

Initial Design Domain Reset Method for Genetic Algorithm with Parallel Processing

O-Kaung Lim*, Keum-Shik Hong, Hyuk-Soo Lee, Eun-Ho Choi

School of Mechanical Engineering,

Pusan National University, Kumjung Ku, Busan 609-735, Korea

The Genetic Algorithm (GA), an optimization technique based on the theory of natural selection, has proven to be a relatively robust means of searching for global optimum. It converges to the global optimum point without auxiliary information such as differentiation of function. In the case of a complex problem, the GA involves a large population number and requires a lot of computing time. To improve the process, this research used parallel processing with several personal computers. Parallel process technique is classified into two methods according to subpopulation's size and number. One is the fine-grained method (FGM), and the other is the coarse-grained method (CGM). This study selected the CGM as a parallel process technique because the load is equally divided among several computers. The given design domain should be reduced according to the degree of feasibility, because mechanical system problems have constraints. The reduced domain is used as an initial design domain. It is consistent with the feasible domain and the infeasible domain around feasible domain boundary. This parallel process used the Message Passing Interface library.

Key Words : Genetic Algorithm, Parallel Processing, Message Passing Interface

1. Introduction

It is more effective to use a probabilistic search technique than the general optimization techniques that require differentiation information about objective function and constraint functions when optimum problems cannot be clearly defined, when they have many design variables, or when they have separate or integer design variables. One of the representative methods is the GA (Jin, 2000).

The GA is an algorithm generalized and organized by John Holland. In the GA, a population of individuals is processed generation-by-generation using probabilistic transition rules.

As the GA operates on a 'population' of individuals improving these generation-by-generation, to search and to find a global optimum point is easy. Differentiation of function and additional information is not needed because the GA uses only objective function values (Choi, 2002). But the GA requires so many numbers of analyses that may cause high computational cost for genetic search. Therefore, improvements in the efficiency have been needed to make optimizing process more speedy and effective in discrete optimization using the GA. Nowadays; many efforts have been made to speed up convergence by taking advantage of improved genetic operators (Kogiso et al., 1994; Ahn et al., 2003), or by personal computer network programming based on TCP/IP and client-sever communication model using Windows socket (Park and Song, 2002).

This research using parallel processing with the Message Passing Interface (MPI) library (Park et al., 2002; Lee, 2002) is improving the defect of

* Corresponding Author,

E-mail : oklim@pusan.ac.kr

TEL : +82-51-582-9164; FAX : +82-51-510-2306

School of Mechanical Engineering, Pusan National University, Kumjung Ku, Busan 609-735, Korea.

(Manuscript Received September 23, 2003; Revised March 15, 2004)

high computational cost. This study selected the CGM as a parallel process technique for the load to be equally divided among several computers. In this case, a given domain is divided among several computers, some of the computers can compute only in the infeasible domain. Because of this, the efficiency of parallel processing is diminished. Constrained optimum problems can grasp a feasible domain according to the degree of feasibility due to constraints. To maximize the efficiency of a parallel process, it is proposed to reset the design domain. The reset domain is consistent with the feasible domain and the infeasible domain around the feasible domain boundary. The parallel process is performed with the reset domain as the initial domain.

2. Parallel Genetic Algorithm

The GA has an evolutionary operator and a genetic operator. The evolutionary operator involves selection. The genetic operator is comprised of a crossover and a mutation. The GA searches for an optimum solution according to repetitive processes with three kinds of operators. Relatively fit individuals are combined to produce new individuals. The maximum fitness in the population tends to increase until the process reaches an ultimate maximum fitness. This may or may not be the optimum situation. Although the GA is powerful in operation when applied to complex mechanical system problems, consideration needs to be given to methods to improve the efficiency and speed of operation because the GA requires many individuals. Parallel computation is selected to improve such defects because individuals and the design domain of the GA can be divided. The MPI library for parallel computation is applied to the parallel process.

Message passing is one of the theoretical paradigms that are applied to the distributed function computer system. Recently several forms of message passing techniques have materialized, but the basic communications idea is the same among the processes. In past years, many application programs were developed through the paradigm of message passing. Each hardware devel-

opment company improves the message passing techniques, which are suitable for its parallel computers. The MPI was developed in accordance with the unified development libraries. The MPI is the initial Application Programming Interface (API) for message passing among the processes of distributed function computer systems. The MPI is not a computer language such as FORTRAN, so using a standard compiler; it makes parallel programs that can be compatible with several hardwares. In addition, the MPI has the advantage of reducing the communications load. Because compatible communications are used the amount of data directly written into the memory process can be minimized (MPI On-Line Document).

Parallel process technique is classified into two methods according to subpopulation's size and number. One is the FGM that decreases the number of individuals in the population, and then increases the number of population. The other is the CGM. It divides the population into several subpopulations. The genetic operator is applied independently among the subpopulations. Individuals are exchanged with other individuals in populations during process of evolution (Korea Advanced Institute of Science and Technology, 2000). This study selected the CGM as the parallel process because the load is equally divided among several computers. Each computer is supplied with subpopulations. This method has several parameters that influence the capability of the method. They consist of migration rate and migration interval. The migration rate is the exchange rate of individuals and the migration interval is the exchange period of individuals. This study chose precedence 5% of fitness as the Migration Rate and 10 generations as the Migration Interval.

The given domain is reset in accordance with the degree of feasibility. The reset domain is equally divided into subpopulations. An initial population using a divided domain is created for each subpopulation. Subpopulations perform equal processes, such as the sequential GA. Individuals of a subpopulation are exchanged with each subpopulation in accordance with

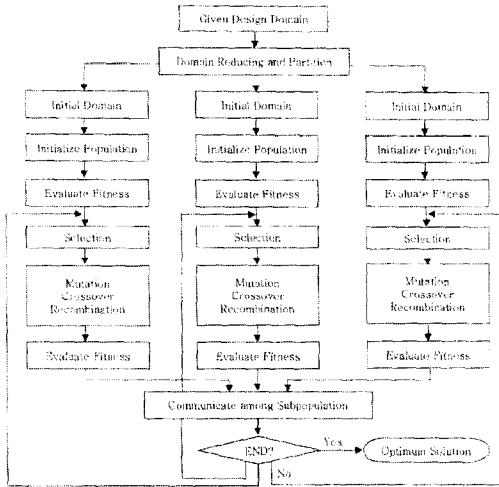


Fig. 1 The flowchart of Parallel genetic algorithm

the chosen Migration Rate and Migration Interval. Figure. 1 shows a flowchart of the parallel GA.

3. Reset of Initial Domain

A given design domain is divided among several computers when the parallel process is applied to the constrained optimum problem. It can generate a node that is computed only in an infeasible domain. This decreases the efficiency of the parallel process. Using the reset of initial domain can improve this problem. A constrained optimum problem can grasp the feasible domain in accordance with the degree of feasibility due to the constraints. The degree of feasibility is determined by whether the constraints are satisfied or not. The GA generates an initial population that consists of individuals in a given domain. Each individual is put in the feasible or infeasible according to the degree of feasibility. The initial domain is divided by a specific gap. This gap, the sampling gap (Sg), is applied to reset a given design domain. Using the Sg , the boundary between the feasible and infeasible domain can be searched. Constrained optimum problems have two cases of feasible domain. One is bounded feasible domain; the other is unbounded feasible domain as truss problems that take unlimited upper-values.

When feasible domain is bounded, initial individuals are generated by random-generation. The Sg is formulated as follows.

$$Sg = (X_U - X_L) / \sqrt{I} \quad (1)$$

In Eq. (1), X_U and X_L represent the upper and lower limits of a given design domain. V is the number of design variables, and I is the number of individuals.

The design domain is reset by the offset value that is based on the feasible individual ratio. The offset value has a value between 1 and 2 according to the feasible individual ratio. The offset value V_o is formulated as follows.

$$V_o = \left(1.2 + \frac{R}{2} \right) \quad (2)$$

In Eq. (2), R is the feasible individual ratio in total individuals. 1.2 is the minimum offset value to involve feasible domain.

In the case that the ratio of feasible individuals is large, the Sg is narrow for the feasible domain. This case includes the infeasible domain around the feasible domain boundary by increasing offset value because outer feasible domain cannot be included. In the case that this ratio is small, the inclusion of the infeasible domain is minimized with a decreasing offset value because the Sg is wide compared with the feasible domain. The upper and lower limit for resetting a domain is formulated as follows.

$$X_{RU} = X_{FU} + V_o \times Sg \quad (3)$$

$$X_{RL} = X_{FL} - V_o \times Sg \quad (4)$$

In Eqs. (3) and (4), X_{RU} and X_{RL} represent the upper and lower limits of the reset domain. X_{FU} and X_{FL} represent the upper and lower limit of feasible individuals. If the ratio of feasible individuals is not between 5% and 90%, when a given domain is reset, it will be modified for the accuracy of the solution and the efficiency of the process.

When feasible domain is unbounded, initial individuals are generated by regular-interval. A given design domain is searched from lower-limit to upper-limit. The upper and lower limit for resetting a domain is formulated as follows.

$$X_{RU} = X_H \tag{5}$$

$$X_{RL} = X_F - V_0 \times S_g \tag{6}$$

In Eqs. (5) and (6), X_F represents the first interval that takes feasible individuals. X_H represents the first interval that takes 100% feasible individuals.

4. Parallel Example

Linear and nonlinear constraint problems are selected as parallel examples. The linear constraint problems are diagrammed by the extreme point. The extreme point of the feasible domain can be found by the Diagram and Simplex method. This study compared the sequential GA by one computer with parallel GA by two computers, using 100 individuals and 500 generations. Example 1 and Example 2 are bounded feasible domain problems. Example 3 is unbounded feasible domain problem.

4.1 Linear constraint problem

4.1.1 Example 1

This example 1 is bounded feasible domain problem. The optimum problem is formulated as follows (Arora, 1994).

$$\text{Maximize } Z(x) = 3x_1 + 2x_2 \tag{7}$$

$$\text{Subject to } G(1) = -x_1 + x_2 \leq 0 \tag{8}$$

$$G(2) = -x_1 - x_2 + 2 \leq 0 \tag{9}$$

$$G(3) = 2x_1 + x_2 - 6 \leq 0 \tag{10}$$

$$x_i \geq 0 \tag{11}$$

The given design domain and feasible design domain are shown in Fig. 2.

The given design domain and the domain with extreme point are shown in Fig. 3, Fig. 4 and Table 1.

Individuals are classified by a descending series. The upper and lower limits of feasible individuals are offset according to the Eqs. (3) and (4). The upper-lower limits of feasible individuals and reset domain are shown in Fig. 5, Fig. 6 and Table 2.

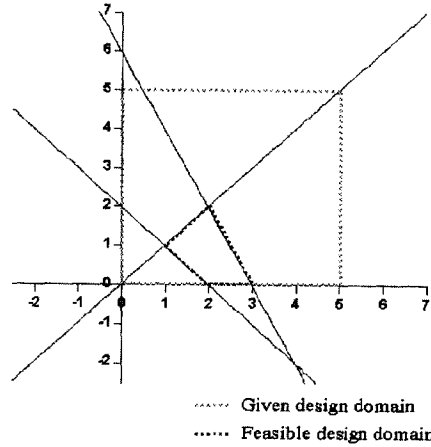


Fig. 2 Given design domain and feasible design domain

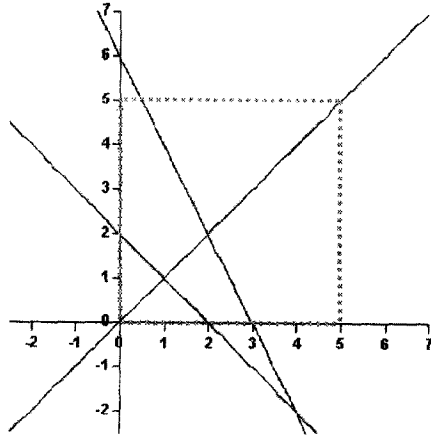


Fig. 3 Given design domain

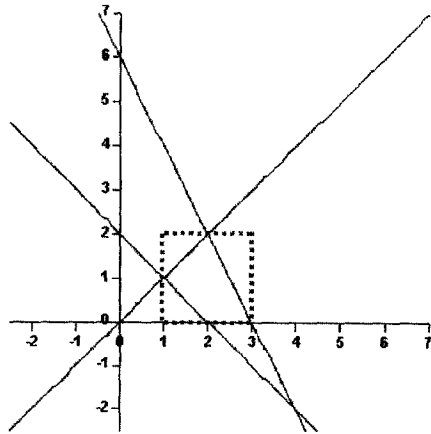


Fig. 4 Domain with extreme point

A sequential process is performed with a given design domain as the initial domain (Fig. 7), and a parallel process is performed with the resetting domain as the initial domain (Fig. 8). The results and computing time are compared with the sequential process. This is shown in Table 3. The difference in the results is small, but the parallel computing time is faster than the ideal parallel

Table 1 Given design domain and domain with extreme point (Example 1)

Given design domain	Extreme domain
$0 \leq x_1 \leq 5$	$1 \leq x_1 \leq 3$
$0 \leq x_2 \leq 5$	$0 \leq x_2 \leq 2$

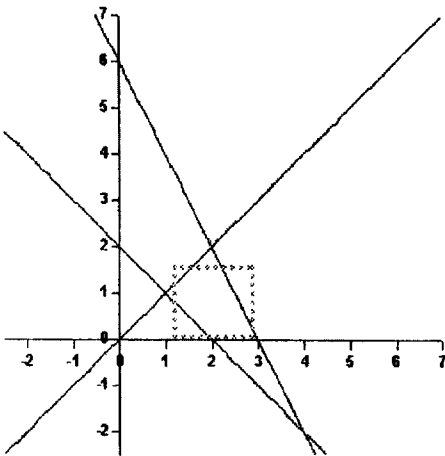


Fig. 5 Upper-lower of feasible individuals

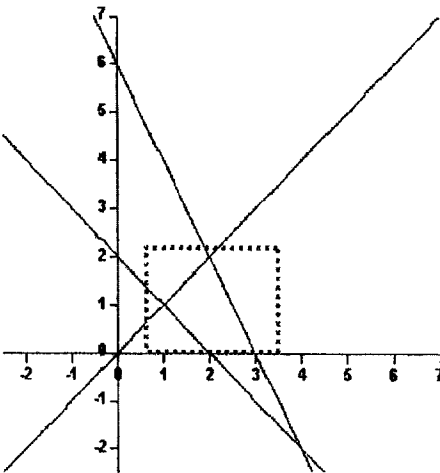


Fig. 6 Reset domain

Table 2 Upper-lower and reset domain (Example 1)

Upper and lower	Reset domain
$1.25 \leq x_1 \leq 2.78$	$0.65 \leq x_1 \leq 3.38$
$0.02 \leq x_2 \leq 1.62$	$0 \leq x_2 \leq 2.22$

Table 3 Optimum value and elapsed time (Example 1)

	Sequence GA	Parallel GA
Optimum value	$x_1=2.09$	$x_1=2.00$
	$x_2=1.82$	$x_2=2.00$
	$F=9.90$	$F=10.00$
Elapsed time (s)	7.73	2.78

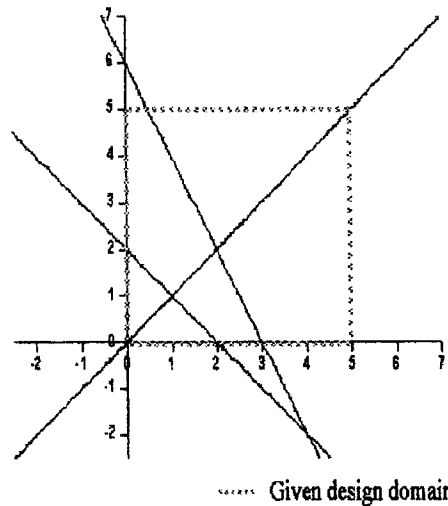


Fig. 7 Initial design domain of sequential process

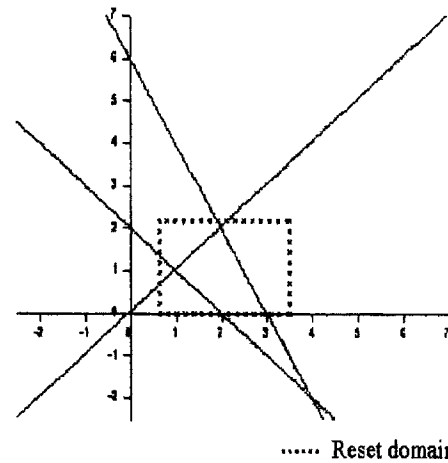


Fig. 8 Initial design domain of parallel process

computing time, which is half the sequential computing time. Using of a descending series causes additional time reduction. A descending series is applied to individual classification and fitness evaluation. In the case of a descending series, the computing time varies in a geometric progression as a classified number.

4.1.2 Example 2

This example 2 is bounded feasible domain problem. The optimum problem is formulated as follows (Hock and Schittkowski, 1981).

Maximize $Z(x) = -x_1x_2x_3$ (12)

Subject to $G(1) = x_1 + 2x_2 + 2x_3 \leq 72$ (13)

$G(2) = x_1 \leq 20$ (14)

$G(3) = x_2 \leq 11$ (15)

$G(4) = x_3 \leq 42$ (16)

$x_i \geq 0 \quad i=1, 2, 3$ (17)

The feasible design domain is shown in Fig. 9

The given design domain and the domain with extreme point are shown in Fig. 10, Fig. 11 and Table 4.

The individuals are classified by a descending series. The upper and lower limits of feasible individuals are offset according to the Eqs. (3) and (4). The upper-lower of feasible individuals and reset domain are shown in Fig. 12, Fig. 13 and Table 5.

A sequential process is performed with a given design domain as the initial domain (Fig. 14), and a parallel process is performed with the reset

Table 4 Given design domain and domain with extreme point (Example 2)

Given Design Domain	Extreme domain
$0 \leq x_1 \leq 50$	$0 \leq x_2 \leq 20$
$0 \leq x_2 \leq 50$	$0 \leq x_2 \leq 11$
$0 \leq x_3 \leq 50$	$0 \leq x_3 \leq 36$

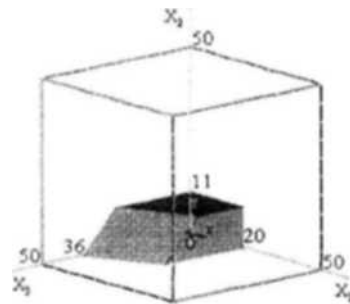
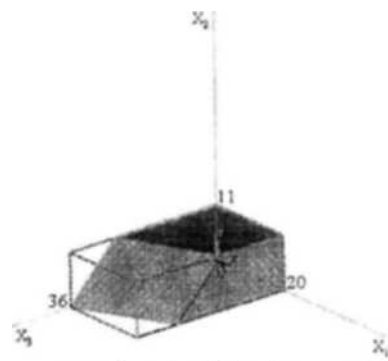


Fig. 10 Given design domain



..... Domain with extreme point

Fig. 11 Domain with extreme point

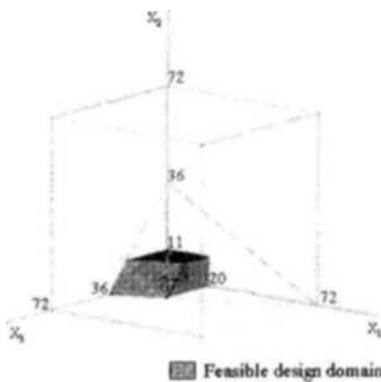
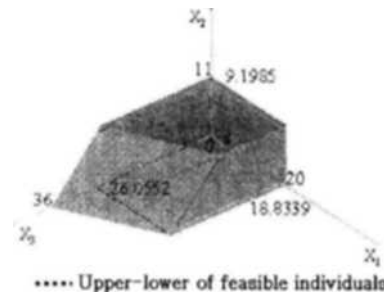


Fig. 9 Feasible design domain



..... Upper-lower of feasible individuals

Fig. 12 Upper-lower of Feasible Individuals

domain as the initial domain (Fig. 15). The results and computing time are compared with the sequential process. This is shown in Table 6. The results and computing time of this problem show the same tendency as in Example 1.

Table 5 Upper-lower and reset domain (Example 2)

Upper and lower	Reset Domain
$0.87 \leq x_1 \leq 18.83$	$0 \leq x_1 \leq 29.61$
$0.52 \leq x_2 \leq 9.20$	$0 \leq x_2 \leq 19.97$
$0.77 \leq x_3 \leq 26.06$	$0 \leq x_3 \leq 36.83$

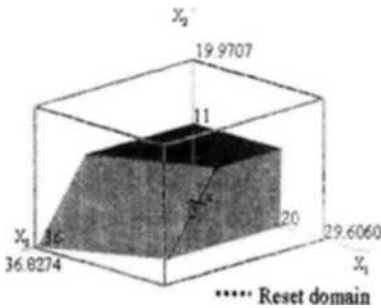


Fig. 13 Reset Domain

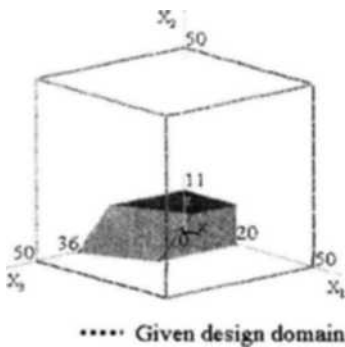


Fig. 14 Initial design domain of sequential process

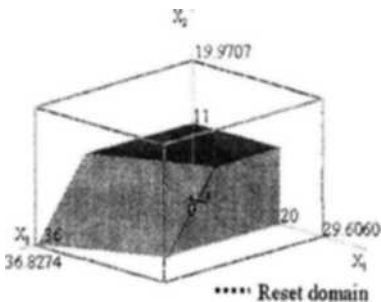


Fig. 15 Initial design domain of parallel process

Table 6 Optimum value and elapsed time (Example 2)

	Sequence GA	Parallel GA
Optimum value	$x_1=19.75$	$x_1=19.75$
	$x_2=10.90$	$x_2=10.89$
	$x_3=14.98$	$x_3=14.89$
	$F=-3225.65$	$F=-3203.43$
Elapsed time (s)	12.55	4.22

4.2 Nonlinear Constraint Problem

4.2.1 Example 3

This example 3 is unbounded feasible domain problem. The three-bar truss is shown in Fig. 16. The objective is to minimize the weight of the structure by varying the cross-sectional areas of the elements. Thus the fittest structure will be the one with the least weight. We will constrain the problem by limiting the stress in the elements and the buckling constraint. Elements 1 and 3 will have the same cross-sectional areas, $A_1=A_3=X_1$, and element 2 will have the cross-sectional area, $A_2=X_2$. The design data for the three-bar truss is shown in Table 7 (Choi, 2002).

Example 3 can be formulated as follows.

$$\text{Minimize } F(x) = \gamma \sum_{i=1}^3 x_i l_i \tag{18}$$

$$\text{Subject to } G_i(x) = \frac{|\sigma_i|}{\sigma_a} - 1 \leq 0, (i=1, 2, 3) \tag{19}$$

$$G_j(x) = |\sigma_j| - \frac{\pi E x_j}{8 l_j^2} \leq 0, j=3(20)$$

This problem is unbounded feasible domain

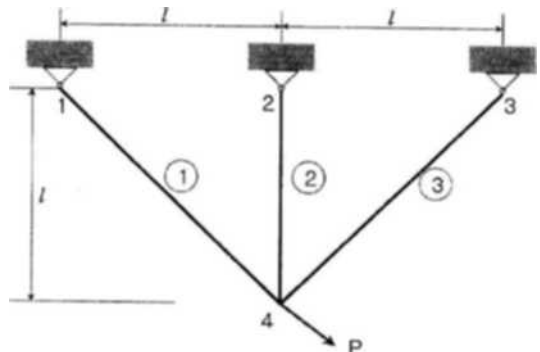


Fig. 16 Symmetric three-bar truss

Table 7 Design data for three-bar truss (Example 3)

Design data	Value
Weight density, γ (N/mm ³)	27.14×10^{-6}
Length, l (mm)	2540
Modulus of elasticity, E (kN/mm ²)	68.95
Load, P (kN)	88.96
Allowable tensile stress, σ_a (kN/mm ²)	0.1379
Allowable compressive stress, σ_a (kN/mm ²)	0.1034
Initial crossover probability	0.5
Initial mutation probability	0.025
Max. generation number	500

Table 8 Given design domain and reset domain (Example 3)

Given design domain	Reset domain
$10^{-3} \leq x_1 \leq 10^4$	$10^{-3} \leq x_1 \leq 10^3$
$10^{-3} \leq x_2 \leq 10^4$	$10^{-3} \leq x_2 \leq 10^3$

Table 9 Optimum value and Elapsed time (Example 3)

	Sequence GA	Parallel GA
Result Value	$x_1=680.91$	$x_1=672.34$
	$x_2=5.99$	$x_2=6.07$
	$F=133.18$	$F=131.51$
Elapsed time (s)	11.39	5.60

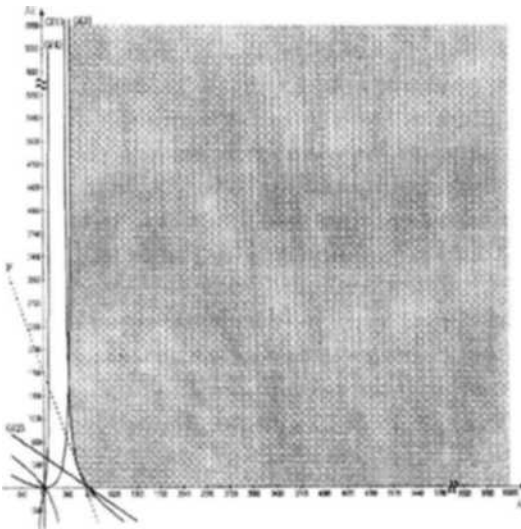


Fig. 17 Given design domain

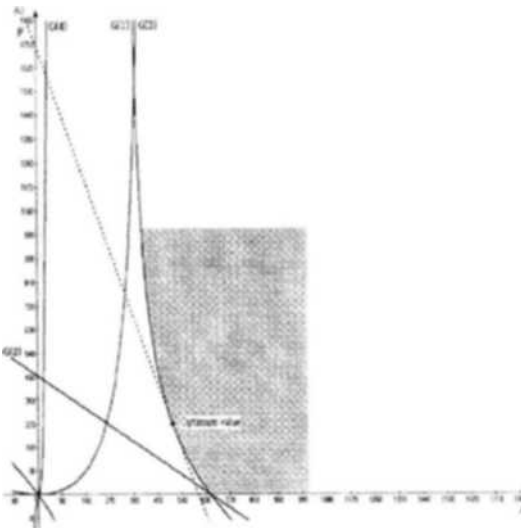


Fig. 18 Reset domain

problem. A given design domain is offset according to the Eqs. (5) and (6). The Given design domain and reset domain are shown in Table 8.

A sequential process is performed with a given design domain as the initial domain (Fig. 17), and a parallel process is performed with the reset domain as the initial domain (Fig. 15). Optimum value exists around constraint function boundary. This is shown in Fig. 18. Accordingly, unbounded feasible domain problem, when parallel processing is applied, can be reduced with Eq. (6). The results and computing time are compared with the sequential process. This is shown in Table 9. The results and computing time of this problem show the same tendency as in Example 1.

4.3 Comparison of examples

The results of two linear constraint problems (Ex. 1, Ex. 2) with parallel processes are compared with the correct solution. The tolerance is not large. Table 10 shows the comparison of result values.

The optimum values of a nonlinear constraint problem (Ex. 3) are applied as the area of each element. Using FEM codes solves the stress problem of each element. The stress values are not more than an Allowable Stress. Therefore the results are satisfied. The examination of the optimum values is shown in Table 11.

Generally, when the parallel process is performed, the computing time is proportionally reduced as increase, the number of nodes, but

Table 10 Optimum value comparison of two linear constraint problems (Example 1, 2)

Problem	Correct solution	Optimum value
Example 1	$x_1=2$	$x_1=2.00$
	$x_2=2$	$x_2=2.00$
	$F=10$	$F=10.00$
Example 2	$x_1=20$	$x_1=19.75$
	$x_2=11$	$x_2=10.90$
	$x_3=15$	$x_3=14.98$
	$F=-3300$	$F=-3225.65$

Table 11 Examination of optimum values (Example 3)

Example 3	Area (mm ²)	Stress (N/mm ²)	Allowable stress (N/mm ²)
Sequence GA	$A_1=680.91$	$\sigma_1=129.84$	137.9 (tensile) 103.4 (compressive)
	$A_2=5.99$	$\sigma_2=129.04$	
	$A_3=680.91$	$\sigma_3=-8.02E-01$	
Parallel GA	$A_1=672.34$	$\sigma_1=131.47$	137.9 (tensile) 103.4 (compressive)
	$A_2=6.07$	$\sigma_2=130.64$	
	$A_3=672.34$	$\sigma_3=-8.33E-01$	

the result of this study can confirm that the computing time is rapidly reduced. This is caused by the use of a descending series. It is applied to the individual classification and fitness evaluation. When a descending series is used, the computing time varies in a geometric progression as a classified number.

5. Conclusions

In this paper, we proposed parallel processing with the MPI library to improve the high computational cost for GA. Moreover we proposed reset the design domain to maximize the efficiency of a parallel process. The proposed parallel process technique was applied to linear and non-linear problems. The parallel process was compared with the sequential process. We get conclusion as follows,

(1) The parallel process with two computers was averagely faster 2.59 times than the sequential process with a computer in Examples 1, 2, 3.

(2) Using of a descending series causes additional time reduction. A descending series is applied to individual classification and fitness

evaluation. In the case of a descending series, the computing time varies in a geometric progression as a classified number.

(3) A given design domain is divided among several computers when the parallel process is applied to the constrained optimum problem. It can generate a node that is computed only in an infeasible domain. This decreases the efficiency of the parallel process. This problem was improved in using reset of initial design domain.

(4) Unbounded feasible domain as truss problems that take unlimited upper-values can be reduced because the optimum value exists in the boundary of constraint conditions.

Acknowledgment

This research was supported by the Center for Innovative Design Optimization Technology, and the Korea Science and Engineering Foundation.

References

- Gang-Gyoo Jin, 2000, *Genetic Algorithms and Their Applications*, Kyowoo publishing, pp. 55~69.
- Hyung-Wook Park, Min-Soo Kim and Dong-Hoon Choi, 2002, "A New Decomposition Method for Parallel Processing Multi-Level Optimization," *KSME International Journal*, Vol. 16, No. 5, pp. 609~618.
- Jasbir s. Arora, 1994, *Introduction to Optimum Design*, McGraw-Hill Book Co., p. 279.
- Jung-Muk Choi, 2002, "A Study on Improvement of Genetic Algorithm Operation Using the Restarting Strategy," *Journal of the Computational Strutural Engineering Institute of Korea*, 6 Vol. 15, No. 2, pp. 305~313.
- Jung-Sun Park, Seok-Bong Song, 2002, "Optimization of a Composite Laminated Structure by Network-Based Genetic Algorithm," *KSME International Journal*, Vol. 16 No. 8, pp. 1038~1038.
- Kogiso, N., Watson, L. T., Gurdal, Z. and Haftka, R. T., 1994, "Genetic Algorithm with Local Improvement Design," *Structural Optimization*, Vol. 7, pp. 207~218.

Korea Advanced Institute of Science and Technology, 2000, "Development of Hi-Speed Parallel Evolutionary Computation S/W and Its Application Techniques," Report, Ministry of Science & Technology.

MPI On-Line Document, <http://www-unix.mcs.anl.gov/mpi/index.html>.

Sang-Ho Lee, 2002, "Introduction to Parallel Computer and Parallel Programming Technique," *Journal of the KSME*, Vol. 42, No. 1, pp. 36~42.

Willi Hock, Klaus Schittkowski, 1981, *Lecture Note in Economics and Mathematical Systems ; 187 Test Examples for Nonlinear Programming Codes*, Springer-Verlag Berlin Heidelberg, p. 59.

Young-Kong Ahn, Young-Chan Kim and Bo-Suk Yang, 2003, "Optimal Design of a Squeeze Film Damper Using an Enhanced Genetic Algorithm," *KSME International Journal*, Vol. 17, No. 12, pp. 1938~1948.