with this approach. One is how to obtain the hessian matrix in determining the Newton direction, since second-order derivatives of the objective function with respect to all high-level variables are needed. The second is when to stop in performing a line search along the Newton direction, as the high-level problem is a maxmini problem looking for a saddle point. In this paper, the hessian matrix is obtained by using a kind of sensitivity analysis. The line search stopping criterion, on the other hand, is based on the norm of the gradient vector. Extensive numerical testings show that our approach performs much better than the simple high-level updating scheme. Since the low level consists of a set of independent subproblems, this method is well suited for parallel implementation in solving large-scale problems. Simulated parallel-processing results show that our method outperforms the one-level Lagrange relaxation method for all the test problems. Furthermore, since convexification terms can be added while maintaining the separability of low-level subproblems, the method is very promising for non-convex problems.

## 1. Introduction

This paper studies static optimization with linear equality constraints and separable structures by using the mixed coordination method. The idea is to relax equality constraints via Lagrange multipliers, and create a hierarchy where the Lagrange multipliers and part of the decision variables are selected as high-level variables (coordination variables). The remaining decision variables are to be optimized at the low level, which is divided into independent subproblems according to problem structure. The method was proposed about ten years ago with a simple high-level updating scheme (Singh 1978). In this paper, we show that the solution of the high-level problem is a saddle point, the two-level problem is equivalent to the original problem, and the simple updating scheme has a linear convergence rate under appropriate conditions. However, faster convergence is essential for the method to be practical. The reason is that under the two-level structure, a high-level function evaluation generally implies solving all low-level subproblems once, and is very

Received 31 August 1988. Revised 24 November 1988.
† Division of Electrical Engineering, Alfred University, Alfred, NY 14802, U.S.A.
‡ Department of Electronics and Systems Engineering, University of Connecticut, Storrs

expensive. The simple updating scheme requires many high-level function evaluations, and thus is not efficient.

To obtain faster convergence, the modified Newton method is adopted at the high level. There are two major difficulties. The first one is how to obtain the hessian matrix in determining the Newton direction, as second-order derivatives of the objective function with respect to all coordination variables are needed. The second is when to stop in performing a line search along the Newton direction, as the high level is a maxmini problem looking for a saddle point. In this paper, the hessian matrix is obtained by using a kind of sensitivity analysis (Armacost and Fiacco 1974, Fiacco 1976). Note that the hessian matrix is with respect to coordination variables only, whose dimension is generally much lower than the dimension of the original problem. The line-search stopping criterion, on the other hand, is based on the norm of the gradient vector (Dennis and Shnabel 1983). Since the low level consists of a set of independent subproblems, this approach is well suited for parallel implementation in solving large-scale problems. As inequality constraints can be converted into equality constraints by using slack variables, the approach can also be extended to problems with inequality constraints.

For non-convex optimization problems, it is known that the lagrangian can be 'augmented' by having additional terms (for example, quadratic terms) to convexify the problem, and also to speed up convergence (see, for example, Bertsekas 1982, Luenberger 1984). Unfortunately, the augmentation process very often destroys the separability of the original formulation, thus prevents the decomposition of the low-level problem. Many researchers have been trying to overcome this difficulty via various approaches (see, for example, Bertsekas 1979, Tanikawa and Mukai 1985, 1987). Bertsekas has considered a convexification procedure in Bertsekas (1979). His approach starts with a conventional two-level Lagrange formulation where the Lagrange multipliers are selected as high-level variables. Additional variables are then created, together with the original decision variables, to form quadratic convexification terms. To preserve the separability of the problem, these additional variables are determined at an even higher level. Consequently, his approach results in three levels of optimization. Because each higher level function evaluation requires solving all the lower level problems once, this three-level optimization may not be efficient. Tanikawa and Mukai presented a two-level approach which also preserves the separability of the low-level problem (Tanikawa and Mukai 1979). In their approach, additional variables are created in forming convexification terms. To avoid three-level optimization, Lagrange multipliers are estimated at each iteration. These estimates may not be good if the initial condition is not close to the optimal solution. In Tanikawa and Makai (1985), the high level uses a gradient-type method with linear convergence rate. In a later version of their approach, the high level is changed to the Newton method (Tanikawa and Mukai 1987). However, the hessian matrix involves many high-dimensional matrix manipulations (with dimensions equal to the dimension of the original problem) and is difficult to obtain.

Although our results in this paper are mainly for convex problems, they have high potential to be extended to non-convex problems. By choosing appropriate convexification terms and selecting Lagrange multipliers and part of the decision variables as high-level variables, our method can convexify the high-level problem, speed up



In § 2, we present the problem formulation for convex optimization, and prove the equivalence of the two-level problem and the original problem. The convergence analysis of the simple updating scheme of Singh (1978) is given in § 3. In § 4, we derive the high-level hessian information and employ the modified Newton method to solve the high-level problem. Numerical testing results presented in § 5 show that our approach performs much better than the simple updating scheme, and achieves significant speed-up when the algorithm is parallelized. Finally in § 6, we indicate one way in extending this method to non-convex problems.

## 2. Problem formulation and a simple high-level updating scheme

Consider the following optimization problem:

$$\min_x \{f(x) \mid g(x) = 0\} \tag{2.1}$$

where $x \in X \subset R^n$, $f$: $R^n \to R$ and $g$: $R^n \to R^m$ $(m < n)$ are given functions. We assume that $f$ is convex and twice continuously differentiable, $g$ is linear and $X$ is a convex set. Thus this is a convex programming problem (the convexity assumption will be relaxed in § 6). We also assume that the problem is separable in the following sense:

$$f(x) = \sum_{i=1}^{Q} f_i(\xi_i) \tag{2.2 a}$$

$$g(x) = \{g_i(x), i = 1, \ldots, Q\} \tag{2.2 b}$$

and

$$g_i(x) = \begin{bmatrix} h_i(\xi_i) \\ \sum_{j=1}^{Q} g_{ij}(\xi_j) \end{bmatrix} = 0 \tag{2.2 c}$$

where $x = (\xi_1, \ldots, \xi_Q)$, $\xi_i \in \Xi_i \subset R^{ki}$, $\Xi_1 \times \Xi_2 \times \ldots \times \Xi_Q = X$, $f_i$: $R^{ki} \to R$, $g_i$: $R^n \to R^{li}$, $h_i$: $R^{ki} \to Rs^i$, $g_{ij}$: $R^{kj} \to R^n$, $s_i + t_i = l_i$, $\sum_{i=1}^{Q} k_i = n$ and $\sum_{i=1}^{Q} l_i = m$. Note that $\Xi_i$ is a convex set for $i = 1, 2, \ldots, Q$. Equation $h_i(\xi_i) = 0$ represents subproblem $i$'s local constraint which does not interact with other subproblems, whereas $\sum_{j=1}^{Q} g_{ij}(\xi_j) = 0$ is the interaction constraint. As a regularity condition, we also assume that constraint gradients $\{\nabla g_i(x), i = 1, 2, \ldots, Q\}$ are linearly independent.

Define

$$z_i = \sum_{\substack{j=1 \\ j \neq i}}^{Q} g_{ij}(\xi_j), \quad i = 1, \ldots, Q \tag{2.3}$$

as interaction variables, then constraints (2.2 $c$) can be rewritten as

$$g_i(\xi_i, z_i) = \begin{bmatrix} h_i(\xi_i) \\ g_{ii}(\xi_i) + z_i \end{bmatrix} = 0, \quad i = 1, \ldots, Q \tag{2.4}$$

Now constraints (2.4) can be viewed as local constraints, and can be dealt with in many different ways. For simple presentation, we shall relax them when we where

$$L \equiv \sum_{i=1}^{Q}\left[f_i(\xi_i) + \beta_i^{\mathrm{T}} g_i(\xi_i, z_i) + \lambda_i^{\mathrm{T}}\left(z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} g_{ij}(\xi_j)\right)\right]$$

$$= \sum_{i=1}^{Q}\left[f_i(\xi_i) + \beta_{hi}^{\mathrm{T}} h_i(\xi_i) + \beta_{gi}^{\mathrm{T}}(g_{ii}(\xi_i) + z_i) + \lambda_i^{\mathrm{T}}\left(z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} g_{ij}(\xi_j)\right)\right]$$

$$= \sum_{i=1}^{Q}\left[f_i(\xi_i) + \beta_{hi}^{\mathrm{T}} h_i(\xi_i) + \beta_{gi}^{\mathrm{T}}(g_{ii}(\xi_i) + z_i) + \lambda_i^{\mathrm{T}} z^i - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i)\right] \tag{2.6}$$

Selecting $\lambda$ and $z$ as high-level coordination variables, we are then left with $Q$ subproblems, (P-$i$), $i = 1, \ldots, Q$:

**(P-$i$):** $\max\limits_{\beta_i} \min\limits_{\xi_i} L_i$, with

$$L_i \equiv f_i(\xi_i) + \beta_{hi}^{\mathrm{T}} h_i(\xi_i) + \beta_{gi}^{\mathrm{T}}(g_{ii}(\xi_i) + z_i) + \lambda_i^{\mathrm{T}} z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i) \tag{2.7}$$

Note that subproblems are independent maxmini problems once $\lambda$ and $z$ are given. The high-level problem is to find the optimal $\lambda$ and $z$:

**(P-H):**
$$\max_{\lambda} \min_{z} L(\beta^*(\lambda, z), x^*(\lambda, z)) \tag{2.8}$$

where $\beta^*$ and $x^*$ are solutions of low-level subproblems. We have the following two theorems.

*Theorem* 2.1

The optimal solution of problem (P-H) is a unique saddle point.

*Theorem* 2.2

Problem (P-H) and problem (P) are equivalent in the sense that if $(\lambda^*, z^*)$ is the optimal solution of (P-H), then $x^*(\lambda^*, z^*)$ is the optimal solution of (P). Conversely, if $x^*$ is the optimal solution of (P) then there exists a $(\lambda^*, z^*)$ such that $x^*(\lambda^*, z^*) = x^*$ and $(\lambda^*, z^*)$ is the optimal solution of (P-H).

The proofs of Theorems 2.1 and 2.2 are included in Appendix A.

To update high-level variables, we note that the first-order necessary conditions of problem (P-H) are

$$\frac{\partial L}{\partial \lambda} = 0, \quad \frac{\partial L}{\partial z} = 0 \tag{2.9}$$

which results in

$$z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} g_{ij}(\xi_j) = 0 \tag{2.10 a}$$



Therefore one updating scheme, as suggested in Singh (1978) and Jamshidi (1983), is

$$\lambda_i^{k+1} = -\beta_{gi}^{k} \tag{2.11 a}$$

and

$$z_i^{k+1} = \sum_{\substack{j=1\\ j\neq i}}^{Q} g_{ij}(\xi_j^k) \tag{2.11 b}$$

for $i = 1, \ldots, Q$, where $k$ is the iteration index. Equation (2.11) will be called 'the simple updating (SU) scheme'.

In many cases, we need a slightly different solution procedure where the local constraints (2.4) are not relaxed at the very beginning. In this case (P) becomes

**(P′):**
$$\max_{\lambda} \min_{zx} L' \quad \text{subject to (2.4)} \tag{2.12}$$

where

$$L' \equiv \sum_{i=1}^{Q}\left[f_i(\xi_i) + \lambda_i^{\mathrm{T}} z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i)\right] \tag{2.13}$$

Low-level problem (P-$i$) becomes

**(P-$i'$):**

$$\min_{\xi_i} L_i' \quad \text{with} \quad L_i' \equiv f_i(\xi_i) + \lambda_i^{\mathrm{T}} z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i) \quad \text{subject to } g_i(\xi_i, z_i) = 0 \tag{2.14}$$

This formulation will be utilized in subsequent sections when needed.

## 3. Convergence analysis of the simple updating scheme

Equation (2.11) is simple and easy to implement. But does it converge? If it converges, what is the convergence rate? These are unsettled issues to the best of the author's knowledge (Jamishidi (1983), p. 180). We shall address these two issues in this section. To do this, we first note that this simple updating scheme is not exactly a gradient method; rather, it is a kind of direct iteration. This direct iteration can be regarded as an iterative procedure in solving simultaneous non-linear equations. It will be shown that under appropriate conditions, this simple updating scheme converges to the optimal solution linearly by using the fixed-point theorem.

From (2.7), it is clear that solutions for problem (P-$i$), $\xi_i$ and $\beta_i$, are functions of $z_i$ and $\lambda$. Thus (2.10) can be rewritten as

$$z_i - \sum^{Q} g_{ij}(\xi_j(z_j, \lambda_1, \lambda_2, \ldots, \lambda_Q)) = 0 \tag{3.1 a}$$

For notational simplicity, define

$$\phi_{i1} = -\beta_{gi}(z_i, \lambda_1, \lambda_2, \ldots, \lambda_Q) \tag{3.2 a}$$

$$\phi_{i2} = \sum_{\substack{j=1\\ j\neq i}}^{Q} g_{ij}(\xi_j(z_j, \lambda_1, \lambda_2, \ldots, \lambda_Q)) \tag{3.2 b}$$

and let $y$ denote the high-level variable with $y = (y_1, \ldots, y_Q)$, where $y_i = \begin{bmatrix} y_{i1} \\ y_{i2} \end{bmatrix}$, and

$$y_{i1} = \lambda_i \tag{3.3 a}$$

$$y_{i2} = z_i \tag{3.3 b}$$

Then (3.1) can be rewritten as

$$y_i = \phi_i(y_1, y_2, \ldots, y_Q) \quad \text{for } i = 1, 2, \ldots, Q$$

or

$$y = \phi(y) \tag{3.4}$$

Under the assumptions that $f(x)$ is convex, twice continuously differentiable and $g$ is linear, it can be shown by using the implicit function theorem that $\phi(y)$ is continuously differentiable on a region $\Omega$ ($\subset R^{2m}$) derived from the mapping (3.3). The simple updating scheme (2.11) thus boils down to solving (3.4) iteratively. To show its convergence, we first present three relevant propositions, with their proofs included in Appendix B. Similar versions of these propositions can be found in Wang (1979).

*Proposition* 3.1

Suppose that $\phi: \Omega \subset R^{2m} \to R^{2m}$ is differentiable at $y^0 \in \Omega$. Then for any $\varepsilon > 0$, there exists a neighbourhood $\Psi \subset \Omega$ of $y^0$ such that for any $y \in \Psi$, the following holds:

$$\|\phi(y) - \phi(y^0)\| \leqslant (\|D\phi(y^0)\| + \varepsilon)\|y - y^0\| \tag{3.5}$$

where $D\phi(y)$ is the jacobian of $\phi(y)$:

$$D\phi(y) = \begin{bmatrix} \dfrac{\partial \phi_1}{\partial y_1} & \dfrac{\partial \phi_1}{\partial y_Q} \\ \vdots & \vdots \\ \dfrac{\partial \phi_Q}{\partial y_1} & \dfrac{\partial \phi_Q}{\partial y_Q} \end{bmatrix} \tag{3.6}$$

*Proposition* 3.2

For any $2m \times 2m$ matrix $A$ and any $\varepsilon > 0$, there exists some norm in $R^{2m}$ such that

$$\|A\| \leqslant \rho(A) + \varepsilon \tag{3.7}$$



*Proposition* 3.3

If there exists a neighbourhood $\Phi = \{y \mid \|y - y^*\| < \delta, \delta > 0\}$ of $y^*$ and a constant $p$ $(0 < p < 1)$ such that

$$\|\phi(y) - \phi(y^*)\| \leqslant p\|y - y^*\| \tag{3.8}$$

for all $y \in \Phi$, then for any $y^0 \in \Phi$, the sequence $\{y^k\}$ formed by $y^{k+1} = \phi(y^k)$ converges to $y^*$ linearly.

Based on these three propositions, we have Theorem 3.1 below.

*Theorem* 3.1

Suppose that $y^* \in \Omega$ is the solution to (3.4). Assume that

$$\rho(D\phi(y^*)) < 1 \tag{3.9}$$

then there exists an open ball $\Phi = \{y \mid \|y - y^*\| < \delta, \delta > 0\} \subset \Omega$ such that for any $y^0 \in \Phi$, the sequence $\{y^k\}$ formed by $y^{k+1} = \phi(y^k) \in \Phi$ converges to $y^*$ linearly.

The proof of Theorem 3.1 is also included in Appendix B.

As a result (2.11) generates a linearly converging sequence if the absolute values of all eigenvalues of $D\phi(y)$ are less than 1. Furthermore, solution $y^*$ is locally unique since (3.4) in this case is actually a contraction mapping. Though the jacobian matrix $D\phi(y)$ may not be easy to obtain and the condition of Theorem 3.1 may not be easy to check, we do have a way of getting the jacobian matrix, as will be described in the next section.

## 4. High-level hessian information and the modified Newton iteration

From Theorem 2.2 we know that problems (P-H) and (P) are equivalent. Thus the convergence of the overall algorithm boils down to the convergence of the high-level approach. In § 3, we proved the linear convergence of the simple updating scheme. However, fast convergence is essential for the method to be practical. The reason is that under the two-level structure, a high-level function evaluation generally implies solving all low-level subproblems once, and is very expensive. The simple updating scheme requires many high-level function evaluations, and thus is not efficient. To obtain faster convergences, the modified Newton method (Luenberger 1984) is adopted at the high level. The modified Newton method is a modification of the standard Newton method with line searches incorporated so that it can be applied to problems with initial conditions not close to the optimal solution. The method updates variables according to

$$y^{k+1} = y^k - \alpha^k H^{-1}(y^k)\nabla L(y^k) \tag{4.1}$$

where $H$ is the hessian of $L$, $\nabla L$ is the gradient of $L$, and $0 \leqslant \alpha^k \leqslant 1$ is determined by an appropriate line search procedure. With the modified Newton method adopted at the high level, the convergence of the two-level approach is apparent. We now turn our attention to the finding of the hessian matrix of $L$.

From (2.9) and (3.1), we get

$$\left.\frac{\partial^2 L}{\partial \lambda_i \lambda z_k}\right|_{k \neq i} = \left.\frac{\partial^2 L}{\partial z_k \partial \lambda_i}\right|_{k \neq i} = -\frac{\partial g_{ik}(\xi_k)}{\partial \xi_k}\frac{\partial \xi_k}{\partial z_k} \tag{4.2 b}$$

$$\frac{\partial^2 L}{\partial \lambda_i^2} = -\sum_{\substack{j=1\\ j\neq i}}^{Q} \frac{\partial g_{ij}}{\partial \xi_j}\frac{\partial \xi_j}{\partial \lambda_i} \tag{4.2 c}$$

$$\left.\frac{\partial^2 L}{\partial \lambda_i \partial \lambda_k}\right|_{k \neq i} = -\sum_{\substack{j=1\\ j\neq i}}^{Q} \frac{\partial g_{ij}}{\partial \xi_j}\frac{\partial \xi_j}{\partial \lambda_k} \tag{4.2 d}$$

$$\frac{\partial^2 L}{\partial z_i^2} = \frac{\partial \beta_{gi}}{\partial z_i} \tag{4.2 e}$$

$$\left.\frac{\partial^2 L}{\partial z_i \partial z_k}\right|_{k \neq i} = 0 \tag{4.2 f}$$

In general, we have the hessian matrix of the following form:

$$H = \begin{bmatrix} \frac{\partial^2 L}{\partial \lambda_1^2} & I & \frac{\partial^2 L}{\partial \lambda_i \partial \lambda_2} & \frac{\partial^2 L}{\partial \lambda_1 \partial z_2} & \frac{\partial^2 L}{\partial \lambda_1 \partial z_Q} \\ I & \frac{\partial^2 L}{\partial z_1^2} & \frac{\partial^2 L}{\partial z_1 \partial \lambda_2} & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \\ \frac{\partial^2 L}{\partial z_Q \partial \lambda_1} & & & & \frac{\partial^2 L}{\partial z_Q^2} \end{bmatrix} \tag{4.3}$$

Equation (4.3) looks complicated. However, the hessian matrix is with respect to high-level coordination variables only, whose dimension is generally much lower than the dimension of the original problem. Furthermore, in many cases of interest, there is not much interaction among subproblems. Therefore, most of the components in $H$ are zero.

To evaulate the components of $H$, we need to have

$$\frac{\partial \xi_i}{\partial \lambda_j}, \quad \frac{\partial \xi_i}{\partial z_i} \quad \text{and} \quad \frac{\partial \beta_{gi}}{\partial z_i}$$

They are obtained by using a kind of sensitivity analysis based on the derivations of Armacost and Fiacco (1974) and Fiacco (1976). Consider subproblem (P-$i'$) and use the penalty function formulation as follows. Define

$$W_i(\xi_i, a_i) \equiv f_i(\xi_i) + \lambda_i z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i) + \frac{c}{2}|g_i(\xi_i, z_i)|^2 \tag{4.4}$$

where $a_i \equiv (z_i, \lambda_1, \lambda_2, \ldots, \lambda_Q)$, $c$ is the penalty coefficient updated according to an appropriate rule satisfying $c^{k+1} > c^k$ ($k$ is the iteration number), and $|g_i|^2 \equiv g_i^{\mathrm{T}} g_i$. The first-order necessary condition of the penalty method requires that

$$\nabla_{\xi_i} W_i(\xi_i^{k*}, a_i) = 0 \tag{4.5}$$



Applying the chain rule to (4.6) we have

$$\nabla_{\xi_i a_i} W_i(\xi_i^{k*}, a_i) + \nabla_{\xi_i \xi_i} W_i(\xi_i^{k*}, a_i)\frac{\partial \xi_i^{k*}}{\partial a_i} = 0 \tag{4.7}$$

Based on our convexity assumption, the matrix $\nabla_{\xi_i \xi_i} W_i$ is invertible if $c^k$ is large enough. Therefore

$$\frac{\partial \xi_i^{k*}}{\partial a_i} = -[\nabla_{\xi_i \xi_i} W_i(\xi_i^{k*}, a_i)]^{-1} \nabla_{\xi_i a_i} W_i(\xi_i^{k*}, a_i) \tag{4.8}$$

Since $\xi_i^{k*}$ approaches $\xi_i^*$ (true solution of (P-$i'$)) as $c^k$ approaches infinity (Fiacco 1976, p. 301), we finally have

$$\lim_{c^k \to \infty} \frac{\partial \xi_i^{k*}}{\partial a_i} = \frac{\partial \xi_i^*}{\partial a_i} \tag{4.9}$$

In practice, we can choose $c$ large enough. It should be noted that we do not need to solve problem (P-$i'$). We only use the penalty function formulation to find the expression for $\partial \xi_i^{k*}/\partial a_i$. In fact, one can use any method to find $\xi_i^*$, then use (4.8) and (4.9) to calculate $\partial \xi_i^*/\partial a_i$.

To calculate $\partial \beta_i^*/\partial a_i$, recall that $g_i$ is linear (or affine) in $\xi_j$ for $j = 1, \ldots, Q$ as assumed at the very beginning. It can be shown that $\beta_i^*$ can be expressed explicitly in terms of $\xi_i$ and $a_i$, i.e.

$$\beta_i^* = M(\xi_i(a_i), a_i) \tag{4.10}$$

For details of the derivation, see Appendix C. Therefore

$$\frac{\partial \beta_i^*}{\partial a_i} = \frac{dM}{d\xi_i}\frac{\partial \xi_i^*}{\partial a_i} + \frac{\partial M}{\partial a_i} \tag{4.11}$$

Note that the dimension of subproblem (P-$i$) (or (P-$i'$)) is relatively small, thus the evaluation of $(\nabla_{\xi_i \xi_i} W_i)^{-1}$ in (4.8) should not pose much difficulty. Note also that once the above derivatives are obtained, the jacobian matrix $D\phi(y)$ mentioned in Theorem 3.1 is readily available. The following example illustrates the decomposition procedure, the derivation of the high-level hessian matrix and the jacobian matrix.

*Example* 4.1

**(T$_1$):**

$$\min_x f(x) = \tfrac{1}{2}(x_1^2 + x_2^2) + 10(x_3^2 + x_4^2 + x_5^2) + x_6^2 \tag{4.12}$$

$$\begin{aligned} \text{subject to} \quad 0{\cdot}5x_1 + x_2 - 1 &= 0 \\ 2x_2 + x_3 + x_4 - 1 &= 0 \\ 0{\cdot}5x_4 + x_5 - 1 &= 0 \\ 0{\cdot}5x_5 + x_6 - 1 &= 0 \end{aligned}$$

Let $\xi_1 = (x_1, x_2, x_3)^{\mathrm{T}}$ and $\xi_2 = (x_4, x_5, x_6)^{\mathrm{T}}$ so that the problem is mapped into the structure of (2.2) as follows:

and

$$g_1 = \begin{bmatrix} 0{\cdot}5x_1 + x_2 - 1 \\ 2x_2 + x_3 + x_4 - 1 \end{bmatrix} = 0 \tag{4.15}$$

$$g_2 = \begin{bmatrix} 0{\cdot}5x_4 + x_5 - 1 \\ 0{\cdot}5x_5 + x_6 - 1 \end{bmatrix} = 0 \tag{4.16}$$

Note that there is only one interaction variable $x_4$ between the two subproblems. Following equation (2.3), we let

$$z_1 = x_4 \tag{4.17}$$

and rewrite constraint $g_1$ as (see (2.4))

$$g_1 = \begin{bmatrix} h_1 \\ g_{11} + z_1 \end{bmatrix} = \begin{bmatrix} 0{\cdot}5x_1 + x_2 - 1 \\ 2x_2 + x_3 - 1 + z_1 \end{bmatrix} = 0 \tag{4.18}$$

By relaxing constraints (4.17), (4.18) and (4.16), we have

$$\max_{\lambda,\beta} \min_{x,z_1} L = f_1(\xi_1) + f_2(\xi_2) + \beta_1^{\mathrm{T}} g_1 + \beta_2^{\mathrm{T}} g_2 + \lambda(z_1 - x_4) \tag{4.19}$$

where $\lambda$ is the Lagrange multiplier. Select $\lambda$ and $z_1$ as high-level variables, $(\mathrm{T}_1)$ can then be decomposed into two subproblems.

$$(\mathrm{T}_{11}): \quad \max_{\beta_1} \min_{\xi_1} L_1 = f_1(\xi_1) + \beta_1^{\mathrm{T}} g_1 + \lambda z_1 \tag{4.20}$$

$$(\mathrm{T}_{12}): \quad \max_{\beta_2} \min_{\xi_2} L_2 = f_2(\xi_2) + \beta_2^{\mathrm{T}} g_2 - \lambda x_4 \tag{4.21}$$

The high-level hessian matrix can be obtained as follows. According to (2.8), we have the high-level problem

$$\max_{\lambda} \min_{z_1} L \tag{4.22}$$

The first-order necessary conditions of the high-level problem are

$$\frac{\partial L}{\partial \lambda} = z_1 - x_4^* = 0 \tag{4.23 a}$$

$$\frac{\partial L}{\partial z_1} = \lambda + \beta_{g1}^* = 0 \tag{4.23 b}$$

where $x_4^*$ and $\beta_{g1}^*$ are the low-level solutions given $\lambda$ and $z_1$. Therefore, the off-diagonal components of the hessian matrix are

$$\frac{\partial^2 L}{\partial \lambda \partial z_1} = \frac{\partial^2 L}{\partial z_1 \partial \lambda} = 1 \tag{4.24}$$

and the diagonal components of the hessian matrix are



and

$$\frac{\partial^2 L}{\partial z_1^2} = \frac{\partial \beta_{g1}^*}{\partial z_1} = -20\frac{\partial x_3^*}{\partial z_1} \tag{4.26}$$

following equation (4.11).

To find $\partial x_4^*/\partial \lambda$ and $\partial x_3^*/\partial z_1$, we use equation (4.4). Let

$$W_1 \equiv f_1(\xi_1) + \lambda z_1 + \frac{c}{2}(|g_1|^2) \tag{4.27}$$

then

$$\nabla_{\xi_1} W_1 = \begin{bmatrix} x_1 + 0{\cdot}5c(0{\cdot}5x_1 + x_2 - 1) \\ x_2 + c(0{\cdot}5x_1 + x_2 - 1) + 2c(2x_2 + x_3 + z_1 - 1) \\ 20x_3 + c(2x_2 + x_3 + z_1 - 1) \end{bmatrix} \tag{4.28}$$

Obviously

$$\nabla_{\xi z_1} W_1 = \begin{bmatrix} 0 \\ 2c \\ c \end{bmatrix} \tag{4.29}$$

and

$$\nabla_{\xi_1 \xi_1} W_1 = \begin{bmatrix} 1 + 0{\cdot}25c & 0{\cdot}5c & 0 \\ 0{\cdot}5c & 1 + 5c & 2c \\ 0 & 2c & 20 + c \end{bmatrix} \tag{4.30}$$

From (4.8) and (4.9), one obtains

$$\frac{\partial x_3}{\partial z_1} = -\frac{1{\cdot}25c^2 + c}{21{\cdot}25c^2 + 106c + 20} \tag{4.31}$$

and from (4.26), one also gets

$$\frac{\partial \beta_{g1}^*}{\partial z_1} = -20 \lim_{c \to \infty} \frac{\partial x_3}{\partial z_1} = 1{\cdot}17647 \tag{4.32}$$

Similarly, $\partial x_4^*/\partial \lambda$ can be obtained as

$$\frac{\partial x_4^*}{\partial \lambda} = \lim_{c \to \infty} \frac{\partial x_4}{\partial \lambda} = 0{\cdot}0398 \tag{4.33}$$

The high-level hessian matrix is therefore

$$H = \begin{bmatrix} -0{\cdot}0398 & 1 \\ 1 & 1{\cdot}17647 \end{bmatrix} \tag{4.34}$$

The jacobian matrix can be shown to be

$$D\phi(y) = \begin{bmatrix} -0{\cdot}0398 & 0 \end{bmatrix} \tag{4.35}$$

One may notice that $H$ in the above example is not positive-definite or even positive semidefinite. In fact, it is an indefinite matrix since we are actually finding the saddle point of $L(\lambda, z)$ (see Theorem 2.1). In this case, if a line search is needed in updating $\lambda$ and $z$ along the Newton direction at the $k$th iteration, what should be the termination criterion for the line search? How are we going to compare $L(\lambda^{k+1}, z^{k+1})$ with $L(\lambda^k, z^k)$? To answer these questions, we note that the high-level first-order necessary condition is given by (2.9), or equivalently by (3.4). We can think of (3.4) as a set of simultaneous non-linear equations, and the goal of the high-level optimization is to reduce to zero the error defined by

$$e(y) \equiv y - \phi(y) \tag{4.36}$$

This in turn is equivalent to the finding of the global minimum of $\frac{1}{2}e(y)^{\mathrm{T}}e(y)$, i.e.

$$\min_y h(y) \quad \text{with} \quad h(y) \equiv \tfrac{1}{2}e(y)^{\mathrm{T}}e(y) \tag{4.37}$$

Therefore, one reasonable line-search stopping criterion is to check the value of $h(y)$. This works if $h(y)$ has no other local minimum (Dennis and Schnabel (1983), pp. 149–151), which is the case here as indicated by the following proposition.

*Proposition* 4.1

$h(y)$ has a unique minimum.

The proof is given in Appendix D assuming that $\partial^2 L/\partial z^2$ is positive definite and $\partial^2 L/\partial \lambda^2$ is negative definite (recall that problem (2.1) is a convex programming problem). It should be emphasized that we do not need to solve problem (4.47). Rather, the value of $h(y)$ is used to check for the stopping of the line search routine. In other words, if

$$h(y^{k+1}) \leqslant h(y^k) \tag{4.38}$$

is satisfied, then the line search is stopped.

## 5. Numerical results

Five functions are tested. In all the testings, low-level subproblems are solved by using the Daviden–Fletcher–Powell (DFP) method (Luenberger 1984) for a given $\beta$, and $\beta$ is updated by using the modified Newton method. For the high level, the modified Newton method is used to update $\lambda$ and $z$ with the hessian matrix derived according to § 4. In both Newton iterations, a simple line search routine is employed. The step size is initially set to 1, and reduced by half as needed until the function value decreases (or increases). The Figure shows the schematic of our algorithm.

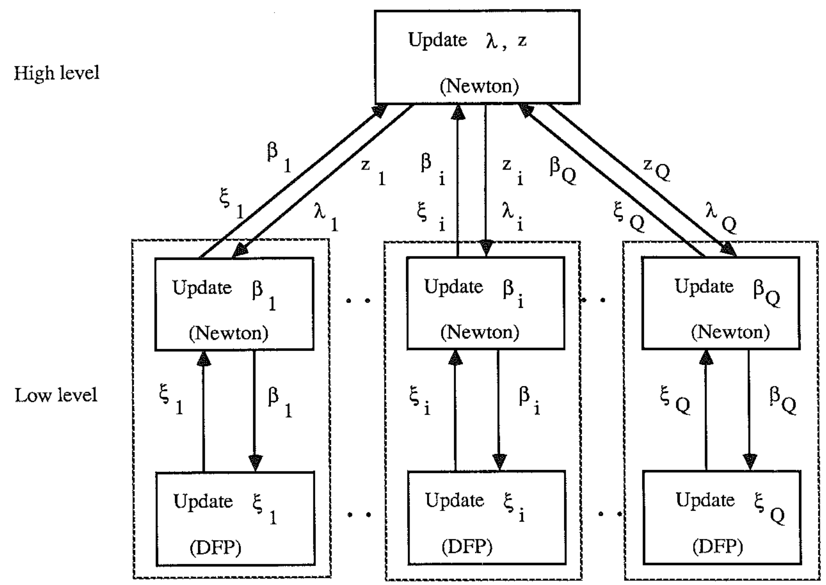
The five test functions are given below. Some of them are modifications of well known test functions. The modifications are done so that the resulting functions are separable, satisfying equation (2.2). The initial conditions used are given in Table 1.

($T_1$): A quadratic function:

$$\min f(x) = \tfrac{1}{2}(x_1^2 + x_2^2) + 10(x_3^2 + x_4^2 + x_5^2) + x_6^2$$

$$\text{subject to} \quad 0{\cdot}5x_1 + x_2 - 1 = 0$$






Algorithm schematic.

The optimal solution is

$$x^* = (1{\cdot}28833, 0{\cdot}35582, -0{\cdot}05552, 0{\cdot}34386, 0{\cdot}82807, 0{\cdot}58596)$$

with the corresponding cost

$$f^* = 9{\cdot}30676$$

($T_2$): A fourth-order polynomial. This function is a modification of ($T_1$) by replacing $x_1^2$ with $x_1^4$ and $x_6^2$ with $x_6^4$:

$$x^* = (1{\cdot}14306, 0{\cdot}42847, -0{\cdot}13864, 0{\cdot}28167, 0{\cdot}85914, 0{\cdot}57025)$$

$$f^* = 9{\cdot}41791$$

($T_3$): This function is a modified Powell function:

$$\min f = (x_1 + 10x_2)^2 + 5(x_3 + x_4)^2 + 10(x_1 - x_2)^4 + (x_3 - 2x_4)^4$$

$$\text{subject to} \quad 2x_1 + x_2 - 2 = 0$$

$$x_2 + x_3 + 4x_4 - 1 = 0$$

$$x^* = (0{\cdot}94839, 0{\cdot}10336, -0{\cdot}08899, 0{\cdot}24632)$$

$$f^* = 9{\cdot}26552$$

($T_4$): This function is a modified Wood function:

$$\min f = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + 90(x_3^2 - x_4)^2 + (x_3 - 1)^2$$

$$\text{subject to} \quad x_1 + x_2 - 1 = 0$$

($T_5$): This is a non-convex problem with a quadratic cost function but non-linear equality constraints. It is obtained by modifying the one in Rosen and Suzuki (1965, p. 113) following the modification of Tanikawa (1985, 1987). It can be shown that this problem has a unique solution (Javdan 1976).

$$\min f = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

$$\text{subject to} \quad 2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 = 0$$

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 = 0$$

$$x^* = (0, 1, 2, -1)$$

$$f^* = -44$$

Each of the five test functions is decomposed into two subproblems as its structure suggests. The decomposition of ($T_1$) has been demonstrated in Example 4.1. The testings were performed on an IBM 3090 mainframe computer on MVS. Table 1 lists the initial conditions. Table 2 provides the stopping criteria used for all the test functions, where gradient is used as the stopping criteria for $\lambda$ and $z$, and gradient and function values are used as the stopping criterion in updating $\beta$ and $\xi$. Simple line search routines are employed in updating $\lambda$, $z$ and $\beta$. Adaptive stopping criterion is used for the line search routine in updating $\xi$. $L_i^a$ and $L_i^b$ are function values corresponding to step sizes $a$ and $b$. The parameters 0·01, 0·001 and 0·0001 are picked based on numerical experience. Table 3 summarizes the test results and compares our approach with the simple updating (SU) scheme of equation (2.11).

From Table 3, we see that our approach performs much better than the SU scheme. In applying the SU scheme to ($T_5$), overflow occurs at the second high-level iteration. This may be caused by the fact that not all the eigenvalues of $D\phi(y)$ are in between $-1$ and 1 (one of the four eigenvalues equals 39·7 at the second iteration). Consequently, the iteration $y^{k+1} = \phi(y^k)$ is not a contraction mapping.

| | High level | Low level | |
|---|---|---|---|
| | | Subproblem 1 | Subproblem 2 |
| | Initial condition | Initial condition | Initial condition |
| ($T_1$) | $z^0 = 0·5$<br>$\lambda^0 = -0·5$ | $\xi_1^0 = (0, 0, 0)^T$<br>$\beta_1^0 = (1, 1)^T$ | $\xi_2^0 = (0, 0, 0)^T$<br>$\beta_2^0 = (1, 1)^T$ |
| ($T_2$) | $z^0 = 0·5$<br>$\lambda^0 = -0·5$ | $\xi_1^0 = (0, 0, 0)^T$<br>$\beta_1^0 = (1, 1)^T$ | $\xi_2^0 = (0, 0, 0)^T$<br>$\beta_2^0 = (1, 1)^T$ |
| ($T_3$) | $z^0 = 1·5$<br>$\lambda^0 = 1·5$ | $\xi_1^0 = (0, 0)^T$<br>$\beta_1^0 = 1$ | $\xi_2^0 = (0, 0)^T$<br>$\beta_2^0 = 1$ |
| ($T_4$) | $z^0 = \lambda^0 = 0·5$ | $\xi_1^0 = (0, 0)^T$<br>$\beta_1^0 = 1$ | $\xi_2^0 = (0, 0)^T$<br>$\beta_2^0 = 1$ |
| ($T_5$) | $z^0 = (2·5, 0·5)^T$ | $\xi_1^0 = (1, 1)^T$ | $\xi_2^0 = (1, 1)^T$ |



| High level | | Low level | | | |
|---|---|---|---|---|---|
| update $\lambda$, $z$ | | Update $\beta$ | | Update $\xi$ | |
| ewton iteration | For line search | For Newton iteration | For line search | For DFR | For line search |
| $L\| \leqslant 0·0001$ | $h(y^{k+1}) \leqslant h(y^k)$ | $\|\nabla_{\beta_i} L_i\| \leqslant 0·0001$ or $L_i^{k+1} - L_i^k \leqslant 0·0001$ | $L_i^{k+1} \geqslant L_i^k$ | $\|\nabla_{\xi_i} L_i\| \leqslant 0·0001$ and $\frac{\|L_i^k - L_i^{k+1}\|}{\|L_i^k\| + 0·01} \leqslant 0·001$ | $\frac{\|L_i^b - L_i^a\|}{\|L_i^a\| + 0·0001} \leqslant$ min (0·001\|$\nabla_{\xi_i} L_i$\|0·0001) |

::
L, high-level gradient (see (2.10));
, lagrangian of subproblem (see (2.7));
$_i L_i$, gradient of $L_i$ with respect to $\beta_i$;
$_{\xi_i} L_i$, gradient of $L_i$ with respect to $\xi_i$;
y) is defined by (4.13).

Table 2. Stopping criteria.

| High level | | Low level | | | | Exec. time (s) | |
|---|---|---|---|---|---|---|---|
| | | Subproblem 1 | | Subproblem 2 | | | |
| Number of ITs (Newton) | Number of ITs (SUS) | Number of ITs (Newton) | Number of ITs (SUS) | Number of ITs (Newton) | Number of ITs (SUS) | Newton | SUS |
| 1 | 6 | 4 | 13 | 5 | 14 | 0·0024 | 0·0070 |
| 2 | 6 | 10 | 27 | 12 | 28 | 0·011 | 0·028 |
| 2 | 7 | 9 | 14 | 10 | 20 | 0·0078 | 0·0155 |
| 4 | 29 | 19 | 64 | 16 | 58 | 0·020 | 0·090 |
| 11 | F† | 49 | F | 37 | F | 0·017 | F |

: iteration. (The number of low-level iterations is the number of times in going through the low-level loop (see the Figure.)
ail to converge.

Table 3. Testing results and comparison with the SU scheme.



As pointed out in § 1, low-level subproblems are independent of each other and can be solved in parallel. We now briefly examine the effects of parallelization. Because no parallel processor is available at this moment, we implemented our algorithm in a simulated parallel-processing environment. Assume for simplicity one processing element for each subproblem, and zero communication time among processors. Computation of individual subproblems are assumed to be synchronous, therefore the low-level CPU time at each iteration is calculated as the longest CPU time in solving individual subproblems for that iteration. The total parallel CPU time $T_p$ is then taken as the high-level CPU time plus the sum of low-level CPU times for all iterations. To see the significance of parallel processing, we compare it with the one-level Lagrange relaxation method (i.e. using the low-level method to solve a problem as a whole without decomposition). The performance measure speed-up ($S_p$) is adopted here. It is defined as $S_p \equiv T_s/T_p$, where $T_s$ is the one-level sequential execution time and $T_p$ is the two-level parallel execution time. It measures the improvement in computation time by using the parallel two-level algorithm over the sequential one-level algorithm. The comparison is summarized in Table 4.

| | $Q$ | Exec. time (s) One level | Exec. time (s) Two-level with parallel processing | $S_p$† |
|---|---|---|---|---|
| ($T_1$) | 2 | 0·0019 | 0·0014 | 1·357 |
| ($T_2$) | 2 | 0·011 | 0·007 | 1·571 |
| ($T_3$) | 2 | 0·006 | 0·0046 | 1·304 |
| ($T_4$) | 2 | 0·023 | 0·012 | 1·917 |
| ($T_5$) | 2 | 0·015 | 0·010 | 1·500 |

† Speed-up: defined as the ratio of the one-level sequential execution time and the two-level parallel execution time.

Table 4. Comparison of execution time between one-level and two-level with parallel processing.

It can be seen that $S_p > 1$ for all the test functions, implying that our two-level algorithm with parallel processing performs better than the one-level Lagrange relaxation method. The reason for the high speed-up for ($T_4$) is that the function without decomposition is a modified Wood function. With decomposition, each subproblem becomes a Rosenbrock-type function, which requires much fewer iterations than those of the Wood function in updating $\xi$ at the DFP level (see the Figure). From Table 4, the average speed-up is 1·53 for $Q = 2$. We believe that more reduction in execution time can be achieved by using parallel processing as $Q$ increases.

In the above tests, the high-level initial conditions are arbitrarily chosen. However, there should be a systematic way in selecting the high-level initial conditions. This issue is currently under investigation and will be reported in Tang *et al.* (1990).

convex and $g$ may not be linear. We first consider the case where each subproblem interacts only with one other subproblem as specified by

$$z_i = g_{i(i+1)}(\xi_{i+1}) \tag{6.1}$$

By adding quadratic convexification terms to (2.13) we have

$$\hat{L} \equiv \sum_{i=1}^{Q} \left[ f_i(\xi_i) + \lambda_i^{\mathrm{T}}(z_i - g_{i(i+1)}(\xi_{i+1})) + \frac{c}{2}|z_i - g_{i(i+1)}(\xi_{i+1})|^2 \right] \tag{6.2}$$

The standard approach of selecting $\{\lambda_i\}$ as high-level variables, unfortunately destroys the separabilitty of the original problem, as the cross-product term $z_i^{\mathrm{T}} g_{i(i+1)}(\xi_{i+1})$ appears. By selecting $\lambda_i$ and $z_i$ as high-level variables, the separability of the original problem is preserved. Problem (6.2) can then be decomposed into the following $Q$ subproblems:

$$\min_{\xi_i} \hat{L}_i \quad \text{with} \quad \hat{L}_i \equiv f_i(\xi_i) + \lambda_i^{\mathrm{T}} z_i - \lambda_{i-1}^{\mathrm{T}} g_{(i-1)i}(\xi_i) + \frac{c}{2}(|z_i|^2 + |g_{(i-1)i}(\xi_i)|^2)$$
$$- c z_{i-1} g_{(i-1)i}(\xi_i)$$
$$\text{subject to} \quad g_i(\xi_i, z_i) = 0 \tag{6.3}$$

with $i = 1, 2, .., Q$, and $\lambda_0 \equiv z_0 = 0$. Note that in the above procedure, only the high-level problem is convexified. Each subproblem is still a non-convex problem in general, except when all the components of $\xi_i$ are present in $g_{(i-1)i}(\xi_i)$ for $i = 1, 2, \ldots, Q$. However, these non-convex subproblems can be solved by using existing non-convex optimization methods (such as the multiplier method). For problems where a subproblem interacts with more than one subproblem, we can, in principle, create a high-level variable for each interaction as follows:

$$\left.\begin{array}{l} z_{i1} = g_{i1}(\xi_1) \\ \vdots \\ z_{i(i-1)} = g_{i(i-1)}(\xi_{i-1}) \\ z_{i(i+1)} = g_{i(i+1)}(\xi_{i+1}) \\ \vdots \\ z_{iQ} = g_{iQ}(\xi_Q) \end{array}\right\} \tag{6.4}$$

The original problem can thus be convexified and decomposed by following a similar procedure. This certainly increases the complexity of the high-level problem. Many practical 'large' problems have the nature of 'loose' interactions, thus can be structured so that each subproblem interacts only with a small number of other subproblems. For problems with strong interactions, decomposition and coordination may not be a good approach anyway.

Other issues, such as the theoretical analysis on convexification effects, convergence rate, and numerical results, etc., are currently under investigation and will be reported in the near future.



high level is a saddle point, and the two-level problem is equivalent to the original problem. Convergence analysis for the simple high-level updating scheme is presented. More importantly, we provide a mechanism to derive the high-level hessian information and also overcome the difficulty when a line search is needed during the high-level Newton iteration. Consequently, the modified Newton method can be employed at the high level. Since every high-level function evaluation generally implies solving all low-level subproblems once, the improvement on convergence rate is therefore significant. Numerical results show that our approach performs much better than the SU scheme. Furthermore, since the low level consists of a set of independent subproblems, the method is well suited for parallel processing. Simulated parallel-processing results with $Q = 2$ show that our approach outperforms the one-level Lagrange relaxation method for all the test functions. As $Q$ increases the reduction in CPU time should be more significant. Also as convexification terms can be added while maintaining the separability of low-level subproblems, the approach can be extended to non-convex optimizations. We therefore believe that this method is very promising for large-scale non-linear programming problems with separable structures.

ACKNOWLEDGMENTS

The work was supported in part by the National Science Foundation under Grants ECS-8513163, ECS-8512815, ECS-8717167 and ECS-8717235. The authors would like to thank Professor H. Mukai of Washington University and Professor Shi-Chung Chang of National Taiwan University for valuable inputs.

**Appendix A**

*Proofs of Theorems 2.1 and 2.2*

A.1. *Proof of Theorem 2.1*

We first consider the variable $z$ in (P-H) (with $\lambda$ fixed), and show that $L$ is convex in $z$. Since the cost function in (P-H) is of additive form as given by (2.6), it is sufficient to show that $L_i^*(\lambda, z_i)$ is convex in $z_i$ with $\lambda$ fixed, where $L_i^*(\lambda, z_i)$ is the solution of (2.7).

From (2.4) we know

$$g_{ii}(\xi_i) + z_i = 0$$

Since $g_{ii}(\xi_i)$ is linear in $\xi_i$, we let

$$g_{ii}(\xi_i) = C_{ii}\xi_i \tag{A 1}$$

where $C_{ii}$ is an $s_i \times s_i$ matrix. Then

$$z_i = -C_{ii}\xi_i \tag{A 2}$$

Define the set

$$Z_i \equiv \{z_i | (\text{A 2}) \text{ holds}\} \tag{A 3}$$

Since $\xi_i \in \Xi_i$, with $\Xi_i$ being a convex set, $z_i \in Z_i$ is therefore also a convex set (Luenberger 1984, p. 464).

for an arbitrary $\alpha \in [0, 1]$ and arbitrary $z_i^1$ and $z_i^2$. Let $\xi_i^{*1}$ and $\xi_i^{*2}$ be the optimal solution of (P-$i'$) for the given $z_i^1$ and $z_i^2$, respectively. Note that $z_i^1$ and $\xi_i^{*1}$ satisfy (A2). Similarly for $z_i^2$ and $\xi_i^{*2}$. Then define

$$\xi_i^0 \equiv \alpha\xi_i^{*1} + (1-\alpha)\xi_i^{*2}, \quad \xi_i^{*1}, \xi_i^{*2} \in \Xi_i \tag{A 4 b}$$

Note that $z_i^0 \in Z_i$, $\xi_i^0 \in \Xi_i$, and they also satisfy (A 2). Let $\xi_i^{*0}$ be the optimal solution of (P-$i'$) for the given $z_i^0$. To show that $L_i^*(\lambda, z_i)$ is convex in $z_i$, we need to show

$$L_i'^*(\lambda, z_i^0) \leqslant \alpha L_i'^*(\lambda, z_i^1) + (1-\alpha) L_i'^*(\lambda, z_i^2) \tag{A 5}$$

This can be shown as follows:

$$\begin{aligned}
L_i'^*(\lambda, z_i^0) &= \min_{\xi_i}\left[ f_i(\xi_i) + \lambda_i^{\mathrm{T}} z_i^0 - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i)\right] \quad \text{subject to} \quad g_i(\xi_i, z_i^0) = 0\\
&= f_i(\xi_i^*(\lambda, z_i^0)) + \lambda_i^{\mathrm{T}} z_i^0 - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i^*(\lambda, z_i^0))\\
&= f_i(\xi_i^{*0}) + \lambda_i^{\mathrm{T}} z_i^0 - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i^{*0})\\
&\leqslant f_i(\xi_i^0) + \lambda_i^{\mathrm{T}} z_i^0 - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i^0)\\
&\leqslant \alpha f_i(\xi_i^{*1}) + (1-\alpha) f_i(\xi_i^{*2}) + \lambda_i^{\mathrm{T}}(\alpha z_i^1 + (1-\alpha) z_i^2)\\
&\quad - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_i^{\mathrm{T}}(\alpha g_{ji}(\xi_i^{*1}) + (1-\alpha) g_{ji}(\xi_i^{*2}))\\
&= \alpha L_i'^*(\lambda, z_i^1) + (1-\alpha) L_i'^*(\lambda, z_i^2)
\end{aligned}$$

The first inequality holds because $\xi_i^{*0}$ is the minimum point given $z_i^0$. The second inequality holds because of the convexity of $f_i$. Since $L_i'^*$ is convex in $z_i$ and the constraint relaxed by using $\lambda$ is linear in $z_i$ (equation (2.3)), we thus conclude that the optimal solution of (P-H) is a unique saddle point. □

A.2. *Proof of Theorem 2.2*

Problem (P) is equivalent to minimizing (2.2 *a*) with respect to $z$ and $x$ subject to constraints (2.3) and (2.4), i.e.

**(P″):**

$$\min_{z,x} \sum_{i=1}^{Q} f_i(\xi_i) \quad \text{subject to} \quad z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} g_{ij}(\xi_j) = 0 \tag{2.3}$$

$$g_i(\xi_i, z_i) = 0, \quad i = 1, \ldots, Q \tag{2.4}$$



relaxing constraint (2.3) first, we have

$$\begin{aligned}
&\max_{\lambda} \min_{z} \min_{x} \sum_{i=1}^{Q}\left[ f_i(\xi_i) + \lambda_i^{\mathrm{T}}\left( z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} g_{ij}(\xi_j)\right)\right]\\
&\qquad\qquad \text{subject to} \quad g_i(\xi_i, z_i) = 0, \quad i = 1, \ldots, Q\\
&= \max_{\lambda} \min_{z}\left[ \sum_{i=1}^{Q} \min_{\xi_i}\left[ f_i(\xi_i) + \lambda_i^{\mathrm{T}} z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i)\right]\right]\\
&\qquad\qquad \text{subject to} \quad g_i(\xi_i, z_i) = 0, \quad i = 1, \ldots, Q\\
&= \max_{\lambda} \min_{z}\left[ \sum_{i=1}^{Q} \max_{\beta_i} \min_{\xi_i}\left[ f_i(\xi_i) + \lambda_i^{\mathrm{T}} z_i - \sum_{\substack{j=1\\ j\neq i}}^{Q} \lambda_j^{\mathrm{T}} g_{ji}(\xi_i) + \beta_i^{\mathrm{T}} g_i(\xi_i, z_i)\right]\right]
\end{aligned} \tag{A 6}$$

The outer maxmini problem is actually problem (P-H). From Theorem 2.1 we know that the optimal solution of (P-H) is a unique saddle point. On the other hand, since (P″) is a convex programming problem it has a unique optimal solution. Therefore, solving problem (P-H) is equivalent to solving problem (P″), and consequently is equivalent to solving problem (P) in the sense that they have the same unique optimal solution. □

## Appendix B

*Proofs of Propositions 3.1, 3.2 and 3.3 and Theorem 3.1*

B.1. *Proof of Proposition 3.1*

Define

$$q(y) = \phi(y) - D\phi(y^0)y, \quad y \in \Omega \tag{B 1}$$

The function $q(y)$ is differentiable at $y^0$ under the problem assumption. Furthermore

$$Dq(y^0) = D\phi(y^0) - D\phi(y^0) = 0 \tag{B 2}$$

Consequently, for any $\varepsilon > 0$, there exists some neighbourhood $\Psi \subset \Omega$ of $y^0$ such that

$$\|q(y) - q(y^0)\| \leqslant \varepsilon\|y - y^0\| \tag{B 3}$$

for all $y \in \Psi$. We therefore have

$$\phi(y) - \phi(y^0) = D\phi(y^0)y - D\phi(y^0)y^0 + q(y) - q(y^0) \tag{B 4}$$

The proof is completed by taking norm on both sides:

$$\|\phi(y) - \phi(y^0)\| \leqslant (\|D\phi(y^0)\| + \varepsilon)\|y - y^0\| \tag{B 5}$$

□

B.2. *Proof of Proposition 3.2*

We first note that for any $n \times n$ matrix $A$, there exists a non-singular matrix $P$ such where

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 & & 0 \\ 0 & \Lambda_2 & & . \\ . & & . & . \\ 0 & & & \Lambda_m \end{bmatrix}$$

$$\bar{I} = \begin{bmatrix} \bar{I}_1 & 0 & & 0 \\ 0 & \bar{I}_2 & & . \\ . & & . & . \\ 0 & & & \bar{I}_m \end{bmatrix}$$

$$\Lambda_i = \begin{bmatrix} \lambda_i & 0 & & 0 \\ 0 & \lambda_i & & . \\ . & & . & . \\ 0 & & & \lambda_i \end{bmatrix}$$

and

$$\bar{I}_i = \begin{bmatrix} 0 & 1 & . & 0 \\ 0 & 0 & 1 & . \\ . & & . & 1 \\ 0 & & & 0 \end{bmatrix}$$

Let the matrix $\tilde{D}$ be defined as

$$\tilde{D} = \operatorname{diag}(1, \varepsilon, \ldots, \varepsilon^{n-1}) \tag{B 7}$$

then

$$\tilde{D}^{-1} J \tilde{D} = \Lambda + \varepsilon \bar{I} \tag{B 7}$$

We therefore have

$$(P\tilde{D})^{-1} A (P\tilde{D}) = \Lambda + \varepsilon \bar{I} \tag{B 8}$$

Letting $S = P\tilde{D}$ and taking the 1-norm on both sides, we obtain

$$\|S^{-1}AS\|_1 \leqslant \rho(A) + \varepsilon \tag{B 9}$$

On the other hand, let $\|x\| = \|S^{-1}x\|_1$, then the following holds:

$$\|A\| = \sup_{\|x\|=1} \|Ax\| = \sup_{\|S^{-1}x\|_1=1} \|S^{-1}Ax\|_1 \tag{B 10}$$

With $y \equiv S^{-1}x$ we finally have



B.3. *Proof of Proposition 3.3*

From

$$\|y^1 - y^*\| = \|\phi(y^0) - \phi(y^*)\| \leqslant p\|y^0 - y^*\| < \delta \tag{B 12}$$

we know that $y^1$ belongs to the set $\Phi$. If there exists some $y^k$ in the set $\Phi$, then from

$$\|y^{k+1} - y^*\| = \|\phi(y^k) - \phi(y^*)\| \leqslant p\|y^k - y^*\| \leqslant \ldots \leqslant p^{k+1}\|y^0 - y^*\| < \delta \tag{B 13}$$

we know that $y^{k+1}$ belongs to the set $\Phi$. Thus we have $\{y^k\} \in \Phi$ for all $k \geqslant 0$. Since $0 < p < 1$, we then conclude that

$$\lim_{k\to\infty} y^k = y^* \tag{B 14}$$

Furthermore, from

$$\|y^{k+1} - y^*\| \leqslant p\|y^k - y^*\| \tag{B 15}$$

the sequence $\{y^k\}$ converges to $y^*$ linearly. □

B.4. *Proof of Theorem 3.1*

From Proposition 3.2, for any $\varepsilon > 0$, there exists an appropriate norm in $R^{2m}$ such that

$$\|D\phi(y^*)\| \leqslant \rho(D\phi(y^*)) + \varepsilon \tag{B 16}$$

holds. From Proposition 3.1, there exists an open ball $\Psi \subset \Omega$ such that

$$\|\phi(y) - \phi(y^*)\| \leqslant (\|D\phi(y^*)\| + \varepsilon)\,\|y - y^*\| \tag{B 17}$$

for all $y \in \Psi$. Substituting (B 16) into (B 17), we have

$$\|\phi(y) - \phi(y^*)\| \leqslant (\rho(D\phi(y^*)) + 2\varepsilon)\,\|y - y^*\| \tag{B 18}$$

Since we can select $\varepsilon$ such that $p = (\rho(D\phi(y^*)) + 2\varepsilon) < 1$, the proof is complete by using Proposition 3.3. □

## Appendix C

*Derivation of expressions for $\partial\beta_i^*/\partial a_i$*

To derive $\partial\beta_i^*/\partial a_i$, note that $g_i$ is linear (or affine) in $\xi_j$ for $j = 1, \ldots, Q$ as assumed. Equations (2.3) and (2.4) can respectively be rewritten as

$$z_i = \sum_{\substack{j=1\\ j\neq i}}^{Q} G_{ij}\xi_j + B_{ij} \tag{C 1}$$

and

$$g_i = \begin{bmatrix} h_i(\xi_i) \\ g_{ii}(\xi_i) \end{bmatrix} + \begin{bmatrix} 0 \\ z_i \end{bmatrix} = [G_{ii}\xi_i + B_{ii}] + \begin{bmatrix} 0 \\ z_i \end{bmatrix} \tag{C 2}$$

where 0 is an $s_i \times 1$ zero vector, and the dimensions of other matrices are as follows: $G_{ii}$, $l_i \times k_i$; $B_{ii}$, $l_i \times 1$; $G_{ij}$, $t_i \times k_j$; and $B_{ij}$, $t_i \times 1$. We now show that $\beta_i^*$ can be expressed explicitly in terms of $\xi_i^*$. First note that $L_i$ in (2.7) can be rewritten as

Differentiate it with respect to $\xi_i$, the first-order necessary condition of problem (P-$i$) requires that

$$\frac{\partial L_i(\beta_i^*, \xi_i^*)}{\partial \xi_i} = \frac{\partial f_i(\xi_i^*)}{\partial \xi_i} + G_{ii}^{\mathrm{T}} \beta_i^* - \sum_{\substack{j=1\\ j\neq i}}^{Q} G_{ji}^{\mathrm{T}} \lambda_j = 0 \tag{C 4}$$

where $\beta_i^*$ is part of the solution of Problem (P-$i$). As assumed in § 2 that $h_i(\xi_i)$ and $g_{ii}(\xi_i)$ are linearly independent in $\xi_i$, we know $l_i \leqslant k_i$. Two cases are discussed here.

(1) $l_i = k_i$. In this case $G_{ii}$ is a square matrix and is full rank, thus

$$\beta_i^* = (G_{ii}^{\mathrm{T}})^{-1} \left[ \sum_{\substack{j=1\\ j\neq i}}^{Q} G_{ji}^{\mathrm{T}} \lambda_j - \frac{\partial f_i(\xi_i^*)}{\partial \xi_i} \right] \equiv M_1(\xi_i(a_i), a_i) \tag{C 5}$$

(2) $l_i < k_i$. This case occurs more often than case (1) because most problems of interest have loose interactions. Since $\beta_i^*$ and $\xi_i^*$ are low-level solutions, the following $k_i + l_i$ simultaneous equations are satisfied:

$$\frac{\partial L_i(\beta_i^*, \xi_i^*)}{\partial \xi_i} = 0 \tag{C 6 a}$$

$$\frac{\partial L_i(\beta_i^*, \xi_i^*)}{\partial \beta_i} = 0 \tag{C 6 b}$$

We can therefore use any $l_i$ equations from (C 6 $a$) (or (C 4)) to express $\beta_i^*$ in terms of $\xi_i^*$, i.e.

$$\beta_i^* = (G_{ii}^{\mathrm{T}})_r^{-1} \left[ \sum_{\substack{j=1\\ j\neq i}}^{Q} (G_{ji}^{\mathrm{T}})_r \lambda_j - \left( \frac{\partial f_i(\xi_i^*)}{\partial \xi_i^*} \right)_r \right] \equiv M_2(\xi_i(a_i), a_i) \tag{C 7}$$

where $(G_{ii}^{\mathrm{T}})_r^{-1}$, $(G_{ji}^{\mathrm{T}})_r$ and $(\partial f_i(\xi_i^*)/\partial \xi_i^*)_r$ are corresponding components of the $l_i$ equations chosen arbitrarily from (A 6 $a$) with dimensions $l_i \times l_i$, $l_i \times t_j$ and $l_i \times 1$, respectively. Therefore

$$\frac{\partial \beta_i^*}{\partial a_i} = \frac{dM_1}{d\xi_i} \frac{\partial \xi_i^*}{\partial a_i} + \frac{\partial M_1}{\partial a_i} \quad \text{if } l_i = k_i \tag{C 8 a}$$

or

$$\frac{\partial \beta_i^*}{\partial a_i} = \frac{dM_2}{d\xi_i} \frac{\partial \xi_i^*}{\partial a_i} + \frac{\partial M_2}{\partial a_i} \quad \text{otherwise} \tag{C 8 b}$$

□

## Appendix D

*Proof of Proposition 4.1*

Equation (4.47) can be rewritten as

$$h(y) = \frac{1}{2} \begin{bmatrix} \lambda_1 + \beta_1 \\ z_1 - \sum_{j=2}^{Q} g_j \\ \lambda_2 + \beta_2 \end{bmatrix}^{\mathrm{T}} \cdot \begin{bmatrix} \lambda_1 + \beta_1 \\ z_1 - \sum_{j=2}^{Q} g_j \\ \lambda_2 + \beta_2 \end{bmatrix}$$



$$= \frac{1}{2} \left[ \frac{\partial L^{\mathrm{T}}}{\partial \lambda} \frac{\partial L}{\partial \lambda} + \frac{\partial L^{\mathrm{T}}}{\partial z} \frac{\partial L}{\partial z} \right] \tag{D 1}$$

The first-order necessary conditions of problem (4.47) are

$$\frac{\partial h}{\partial \lambda} = \frac{\partial L^{\mathrm{T}}}{\partial z} \frac{\partial}{\partial \lambda} \left( \frac{\partial L}{\partial z} \right) + \frac{\partial L^{\mathrm{T}}}{\partial \lambda} \frac{\partial}{\partial \lambda} \left( \frac{\partial L}{\partial \lambda} \right) = 0 \tag{D 2 a}$$

$$\frac{\partial h}{\partial z} = \frac{\partial L^{\mathrm{T}}}{\partial z} \frac{\partial}{\partial z} \left( \frac{\partial L}{\partial z} \right) + \frac{\partial L^{\mathrm{T}}}{\partial \lambda} \frac{\partial}{\partial z} \left( \frac{\partial L}{\partial \lambda} \right) = 0 \tag{D 2 b}$$

Equations (D 2) can be rewritten in matrix form as

$$\begin{bmatrix} \frac{\partial L^{\mathrm{T}}}{\partial \lambda} & \frac{\partial L^{\mathrm{T}}}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial^2 L}{\partial \lambda^2} & \frac{\partial^2 L}{\partial \lambda \partial z} \\ \frac{\partial^2 L}{\partial \lambda \partial z} & \frac{\partial^2 L}{\partial z^2} \end{bmatrix} = 0 \tag{D 3}$$

Let

$$H_1 = \begin{bmatrix} \frac{\partial^2 L}{\partial \lambda^2} & \frac{\partial^2 L}{\partial \lambda \partial z} \\ \frac{\partial^2 L}{\partial \lambda \partial z} & \frac{\partial^2 L}{\partial z^2} \end{bmatrix} \tag{D 4}$$

Comparing equations (D 4) with (4.3), we see that $H_1$ is actually the high-level hessian matrix with some rows swapped and some columns swapped. The determinant of $H_1$ is (Kailath 1980, p. 650)

$$|H_1| = \left| \frac{\partial^2 L}{\partial z^2} \right| \left| \frac{\partial^2 L}{\partial \lambda^2} - \frac{\partial^2 L^{\mathrm{T}}}{\partial z \partial \lambda} \left( \frac{\partial^2 L}{\partial z^2} \right)^{-1} \frac{\partial^2 L}{\partial z \partial \lambda} \right| \tag{D 5}$$

Under the assumption that $\partial^2 L/\partial z^2 > 0$ and $\partial^2 L/\partial \lambda^2 < 0$, we have

$$\left| \frac{\partial^2 L}{\partial z^2} \right| \neq 0$$

and

$$\left| \frac{\partial^2 L}{\partial \lambda^2} - \frac{\partial^2 L^{\mathrm{T}}}{\partial z \partial \lambda} \left( \frac{\partial^2 L}{\partial z^2} \right)^{-1} \frac{\partial^2 L}{\partial z \partial \lambda} \right| \neq 0$$

Therefore $|H_1| \neq 0$. $H_1$ is full rank, and (D 3) equals zero if and only if

$$\begin{bmatrix} \frac{\partial L}{\partial \lambda} \\ \frac{\partial L}{\partial z} \end{bmatrix} = 0 \tag{D 6}$$

Since (D 6) has a unique solution, $h(y)$ has a unique solution. □

BERTSEKAS, D. P., 1979, Convexification procedure and decomposition methods for nonconvex optimization problems. *Journal of Optimization Theory and Applications*, **29**, 169–197; 1982, *Constrained Optimization and Lagrange Multiplier Methods* (New York: Academic Press).
CHEN, C. T., 1984, *Linear System Theory and Design* (New York: Holt, Rinehart & Winston).
DENNIS, J., Jr., and SCHNABEL, R., 1983, *Numerical Methods for Unconstrained Optimization and Numerical Equations* (Englewood Cliffs, NJ: Prentice Hall).
FIACCO, A. V., 1976, Sensitivity analysis for nonlinear programming using penalty method. *Mathematical Programming*, **10**, 287–311.
JAMSHIDI, M., 1983, *Large-Scale Systems* (Amsterdam: North-Holland).
JAVDAN, M. R., 1976, Extension of dual coordination to a class of nonlinear systems. *International Journal of Control*, **24**, 551–571.
KAILATH, T., 1980, *Linear Systems* (Englewood Cliffs, NJ: Prentice Hall).
LUENBERGER, D., 1984, *Linear and Nonlinear Programming*, second edition (New York: Addison-Wesley Publishing Company, Inc.).
ROSEN, J., and SUZUKI, S., 1965, Construction of nonlinear programming test problems. *Communications of the ACM*, **8**, 113.
SINGH, M. G., and TITLI, A., 1978, *Systems: Decomposition, Optimization and Control* (Oxford: Pergamon Press).
TANG, J., LUH, P. B., and CHANG, T. S., 1990, The mixed coordination method for long horizon optimal control problems. *International Journal of Control*, **51.**
TANIKAWA, A., and MUKAI, H., 1985, A new technique for nonconvex prime–dual decomposition of a large scale separable optimization problem. *I.E.E.E. Transactions on automatic Control*, **30**, 133–143.
TANIKAWA, A., and MUKAI, H., 1987, New lagrangian function for nonconvex primal dual decomposition. *Computers and Mathematics with Applications*, **13**, 661–676.
WANG, D., 1979, *Solutions of Nonlinear Simultaneous Equations and Optimization Methods* (People's Republic of China: People's Education Publication) (in Chinese).