Parallel distributed optimization of discrete structures

M. Miki
Department of Knowledge Engineering, Doshisha University, Kyo-Tanabe, Kyoto, 610-03 Japan
Email: mmiki@mail.doshisha.ac.jp

Abstract

A distributed optimization method for the minimization of the total resource of a system with discrete elements is presented, and a theoretical and experimental investigations are carried out in this paper. The distributed optimization algorithm consists of two processes, namely the resource reduction process and the resource addition process. In the former process, each element discards its critical resource margin with respect to the global and local constraints, while in the latter process, a small amount of resource is added to each element. The proposed method is successively applied for optimizing truss structures, and the method is found to be very robust and suitable for parallel processing.

1 Introduction

New methods have attracted much attention in the field of optimum design in the past decade, such as genetic algorithms, simulated annealing, artificial neural networks, cellular automata, and object-oriented approaches. These methods are classified roughly into two categories, such as evolutional strategies and distributed problem solvings.

Evolutional strategies are discussed in detail in Reference 1. This type of method is based on random variations and preservation of the well adapted. These strategies are able to provide optimum or near optimum solutions even for highly nonlinear problems and extremely complicated problems. However, these methods do not provide knowledge on tactics in optimization, that is, the knowledge for changing a non-optimum solution to an optimum. Consequently, the evolutional strategies are very useful in designing structures with specific purposes and the knowledge on optimum solutions can be accumulated, but they do not provide general knowledge on optimization processes.

On the other hand, distributed problem solvings[2] are based on local interactions between the elements of a system, and there is a possibility in finding a local rule for optimizing a whole system by testing various local rules. Therefore, this approach provides much knowledge on optimization processes and optimum solutions. The knowledge obtained thus is useful in optimizing another systems.

Distributed problem solvings are considered to be useful from the following points: 1) optimization of highly nonlinear problems, 2) optimization with massively parallel computers, and 3) discovery of local rules to obtain optimum status.

The object-oriented optimization method[3] is a new optimization method using the distributed problem solving approach, in which the structural analysis and the optimization of structures are fully integrated, and the design variables are changed autonomously using local rules. However, it is difficult to use this method with parallel computers since the change in design variables have to be sequentially performed in the resource transfer and reduction processes in the method. From this point of view, the object-oriented optimization method is not a distributed optimization method.

According to recent demand for massive numerical analysis in science and technology many researches have been performed on parallel processing. The researches on parallel optimization, however, are relatively few. Those are parallel genetic algorithms, parallel simulated annealings, parallel neural networks. Further, researches on parallel processing of nonlinear optimization problems with continuous variables are few.

Those researches are reviewed in Reference 4 where the parallelization in optimization is considered as a new challenge and there are opportunities to develop new algorithms suitable for parallel processing, such as a new quasi-Newton method with parallel processing of Hessian matrix[5], a new derivative-free method[6], and a new block-truncated Newton method[7].

The purpose of this paper is to develop such a new parallel algorithm for optimization and to propose a distributed optimization method.

2 Distributed Optimization of Resource

2.1 Problems and Proposed method

Another approach which is different from those mentioned above is considered. The approach is that the distributed elements of a system change their design variables in parallel using local information and local rules being based on an objective function and constraints. This approach is a distributed problem solving and it is based on a biological metaphor for optimization.

An optimization problem of a system with discrete elements is considered in this paper. Each element has its resource, and various characteristics are manifested as functions of the resource. The aim is to minimize the total resource which is the sum of each resource, global and local constraints are imposed to the system, and the design variables are the resources of the elements. The optimization problem is then expressed as:

$$\begin{aligned}
&\text{Minimize} \quad R = \sum_{i=1}^{N} R_i \\
&\text{Subject to} \\
&g_{ik} \le 0 \quad (i = 1,\cdots,N\,;\, k = 1,\cdots,n_i) \\
&G_j \le 0 \quad (j = 1,\cdots,m)
\end{aligned} \tag{1}$$

where R is the total resource, R_i is the resource of element i, N is the number of elements, g_{ik} is the k-th local constraint for element i, and G_j is the j-th global constraint. This kind of problem is adopted in the object-oriented optimization[3]. Many optimization problem can be rewritten into this form.

This problem is solved in parallel, that is, each element of the system change its resource using its local information and local rules. An optimum solution is obtained by the iteration of this process. The information that each element can utilize is concerning to the inner state and the local constraints of the element, the global constraints of the system, and the sensitivities of those constraints with respect to the resource of the element:

$$\begin{aligned}
&S_i\,, \\
&g_{ik} \quad (k = 1,\cdots,n_i), \quad G_j \quad (j = 1,\cdots,m)\,, \\
&\frac{\partial g_{ik}}{\partial R_i}, \quad \frac{\partial G_j}{\partial R_i}
\end{aligned} \tag{2}$$

where S_i is the state of element i.

The separation of the constraints into local and global ones is important in the proposed method. Such separation is not carried out in the conventional methods since any constraint is a function of all the design variables. However, we consider such separation from the following viewpoint. That is, the constraint is considered as local when it can be satisfied by changing the resource of the element assuming that the inner state of the element is not changed. For structures, constraints on stress or buckling of a member are considered as local. On the other hand, constraints except for these local constraints are global.

The proposed method is as follows.

1) Each element evaluates its resource margins with respect to its local constraints as:

$$M_{gik} = \left(g_{ik} \bigg/ \frac{\partial g_{ik}}{\partial R_i} \right) \tag{3}$$

2) Each element evaluates its resource margins with respect to the global constraints as:

$$M_{iGj} = \alpha_{ij} \left(G_j \bigg/ \frac{\partial G_j}{\partial R_i} \right) \tag{4}$$

where a responsibility factor α is introduced since the global constraint is satisfied cooperatively with another elements. The responsibility factor is constant (=1/N) when there is no information about other elements.

3) The minimum value of the resource margins with respect to the local and global constraints is the critical resource margin of the element, and each element reduces its resource by it. The critical resource margin of element i is:

$$M_i = \mathrm{Min}(M_{gi1}, M_{gi2}, \cdots, M_{giml}, M_{iG1}, M_{iG2}, \cdots, M_{iGm}) \tag{5}$$

4) Add a small constant amount of resource to each element.

5) Iterate steps 1 - 4 until a converged solution is obtained.

This algorithm is different from the object-oriented optimization method[3] with respect to two distinguished points. One is the removal of the resource transfer and reduction process, and another is the addition of step 4, that is, the resource addition process. The resource transfer and reduction process is very expensive in computation cost since the number of the combination of a pair of elements for the resource transfer increases rapidly when the number of elements increases. Also, the parallelization of this process is difficult. A newly introduced process, the resource addition process, make such expensive resource transfer and reduction process unnecessary. We call this method the DORAR (Distributed Optimization by Resource Addition and Reduction) method.

2.2 Movement and Convergence of Design Solution

Figure 1 shows a resource plane where the number of the elements in a system is two, and the abscissa and the ordinate are the resources of elements 1 and 2, respectively. There is a global constraint, G, and the local constraints are g_{11} and

g_{12} for element 1, g_{21} and g_{22} for element 2. In the proposed method, the separation between global and local constraints allows that an element does not observe the local constraints of the other elements.

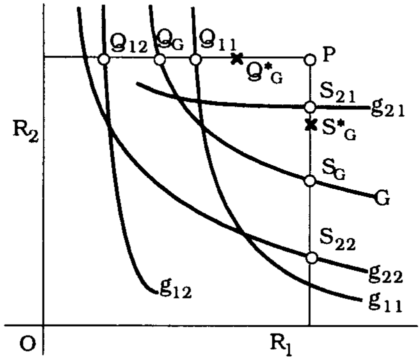


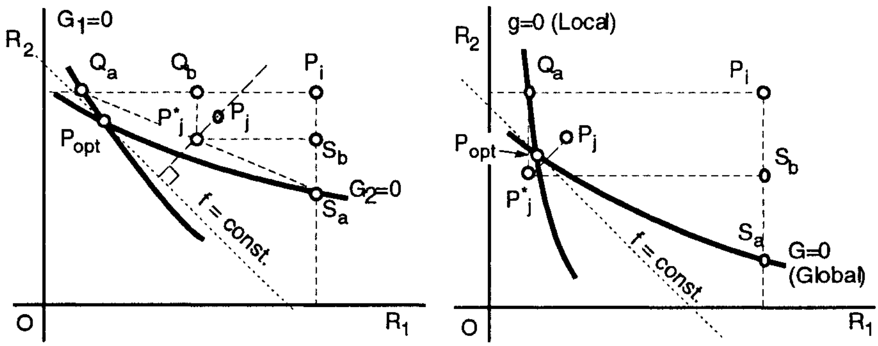
Figure 1: Resource margin.

At a design point, P, in the figure, the resource margins of element 1 for its local constraints are expressed by line segments P-Q_{11} and P-Q_{12}. For the global constraint, the resource margin of element 1 is expressed by line segment P-Q_G* where the apparent margin P-Q_G is reduced with a responsibility factor. The responsibility factor for global constraints is constant (=1/2) here. In this situation, the critical resource margin of element 1 becomes P-Q_G*, that is, the global constraint is critical for element 1. On the other hand, the resource margins of element 2 for its local constraints are expressed by line segments P-S_{21} and P-S_{22}. For the global constraint, the resource margin of element 2 is expressed by line segment P-S_G* where the apparent margin P-S_G is reduced with the responsibility factor. In this situation, the critical resource margin of element 2 becomes P-S_{21}, that is, the local constraint g_{21} is critical for element 2. Thus, a single constraint becomes critical in one element.

Figure 2 shows the movement of a design solution on a resource plane for a two-element system where the current design is represented by P_i. The effective constraints are two global constraints in Fig 2 (a), one global and one local constraints in Figure 2 (b). In Figure 2 (a), the critical resource margin of element 1 is represented by line segment P_i-Q_b, while it is Pi-Sb for element 2. These critical resource margins are discarded and then the design solution moves to P_j* which is also a mid-point of Q_a-S_a. The resource addition process adds a small amount of resource to each element so that the design point moves to P_j where direction P_j*-P_j corresponds to the direction of the gradient of the total resource. In Figure 2 (b), the critical resource margin of element 1 is represented by P_i-Q_a, while it is P_i-S_b for element 2, design solution P_i moves to P_j* after

Transactions on the Built Environment vol 28, © 1997 WIT Press, www.witpress.com, ISSN 1743-3509

the resource reduction process, then a small amount of resource is added to each element so that the design point moves to P_j.



(a) For global constraints. (b) For local and global constraints.

Figure 2: Movement of design solution.

The convergence of the design solution is shown in Figure 3. The convergence occurs when the resource reduction and the resource addition cancel each other as shown in the figure where the resource reduction vector P_i-P_j is in a reverse direction from that of the gradient vector of the total resource having the same magnitude, and design solution remains at point P_i. This figure shows a case with a single effective global constraint, and a similar conversion also occurs for a case including a local constarint.

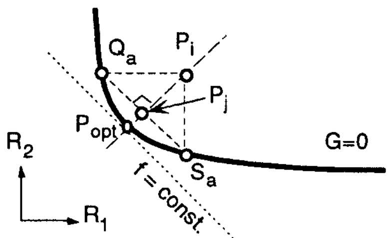


Figure 3: Convergence of design solution.

3 Results and Discussions

The DORAR method is applied to the minimum volume problem of a 11-member truss shown in Figure 4. The material used is a linear elastic body with the longitudinal modulus of 1 GPa, but the nonlinearity in its deformation is considered. The resource is the volume of truss members here. Local constraints are imposed on the tensile and compressive strength, both 40 MPa, and on the buckling, while a single global constraint is imposed on the displacement of node 6, that is, it is equal to or less than 0.03 m. The truss members have circular cross sections. The loading condition is shown in the figure. Member 1 is not subject to any load and it is meaningless, but it is provided in order to investigate the change of such elements.

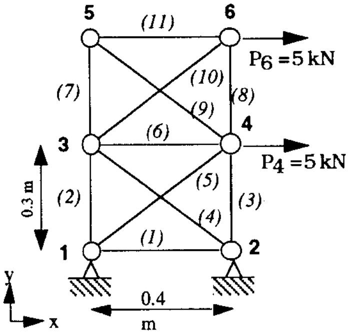


Figure 4: A 11-member truss structure.

The parallel computer used here is nCUBE 2 with 64 processors. Its processor network is hypercube, and the programming language is C++ with additional functions for processor communications. The code developed contains the object-oriented analysis of truss structures[8] and the DORAR method, called DORAR-VT (Distributed Optimization by Resource Addition and Reduction for the Volume of Trusses). In this code, each processor is assigned to each truss member, and each member changes its volume autonomously and distributedly. The evaluation of the sensitivities of its resource to the local constraints and the global constraints are performed independently in each processor assigned to each member.

The code for structural analysis in each processor is not a FEM code, but it is based on the object-oriented structural analysis[9]. This method is valid for any nonlinear behavior since it is based on the local rule at the truss node, that is, the minimization of the unbalanced nodal force. The amount of resource used for the resource addition process is 0.1 % of the total volume at that time.

Five initial configurations are created using random numbers. The minimum radius of the member is 1 mm and the maximum is 50 mm, yielding 2500-times difference in the sectional areas. The distributions of the sectional areas of the initial configurations are shown in Figure 5 where a considerable difference can be seen. A large deformation is expected in some configurations, but the object-oriented structural analysis method is able to treat any large deformation.

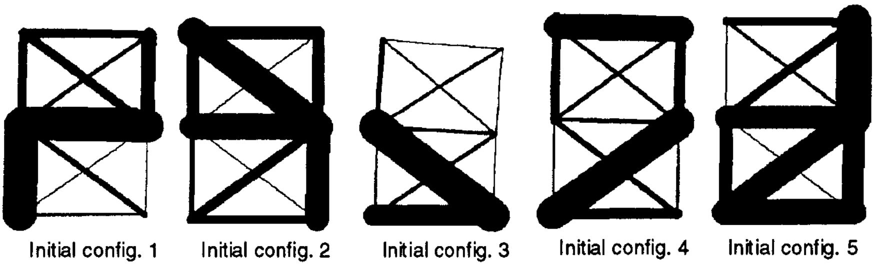


Figure 5: Distributions of the initial sectional areas of the members under loading.

The history of the total volume during the iteration in the DORAR method is shown in Figure 6. The initial volumes are decreased very quickly at the first stage and then slowly. It should be noticed that the good convergence is obtained although the initial configurations are remarkably different. Consequently, the DORAR method is found to be very robust.

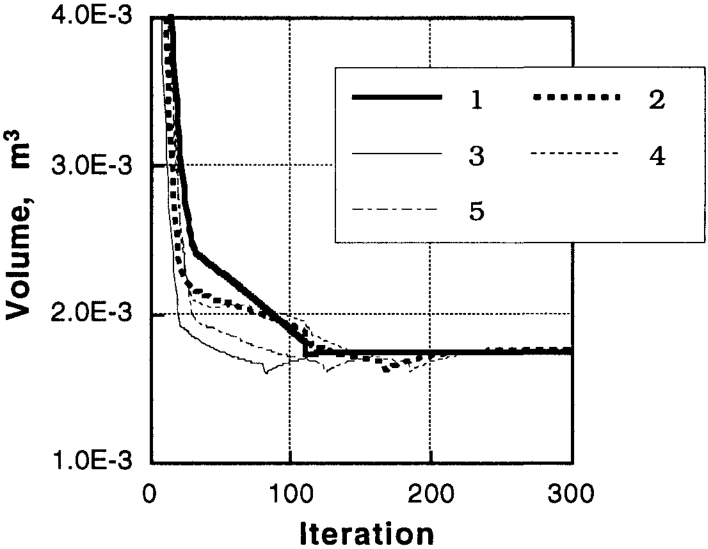


Figure 6: History of total volume of trusses during optimization process.

The similar optimum solutions are obtained from five initial configurations, and they are almost the same as the one shown in Figure 7. It is found from this result that the converged solutions are very similar to the practically optimum solution which is obtained by a genetic algorithm, and the DORAR method is found to be effective. It should be noticed that although there is no lower limit for the design variables in this procedure, very thin members remain even in the converged solutions since the DORAR method containes the resource addition process where a small constant amount of resource is addded to each element in every iteration. This remaining thin members can be deleted if the small amount of resource used in the resource addition process is guradually decreased in the final iteration process.

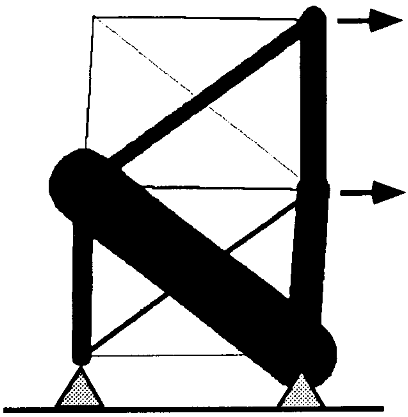


Figure 7: Distributions of optimum sectional areas of the members under loading.

The most important point in the DORAR method is that the method is almost parameter-free, while many nonlinear optimization methods have various parameters controlling their behavior. Only one parameter in the DORAR method is the small constant amount of resource in the resource addition process, but this amount is very easy to determine since this amount directly corresponds to the accuracy of the converged solutions. That is, the amount times the number of elements in the system roughly determins the accuracy of the solutions.

4 Conclusions

A distributed optimization method with some smart behaviors for the minimization of the total resource of a system, the distributed optimization by resource addition and reduction (DORAR) method, is proposed, and a theoretical and experimental investigations are carried out in this paper. The conclusions are as follows:

1) Minimizing the total resource of a system composed of discrete elements which have local resources is distributedly carried out by repeating the resource reduction process and the resource addition process.

2) The separation of the whole constraints into local and global ones makes it possible to evaluate the sensitivities of the local constraints with respect to the local resource in a distributed manner.
3) The proposed distributed optimization method is very robust.
4) The proposed method is highly suitable for parallel processing.

References

1. Schwefel, H.P., *Evolution and Optimum Seeking*, John Wiley & Sons, New York, 1995.
2. Lesser, V.R. and Corkill, D.D., Distributed Problem Solving, *Encyclopedia of Artificial Intelligence,* ed S. C. Shapiro, John Wiley & Sons, New York, 1987, 245-251
3. Miki, M., "Object-Oriented Optimization of Discrete Structures," AIAA Journal, 1995, **33**, 1940-1945.
4. Schnabel R.B., A View of the Limitations, Opportunities, and Challenges in Parallel Nonlinear Optimization, *Parallel Computing*, 1995, **21**, 875-905.
5. Laarhoven, P.J., Parallel Variable Metric Methods for Unconstrained Optimization, *Math. Programming*, 1985, **33**, 68-81.
6. Dennis Jr., J.E. and Torczon, V., Direct Search Method Methods on Parallel Computers, *SIAM J. Optimization*, 1991, **1**, 448-474.
7. Nash, A.G. and Sofer, A., Block Truncated-Newton Methods for Parallel Optimization, *Math. Programming*, 1989, **45**, 529-546.
8. Miki, M. and T. Koita, Parallel Computing for Analysis of Variable Geometry Truss, *AIAA Journal*, 1996, **34**, 1468-1473.
9. Miki, M., Object-Oriented Approach to Modeling and Analysis of Truss Structures, *AIAA Journal.*, 1994, **33**, 48-354.