KYBERNETIKA — VOLUME 23 (1987), NUMBER 4

HIERARCHICAL CONTROL VIA AUGMENTED LAGRANGIANS

JAROSLAVA MIKULECKÁ

Hierarchical control structures are composed of local decision making units and a supremal unit (coordinator). The present work deals with a well known hierarchical algorithm based on linearized augmented Lagrangian. An extension of the algorithm for the case of global constraints is given. The algorithm has been applied to several problems and the execution time of the hierarchical algorithm has been compared to that of the global solution.

1. INTRODUCTION

The basic idea of hierarchical system theory is to decompose a large system into several smaller subsystems which can be analysed and optimized more easily. The optimizations of the subsystems are then coordinated in such a way that the overall system optimization is solved. The problem considered will be static optimization. There are various two-level optimization algorithms having different areas of applicability. They require different properties of the mappings and the constraint sets that describe the system and its optimization problem. In spite of their practical importance (the optimization of the steady-state is current practice of many industrial processes), only little attention has been paid to their implementation aspects and to their time complexity. In some systems, the following question can arise: Is a hierarchical method better than a global optimization from the point of view of time complexity? The answer to the last question obviously depends to a great extent on the coordination algorithm used. Some work in this field has been undertaken by Irwing, Nicholson and Sterling [3] where a comparison of the different techniques from the point of view of accuracy and computer requirements are given.

We shall be concerned with a well known hierarchical algorithm based on linearized augmented Lagrangian [2]. The local problems were optimized using the LPNLP code [4] in order to have the same theoretical background for both levels.

There had been several other reasons for using the LPNLP code:

— it solves the general constrained problem of the form: maximize a function of n

variables subject to the equality constraints, inequality constraints, and bounds on each variable;
- it can be tailored to meet particular problem needs by the selection of various algorithm parameters;
- it is relatively effective in comparison with other nonlinear programming codes [6];
- it is convenient and versatile for the user; some subroutines of the LPNLP can be used in the implementation of the coordination algorithm (involving the case of global inequality constraints).

We have applied the LPNLP code to several problems and the execution time of the hierarchical algorithm has been compared to that of a global solution (obtained by the same LPNLP code).

2. PROBLEM DESCRIPTION

Consider a system split into N subsystems (Fig. 1) with the following notation:

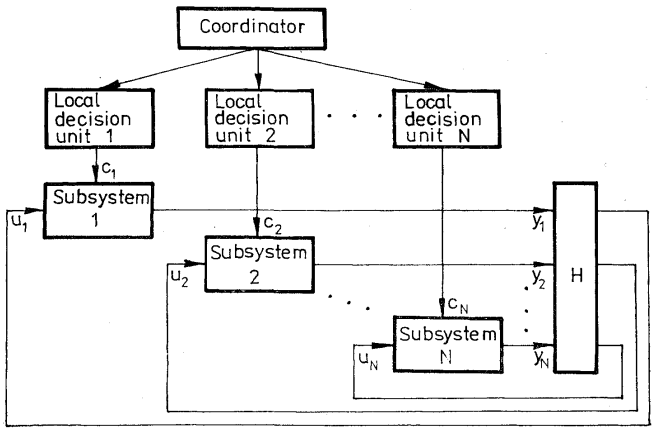

Fig. 1. Hierarchical structure for steady-state control.

u_i = vector of interaction inputs from other subsystems to the ith subsystem,
c_i = vector of manipulated variables (controls) for subsystem i,
y_i = vector of output variables for subsystem i,
H_i = ith interconnection matrix composed of zeroes and ones,
F_i = ith subsystem input-output mapping,
$y = (y_1, y_2, \ldots, y_N)$.

Then the controlled system will be described as follows

$$y_i = F_i(c_i, u_i), \quad u_i = H_i y, \quad i = 1, 2, \ldots, N \tag{2.1}$$

The couplings are separable:

$$u_i = \sum_{j=1}^{N} H_{ij} y_j$$

When we denote

$$c = (c_1, c_2, , \ldots, c_N)$$

$$u = (u_1, u_2, \ldots, u_N)$$

then

$$y = F(c, u) \quad \text{and} \quad u = Hy \tag{2.2}$$

where

$$F(c, u) = \begin{pmatrix} F_1(c_1, u_1) \\ F_2(c_2, u_2) \\ \vdots \\ F_N(c_N, u_N) \end{pmatrix} \quad \text{and} \quad H = \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_N \end{pmatrix}$$

Next we assume that the problem constraints have the following form:

– *local constraints*

$$\begin{aligned} & p_{ij}(c_i, u_i) = 0, \quad j = 1, 2, \ldots, D_{1i} \\ & q_{ij}(c_i, u_i) \leqq 0, \quad j = 1, 2, \ldots, D_{2i} \\ & a_{ik} \leqq c_{ik} \leqq b_{ik}, \quad k = 1, \ldots, \dim c_i \\ & a_{ik} \leqq u_{ik} \leqq b_{ik}, \quad k = \dim c_i + 1, \ldots, \dim c_i + \dim u_i \end{aligned} \tag{2.3}$$

– *global resource constraints*

$$\sum_{i=1}^{N} r_{ij}(c_i, u_i) - r_{0j} \leqq 0, \quad j = 1, 2, \ldots, D \tag{2.4}$$

where the a_{ik}, b_{ik} and r_{0j} are real constants. The functions p_{ij}, q_{ij} and r_{ij} can be linear or nonlinear.

We shall assume that with each subsystem a known local objective function $Q_i(c_i, u_i)$ is associated and the local decision maker will tend to maximize it with regard to local constraints only. The overall objective function is additively separable and given by

$$Q(c, u) = \sum_{i=1}^{N} Q_i(c_i, u_i) \tag{2.5}$$

In the above description the overall system is considered in the decomposed way, interactions are unbalanced, and in general

$$u \neq Hy, \quad \sum_{i=1}^{N} r_i(c_i, u_i) \nleqq r_0$$

The task of the coordinator will be to influence the local decision making in such a way as to maximize Q and simultaneously to force the interactions into balance and to satisfy the global resource constraints.

3. THE LINEARIZED AUGMENTED LAGRANGIAN METHOD

As stated above, the optimization problem is

$$\max_{(c,u)\in\mathscr{V}} \sum_{i=1}^{N} Q_i(c_i, u_i) \tag{3.1}$$

subject to

– interconnection constraints

$$u - Hy = 0 \tag{3.2}$$

– global resource constraints

$$\sum_i r_i(c_i, u_i) - r_0 \leqq 0 \tag{3.3}$$

where

$$r_i = \begin{pmatrix} r_{i1} \\ r_{i2} \\ \vdots \\ r_{iD} \end{pmatrix} \quad r_0 = \begin{pmatrix} r_{01} \\ r_{02} \\ \vdots \\ r_{0D} \end{pmatrix}$$

and $\mathscr{V}$ is the set of (c_i, u_i) satisfying the local constraints. *The augmented Lagrangian* of the above problem is given by

$$L_a = L - w_1 P_1 - w_2 P_2 - w_3 P_3 \tag{3.4}$$

where $w = (w_1, w_2, w_3)$ is a set of three penalty weights with each $w_i > 0$, α and β are Lagrange multiplier vectors and

$$L = Q - \langle \alpha, u - Hy \rangle - \langle \beta, \sum_{i=1}^{N} r_i(c_i, u_i) - r_0 \rangle \tag{3.5}$$

$$P_1 = \|u - Hy\|^2 \tag{3.6}$$

$$P_2 = \sum_{j\in C_a} \left(\sum_{i=1}^{N} r_{ij}(c_i, u_i) - r_{0j} \right)^2 ; \quad C_a = \{j: \beta_j > 0\} \tag{3.7}$$

$$P_3 = \sum_{j\in C_b} \left(\sum_{i=1}^{N} r_{ij}(c_i, u_i) - r_{0j} \right)^2 ; \quad C_b = \{j: \beta_j = 0 \quad \text{and} \quad \sum_{i=1}^{N} r_{ij} \geqq r_{0j}\} \tag{3.8}$$

With the augmented Lagrangian defined in (3.4) there is a direct relationship between the constrained local maxima of the original problem and the unconstrained local maxima of L_a. This relationship is clarified by the following theorem.

Theorem 3.1. Given $(c^*, u^*, \alpha^*, \beta^*)$ that satisfy Kuhn-Tucker relations, and given sufficiently large but finite w, $L_a(c, u, \alpha^*, \beta^*, w)$ satisfies second-order sufficient conditions for an unconstrained local maximum at (c^*, u^*), if and only if the nonlinear problem satisfies second-order sufficient conditions for a constrained local maximum at $c^*, u^*, \alpha^*, \beta^*$.

The proof is given in [4].

The augmented Lagrangian suffers from a serious drawback, namely the separa-

bility of ordinary Lagrangian has been destroyed due to the crossproduct terms in the penalty terms P_1, P_2, and P_3. The terms $\langle u, HF(c, u)\rangle$ and $[\sum_{i=1}^{N} r_{ij}(c_i, u_i)]^2$ cannot be directly transformed into a summation of "local" terms, each of which dependent only on (c_i, u_i) for some $i = 1, 2, \ldots, N$. To overcome this difficulty, Stephanopoulos and Westerberg [7] proposed the following expansion of the cross term $\langle u, HF(c, u)\rangle$ into a Taylor series around some point (c^0, u^0):

$$\langle u, HF(c, u)\rangle \doteq -\langle u^0, HF(c^0, u^0)\rangle + \langle u, HF(c^0, u^0)\rangle + \langle u^0, HF(c, u)\rangle \tag{3.9}$$

Analogically,

$$[\sum_{i=1}^{N} r_{ij}(c_i, u_i)]^2 \doteq [\sum_{i=1}^{N} r_{ij}(c_i^0, u_i^0)]^2 + \tag{3.10}$$
$$+ \sum_{i=1}^{N} 2(\sum_{i=1}^{N} r_{ij}(c_i^0, u_i^0)) \cdot [r_{ij}(c_i, u_i) - r_{ij}(c_i^0, u_i^0)]$$

Denote:

$$r_{ij}^0 = r_{ij}(c_i^0, u_i^0), \quad r_{ij} = r_{ij}(c_i, u_i), \quad F_i^0 = F_i(c_i^0, u_i^0),$$
$$F_i = F_i(c_i, u_i), \quad Q_i^0 = Q_i(c_i^0, u_i^0), \quad Q_i = Q_i(c_i, u_i)$$

Let us substitute now (3.9) and (3.10) into (3.4). Then the linearized augmented Lagrangian can be written as follows

$$L_a(c, u, \alpha, \beta, w, c^0, u^0) =$$
$$= \sum_{i=1}^{N} Q_i - \langle \alpha, u - HF\rangle - \langle \beta, \sum_i r_{ij} - r_0\rangle - w_1\{\|u\|^2 + \|HF\|^2 +$$
$$+ 2w_1\{-\langle u^0, HF^0\rangle + \langle u, HF^0\rangle + \langle u^0, HF\rangle - w_2 \sum_{j\in C_a}\{[r_{0j}^2 -$$
$$- (\sum_i r_{ij}^0)^2 + 2(\sum_i r_{ij}^0 - r_{0j}) \sum_i r_{ij} -$$
$$- w_3 \sum_{j\in C_b}\{[r_{0j}^2 - (\sum_i r_{ij}^0)^2 + 2(\sum_i r_{ij}^0 - r_{0j}) \sum_i r_{ij}\}$$

The last expression can be separated into N "local" augmented Lagrangians:

$$L_{a_i}(c_i, u_i, \alpha, \beta, w, c^0, u^0) =$$
$$= Q_i - \langle \alpha_i, u_i\rangle + \sum_{j=1}^{N} \langle \alpha_j, H_{ji}F_i\rangle - w_1\|u_i\|^2 - w_1\|F_i\|^2 -$$
$$- 2w_1[\langle u_i^0, H_iF^0\rangle - \langle u_i, H_iF^0\rangle - \sum_{j=1}^{N} \langle u_j^0, H_{ji}F_i\rangle - \tag{3.11}$$
$$- \sum_{j\in C_a} [\beta_j + 2w_2(\sum_i r_{ij}^0 - r_{0j})] r_{ij} - 2w_3 \sum_{j\in C_b} [(\sum_i r_{ij}^0) - r_{0j}] r_{ij} - K$$

where K does not depend on (c_i, u_i).

At this stage we can use the LPNLP-code (see [4]) which has the same theoretical background. The algorithm encoded uses augmented Lagrangian to solve a nonlinear optimization problem. The algorithm consists of alternating unconstrained maximization phases and update phases. During the mth maximization phase, the multipliers

α, β and penalty weights w_1, w_2, w_3 are held fixed, and a quadratically convergent search is employed to find the maximum of L_a. At the completion of the mth maximization, phase, the multipliers and weights are updated. The LPNLP-code involves a general subroutine UPDATE for updating the multipliers and weights and therefore it is possible to use this subroutine in the upper (coordination) level to adjust the multipliers (i.e., the coordination variables).

4. THE ALGORITHM

The *hierarchical* (two-level) algorithm for solving problem (3.1) with the constraints (3.2) and (3.3) is given below.

1. Set the initial values $k = 0$, c^k, u^k, α^k, β^k, w_1^k, w_2^k, w_3^k
2. On the *local level* find the maximal values (c_i^{k+1}, u_i^{k+1}), $i = 1, 2, \ldots, N$ of "local" augmented Lagrangians $L_a(c_i, u_i, \alpha^k, \beta^k, w^k, c^k, u^k)$ given by (3.11)
3. On the *upper (coordination) level* test the stop condition

$$\|c^{k+1} - c^k\| < \varepsilon_1 \wedge \|u^{k+1} - u^k\| < \varepsilon_2 \wedge \|u^{k+1} - HF(c^{k+1}, u^{k+1})\| < \varepsilon_3 \tag{4.1}$$

If the condition is not satisfied, calculate

$$\alpha^{k+1} = \alpha^k + 2w^k[u^k - HF(c^k, u^k)] \tag{4.2}$$

If $j \in C_a \equiv \{j: \beta_j^k > 0\}$ then

$$\beta_j^{k+1} = \begin{cases} 0 & \text{if} \quad \beta_j^k - 2w_2^k[r_{0j} - \sum_{i=1}^{N} r_{ij}^k] \leqq 0 \\ \beta_j^k - 2w_2^k[r_{0j} - \sum_{i=1}^{N} r_{ij}^k] & \text{otherwise} \end{cases} \tag{4.3}$$

If $j \in C_b = \{j: \beta_j^k = 0, \sum_{i=1}^{N} r_{ij}^k \geqq r_{0j}\}$ then

$$\beta_j^{k+1} = \begin{cases} 0 & \text{if} \quad r_0 - \sum_{i=1}^{N} r_{ij}^k \geqq 0 \\ -2w_3^k[r_{0j} - \sum_{i=1}^{N} r_{ij}^k] & \text{otherwise} \end{cases} \tag{4.4}$$

The w_i^k values used in (4.2), (4.3) and (4.4) are those values which were used during the preceding maximization phase. After the new β_j's are generated, the indices belonging to the sets C_b and C_a are updated, and are held fixed for the next maximization phase. The w_i's are updated after the multipliers are updated, by a factor $\gamma \geqq 1$. This procedure continues until given upper bounds $w_{i\max}$'s are reached. The penalty weight update rule then is

$$w_i^{k+1} = \begin{cases} w_{i\max}, & \text{if} \quad w_i \geqq w_{i\max}, \quad i = 1, 2, 3 \\ \gamma w_i, & \text{otherwise} \end{cases}$$

It is important to note that the initial w_i's must be large enough to prevent a constraint breakthrough from occurring.

5. COMPUTATIONAL EXPERIENCE

We have applied the algorithm given in the previous part to the following examples. Programs were written in FORTRAN IV and implemented on an M 4030 computer. All local-level problems were solved by means of the LPNLP-code [4] with the precision

$$\|(c^{k+1} - c^k) - (u^{k+1} - u^k)\| < 0{\cdot}05$$

and

$$\|\nabla L_a\| < 0{\cdot}03 \quad \text{(The gradient with respect to } (c, u)\text{.)}$$

The coordinator problem was solved with the precision 0·1.

Example 1 ([1]).

Let us consider the steady-state system shown in Figure 2. The subsystem models are as follows.

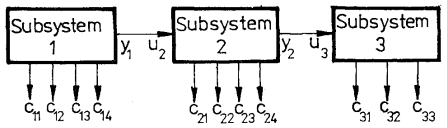

Fig. 2. System structure used in Example 1.

Subsystem 1:

$$y_1 = 8 - c_{11} - c_{12} - c_{13} - c_{14}$$

$$Q_1(c_1, u_1) = -(1 - c_{11})^2 - (2 - c_{12})^2 - (3 - c_{13})^2$$

$$0 \leqq c_{11} \leqq 1 \wedge 0 \leqq c_{12} \leqq 2 \wedge 0 \leqq c_{13} \leqq 3 \wedge 0 \leqq c_{14} \leqq 8$$

Subsystem 2:

$$y_2 = 4 + u_2 - c_{21} - c_{22} - c_{23} - c_{24}$$

$$Q_2(c_2, u_2) = -(2 - c_{21})^2 - (2{\cdot}5 - c_{22})^2 - (3{\cdot}5 - c_{23})^2$$

$$0 \leqq c_{21} \leqq 2 \wedge 0 \leqq c_{22} \leqq 2{\cdot}5 \wedge 0 \leqq c_{23} \leqq 3{\cdot}5 \wedge 0 \leqq c_{24} \leqq 4$$

Subsystem 3:

$$Q_3(c_3, u_3) = -(1 - c_{31})^2 - (1{\cdot}5 - c_{32})^2 - (2{\cdot}5 - c_{33})^2$$

$$2 + u_3 - c_{31} - c_{32} - c_{33} \geqq 0$$

$$0 \leqq c_{31} \leqq 1 \wedge 0 \leqq c_{32} \leqq 1{\cdot}5 \wedge 0 \leqq c_{33} \leqq 2{\cdot}5$$

The overall objective function is given by

$$Q(c, u) = \sum_{i=1}^{3} Q_i(c_i, u_i)$$

The system structure is given by

$$H = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

The algorithm stops after 6 upper-level calls.

Example 2 ([2]).

We consider the steady-state system shown in Figure 3. The subsystem models are as follows:

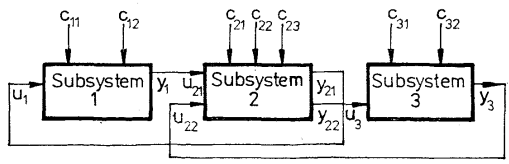

Fig. 3. System structure used in Example 2.

Subsystem 1:

$$y_1 = c_{11} - c_{12} + 2u_1$$
$$Q_1(c_1, u_1) = -(u_1 - 4)^4 - 5(c_{11} + c_{12} - 2)^2$$
$$c_{11}^2 + c_{12}^2 \leqq 1 \wedge 0 \leqq u_1 \leqq 0{\cdot}5$$

Subsystem 2:

$$y_{21} = c_{21} - c_{22} + u_{21} - 3u_{22}$$
$$y_{22} = 2c_{22} - c_{23} - u_{21} + u_{22}$$
$$Q_2(c_2, u_2) = -2(c_{21} - 2)^2 - c_{22}^2 - 3c_{23}^2 - 4u_{21}^2 - u_{22}^2$$
$$0{\cdot}5c_{11} + c_{22} + 2c_{23} \leqq 1 \wedge 4c_{21}^2 + 2c_{21} + 0{\cdot}4u_{21} + c_{21}c_{23} + 0{\cdot}5c_{23}^2 + u_{21}^2 \leqq 4$$

Subsystem 3:

$$y_3 = c_{31} + 2{\cdot}5c_{32} - 4u_3$$
$$Q_3(c_3, u_3) = -(c_{31} + 1)^2 - (u_3 - 1)^2 - 2{\cdot}5c_{32}^2$$
$$c_{31} + u_3 + 0{\cdot}5 \geqq 0 \wedge 0 \leqq c_{32} \leqq 1$$

The overal objective function is given by

$$Q(c, u) = \sum_{i=1}^{3} Q_i(c_i, u_i)$$

The system structure is given by

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The algorithm stops after 7 upper-level calls.

We have found by the global solution of the above examples by means of the

LPNLP-code with the precision

$$\|(c^{k+1} - c^k) - (u^{k+1} - u^k)\| < 0{\cdot}1$$

and

$$\|\nabla L_a\| < 0{\cdot}1 \quad \text{(the gradient with respect to } (c, u)\text{)}$$

The comparison of the execution time of the hierarchical algorithm with the global optimization is given in Table 1.

Table 1. The Execution Time for Getting Optimal Solution. N — number of variables, k — number of upper-level calls, T — average execution time in 1 local-level optimization.

	Example 1			Example 2		
	N	k	T	N	k	T
Subsystem 1	4	6	1·3	3	7	1·4
Subsystem 2	5	6	1·7	5	7	1·7
Subsystem 3	4	6	1·8	3	7	1·4
Global solution	11	—	3·4	11	—	5·5

6. CONCLUSION

If the computation speed of the hierarchical algorithm is judged from the point of view of the execution time in the slowest subsystem, then the hierarchical solution in the Example 1 will be 3 times slower, and the hierarchical solution in the Example 2 will be twice slower than the global solution.

Let us assume that the execution time of the same type problems solution (using the LPNLP-code) is approximately the linear function of the number of variables N (for greater N). The above assumption is confirmed by the test problems analysis given in [4]. If the execution time of the global solution with N variables is T, then the executive time of the problem with N/k variables will be approximately T/k. If we use the hierarchical algorithm which stops after k upper-level calls, then one local-level should not have more than N/k variables.

(Received June 5, 1986.)

REFERENCES

[1] J. Cirák: Multicriterial Control of Systems (in Slovak). Veda, Bratislava 1983.

[2] W. Findeisen, F. N. Bailey, M. Brdyś, K. Malinowski, P. Tatjewski and A. Woźniak: Control and Coordination in Hierarchical Systems. Wiley, Chichester 1980.

[3] M. R. Irwing, H. Nicholson and M. J. H. Sterling: Hierarchical Control of Electric Power Systems. In: Large Scale Systems Engineering Applications (Singh, M., Titli, A., eds.), North Holland, Amsterdam 1979, pp. 496—515.

[4] D. A. Pierre and M. J. Lowe: Mathematical Programming via Augmented Lagrangians. Addison-Wesley, Reading, Mass. 1975.

[5] P. D. Roberts, C. W. Li., I. A. Stevenson and D. S. Wadhwani: On-line distributed hierarchical control and optimization of large scale processes using a micro-computer based system. In: Real Time Control of Large Scale Systems (Balakrishnan, V., Thoma, M., eds.), Springer-Verlag, Berlin 1984.
[6] K. Schittkowski: Nonlinear Programming Codes. Springer-Verlag, Berlin 1980.
[7] G. Stephanopoulos and A. W. Westerberg: The use of Hesten's method of multipliers to resolve dual gaps in engineering systems optimization. J. Optim. Theory Appl. *15* (1975), 285—309.

RNDr. Jaroslava Mikulecká, CSc., Katedra kybernetiky SVŠT, (Department of Cybernetics — Slovak Technical University), Gottwaldovo nám. 19, 812 43 Bratislava. Czechoslovakia.