

Comparison of Pulling Back and Penalty Methods for Constraints in DPMBGA

Hisashi Shimosaka

Graduate School of Engineering,
Doshisha University
1-3 Tatara Miyakodani
Kyotanabe, Kyoto 610-0394
hisashi@mikilab.doshisha.ac.jp

Tomoyuki Hiroyasu

Department of Engineering,
Doshisha University
1-3 Tatara Miyakodani
Kyotanabe, Kyoto 610-0394
tomo@is.doshisha.ac.jp

Mitsunori Miki

Department of Engineering,
Doshisha University
1-3 Tatara Miyakodani
Kyotanabe, Kyoto 610-0394
mmiki@mail.doshisha.ac.jp

Abstract-

To solve real-world problems by Genetic Algorithms (GAs), GAs that have a strong searching capability are needed. In this paper, Distributed Probabilistic Model-Building Genetic Algorithm (DPMBGA) is applied to solve the problems. The DPMBGA is an extended algorithm of Probabilistic Model-Building GA (PMBGA) and it also has a strong searching capability. In real-world problems, constraints often exist. As such, mechanisms that can treat the constraints should be added to the GAs. Two mechanisms for treating constraints are the penalty method and pulling back method.

The DPMBGA with the penalty method and pulling back method is applied to truss structural optimization problems. Through a simulation, the searching capability and efficiency of the pulling back method and penalty method are discussed. From the discussion, it is concluded that the pulling back method can derive the solutions even if the problem is difficult. Compared to the penalty method, the number of the individuals that violate the constraints is smaller in the pulling back method.

1 Introduction

The Genetic Algorithm (GA) is one of the optimization methods that simulate creatures' evolution and heredity [Goldberg89, Holland75]. To perform an effective search in the GAs, it is important that the children inherit the good information of the parents with good fitness values. To perform this operation, Probabilistic Model-Building Genetic Algorithm (PMBGA) [Pelikan99] and Estimation of Distribution Algorithm (EDA) [Larranaga01] were introduced. In the PMBGA, children are generated from the probabilistic model that is developed using the statistic information of the good parents. We also introduced Distributed PMBGA (DPMBGA) [Tomoyuki03]. The DPMBGA is an extended algorithm of the PMBGA. In the DPMBGA, the population is divided into sub-populations using the concept of the Distributed GA (DGA) [Tanese89]. In each sub-population, the operations of the PMBGA are performed. The probabilis-

tic model is developed using Principal Component Analysis (PCA) [Jolliffe86] and normal distributions. The children are generated from the probabilistic model. The precise explanation of these operations is described in the following section. Because of the distributed population mechanism, the diversity of the solutions during the search can be maintained.

In many real-world problems such as structural optimization problems, constraints can often be found. Therefore, the optimization method that treats these problems should have a mechanism for handling the constraints. For example, the GA is an optimization method that cannot treat constraints explicitly. Thus, when the GA is applied to solve optimization problems with constraints, some mechanisms to treat the constraints have to be prepared. In previous studies, some mechanisms for the GA that can treat the constraints were introduced [Coello99].

The typical and easiest way for treating constraints in the GA is a penalty method [Goldberg89, Richardson89]. In the penalty method, the fitness value is modified to add a certain penalty value when the searching point is out of the feasible region. This operation is very simple and easy to implement. However, it is very difficult to set the appropriate penalty. At the same time, when the feasible region is very narrow compared to the design field, most of the generated individuals are out of feasible region and most of the fitness values are bad. In such a case, an effective search cannot be performed. To perform an effective search in problems with constraints, other mechanisms that can treat constraints are necessary.

In this paper, besides the penalty method, the pulling back method [Mimura02] is discussed. In the pulling back method, the search point that is out of the feasible region is pulled back to the nearest point that satisfied the constraints. In this method, since all the searching points will eventually satisfy the constraints, an effective search can be expected.

In this paper, the truss structure is designed by the DPMBGA. Designing the truss structure problem has several types of constraints. In these problems, the searching ability of the penalty method and the pulling back method are compared.

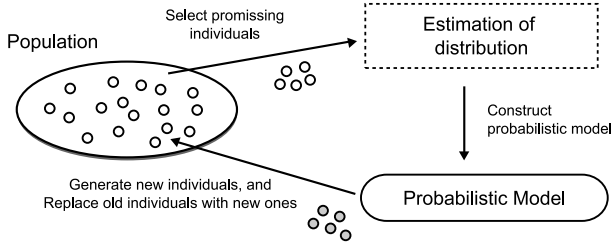


Figure 1: Outline of PMBGA

2 Distributed Probabilistic Model-Building Genetic Algorithm

In this paper, DPMBGA is applied for solving problems. The DPMBGA is an extended algorithm of the PMBGA using the Distributed GA (DGA) mechanism. In this section, the outline of the general PMBGA and the DGA are described. After that, the DPMGA is illustrated.

2.1 Probabilistic Model-Building Genetic Algorithm

In GAs, there are several genetic operations, such as selection, crossover, mutation and evaluation. Among these operations, the crossover and the mutation can derive new searching points. In the crossover and mutation operations, it is important to inherit the characteristic of the good individuals. An approach to inheriting good characteristics is a new type of GA called PMBGA [Pelikan99] or EDA [Larranaga01]. The PMBGA or EDA focuses on using the statistical information of the good individuals. The outline of the basic procedures of the PMBGA is summarized in Figure 1.

In the PMBGA, the following steps are performed repeatedly until termination criteria are met. Firstly, the good individuals are selected from the population. Secondly, a probabilistic model is estimated by the distribution of the selected individuals. Finally, new individuals are generated from the probabilistic model and some individuals in the population are replaced with the new individuals. In conclusion, it is said that the operations of the PMBGA are substituted for the crossover and mutation operation.

Several types of PMBGAs and EDAs are proposed. Pelikan summarized these methods from several point of views [Pelikan99]. From the viewpoint of coding, they are classified into bit-string methods and real-vector methods. From the viewpoint of consideration of correlation among design variables, there are methods that do not take care of correlations, methods that take care of correlation between two design variables, and methods that take care of correlation among more than three design variables.

In this paper, DPMBGA is applied to solve problems. This DPMBGA is a method where real vectors are used and

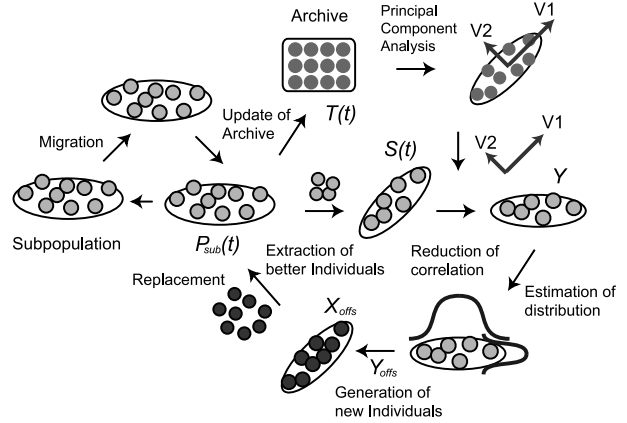


Figure 2: Outline of DPMBGA

correlation among more than three design variables is considered.

2.2 Distributed Genetic Algorithm

GA is one of the multi-point searching methods that need many iterations. As a result, the GA requires a high calculation cost. One of the solutions of this is to perform the GA operations on parallel. There are many previous studies that are concerned with parallel GAs [Paz98].

One of the parallel models of GA is DGA [Tanese89]. The DGA is often called the "island model". In this model, the total population is divided into sub-populations and genetic operations are performed in each island. Additionally, some individuals in each island are chosen and moved to other islands at regular intervals. This operation is called a "migration". The ratio of the migrated individuals is called the "migration ratio" and the interval between the migrations is called the "migration interval". Compared to the conventional GA, the DGA is good at maintaining the diversity of the solutions. Therefore, the DGA can find the optimum with a smaller calculation cost than the conventional GA [Tanese89]. In this paper, the concept of the DGA is used for maintaining diversity.

2.3 Outline of the Distributed Probabilistic Model-Building Genetic Algorithm

In this paper, DPMBGA [Tomoyuki03] is utilized. The DPMBGA is an extended model of the PMBGA and it has the DGA scheme as the mechanism to maintain the diversity of the individuals. There are several sub-populations. The migration operation is also performed at regular intervals.

In the DPMBGA, in order to generate new searching points in consideration of the correlation among the design variables, real-coded GAs and Principal Component Analysis (PCA) [Jolliffe86] are used for transforming the distribu-

tion of the solutions. The outline of the DPMBGA is shown in Figure 2. In each island of the DPMBGA, the following procedures are performed.

1. Some individuals who have the good fitness values are sampled.
2. The sampled individuals are transferred by the PCA into new space to reduce the correlation among the design variables.
3. A probabilistic model is estimated by the normal distributions.
4. New individuals are generated from the probabilistic model in the new space.
5. The new individuals are transferred into the original space and substituted for the old individuals.
6. The mutation is performed on the new individuals.
7. The new individuals are evaluated.
8. The archive is updated.

In the following sections, the detail of the algorithm of the DPMBGA is described.

2.4 Reduction of the correlation among the design variables using Principal Component Analysis

At first, in the DPMBGA, the sampling individuals $S(t)$ that have the good evaluation value are extracted from $P_{sub}(t)$ in each island. The number of these individuals is determined with sampling rate R_s . $S(t)$ exists in each island. These sampling individuals are chosen in order of the better evaluation value. However, the individual who has the same design variables is not chosen repeatedly. New individuals are generated from the statistical information of these individuals.

$S(t)$ is transferred by the PCA operation. However, the PCA is determined using the information of the individual set of $T(t)$. $T(t)$ is different from $S(t)$. $T(t)$ consists of the individuals who are the best in the past search. $T(t)$ is called the "archive". The archive is formed in the following way. The archive is always updated when an individual is evaluated. The individuals are stored in the archive in order of the better evaluation value. When the size of the archive is exceeded, the worst individuals in the archive are eliminated one by one. With this operation, the arbitrary number of the individuals can be used for the information of the PCA. The archive also exists in each island.

The average of $T(t)$ is subtracted from $T(t)$ and $T(t)$ becomes matrix \mathbf{T} (N_T column \times D line). N_T is the size of the archive and D is the number of the design variables. The average of $S(t)$ is also subtracted from $S(t)$ and $S(t)$ becomes \mathbf{X} (N_S column \times D line). N_S is the number of the sampling individuals $S(t)$.

Then, the covariance matrix \mathbf{S} of \mathbf{T} is derived and the

eigen values and vectors of \mathbf{S} are obtained. \mathbf{S} is a real symmetric matrix and derived as follows.

$$\mathbf{S} = \frac{1}{N_T - 1} \mathbf{T}^T \mathbf{T} \quad (1)$$

The eigen vector indicates the axis of the new space.

Using the derived eigen vectors, the design variables \mathbf{X} of the solution set $S(t)$ are transferred. After the transfer into the new space, there is no correlation among the design variables. The coordinate transfer matrix consists of the vectors $\mathbf{V} = [v_1, v_2, \dots, v_D]$. After multiplying \mathbf{V} , the vector \mathbf{X} becomes \mathbf{Y} (N_S column \times D line). The coordinate of \mathbf{Y} corresponds to the eigen vectors.

2.5 Generation of New Individuals

New individuals are generated using the normal distribution of the information of \mathbf{Y} . Each value of the design variable in a new individual is also independently determined one by one. Therefore, when there are n design variables in an individual, there should be n different normal distributions. The normal distribution is formed as follows: the average is the same as the average value of the target design value of \mathbf{Y} . The variance is derived by multiplying the variance value of the target design variable of \mathbf{Y} by the parameter Amp . The values of the design variables are determined randomly, but the total distribution of the new individuals should be the same as the formed normal distribution. The number of the newly created individuals is the same as the number of the individuals in an island (N_P) and the generated individuals are stored in \mathbf{Y}_{offs} (N_P column \times D Line). The derived \mathbf{Y}_{offs} is then substituted into the original space as the follows.

$$\mathbf{X}_{offs} = \mathbf{Y}_{offs} \cdot \mathbf{V}^{-1} \quad (2)$$

The average of $S(t)$ is added from \mathbf{X}_{offs} and \mathbf{X}_{offs} becomes the new individuals. These new individuals are absolutely substituted for all of the old individuals $P_{sub}(t)$ and those become $P_{sub}(t+1)$.

2.6 Distributed Environment Scheme

In the DPMBGA, the new individuals are generated using the PCA. By this operation, the information of the correlation among the design variables that is found during the GA search is reflected to the new individuals. In problems where there is a correlation among the design variables, reducing the correlation is expected to allow efficient searching. On the other hand, in problems where there is not a correlation among the design variables, the PCA may not function effectively. Therefore, Distributed Environment GA (DEGA) is applied to the DPMBGA.

DEGA is one of the distributed GA schemes [Miki99]. In the DEGA, the different parameters or the different operations are performed in each island. It is well known that the

search capability of GA depends on the value of the parameters. The optimum values of these parameters also depend on the targeted problems. Therefore, preliminary experiments are necessary in order to derive the optimum values of the parameters. In the DEGA, the values of the parameters and the operations are different in each island. These parameter values are not the best but can derive adequate solutions.

In this paper, the DEGA scheme is applied to the DPM-BGA. The PCA is performed in one half of the islands but not in the other half of the islands. In the islands where the PCA is not performed, the crossover among the design variables between a migrated individual and an individual in the island is also performed on migration. This mechanism maintains the diversity of the population. The application of the DEGA scheme is expected to permit efficient searching regardless of the correlation among the design variables.

3 Methods for Dealing with Constraints

In this paper, GAs are applied to optimization problems with constraints such as structural optimization problems. In order to solve the optimization problems with constraints, some mechanisms of treating the constraints should be added to the GAs, since the GAs cannot treat constraints explicitly. In this study, the penalty method and the pulling back method are applied to the DPMBGA. The penalty method and pulling back methods are explained briefly.

3.1 The Penalty method

The penalty method is a method that optimizes the modified objective function that is added to the penalty for each violated constraint [Goldberg89, Richardson89]. Suppose there is an optimization problem with the constraints like equation (3) below.

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{such that} \quad & g_j(\mathbf{x}) \geq 0 \quad j = 1..m \end{aligned} \quad (3)$$

In this time, the penalty method defines a function like equation (4) below. ρ is more than 0 and is called the "penalty parameter".

$$\min F_\rho(\mathbf{x}) = f(\mathbf{x}) + \rho(\sum_{i=1}^m \max\{0, -g_j(\mathbf{x})\}) \quad (4)$$

It is very easy to implement the penalty method. In the penalty method, proper penalty values are essential for effective searching. This is a disadvantage point for users. At the same time when the feasible region is very narrow compared to the design field, most of the initial individuals are out of the feasible region. Then, penalty values are added to derive the fitness function and may prevent an effective search.

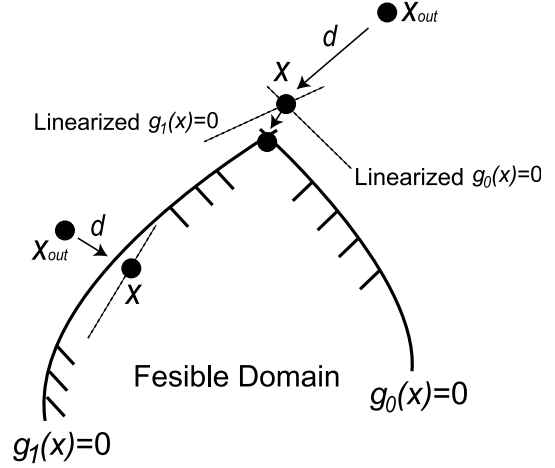


Figure 3: Outline of Pulling Back Method

3.2 The Pulling Back method

In the pulling back method, when an individual violates the constraints during a GA search, the individual is pulled back to the nearest point in the feasible region [Mimura02]. Figure 3 shows the outline of the pulling back method. Suppose there is an optimization problem with constraints like equation (3). If an individual (\mathbf{x}_{out}) violates one or more constraints, equation (5) should be solved in the pulling back method. \mathbf{d} shows $\mathbf{x} - \mathbf{x}_{out}$.

$$\begin{aligned} \min \quad & F_{pb}(\mathbf{d}) = \sqrt{\mathbf{d} * \mathbf{d}} \\ \text{such that} \quad & \nabla g_j(\mathbf{x}_{out})\mathbf{d} + g_j(\mathbf{x}_{out}) \geq 0 \\ & , \quad j = 1, \dots, h \end{aligned} \quad (5)$$

The objective function is formulated as the minimization of the distance between a point in the feasible region and the individual (\mathbf{x}_{out}) who violates the constraints. In the equation (5), the violated constraints are linearized. By solving this equation (5), a new searching point \mathbf{x} in the feasible region is obtained from \mathbf{d} .

The equation (5) is a quadratic programming problem and it can be solved by several optimization techniques. In this paper, an active constraint method is used. However, because the pulling back method linearizes the constraints, the individual may not satisfy all the constraints after just one pulling back operation. Therefore, the pulling back will continue until all the constraints are satisfied. The terminal criteria of the pulling back are listed below.

1. The individual that violates the constraints before an operation satisfies all of the constraints.
2. The number of times that the pulling back is performed exceeds a certain number.
3. The distance after the operation is smaller than the preset distance.

In this paper, the maximum time of the pulling back is set 20 times per individual and the minimum distance in one pulling back is set $1e-8$. If the individual does not satisfy all of the constraints after the pulling back operations, the fitness of the new searching point x is obtained by the penalty method.

4 Numerical Example

In this section, the searching capability and effectiveness of the penalty method and the pulling back methods are discussed through truss structural optimization problems. The truss structural optimization problem is one of typical optimization problems with several type of constraints.

4.1 Truss Structural Optimization Problem

In this paper, a truss structure that consists of nodes and members is designed by the DPMBGA with the penalty method and the pulling back method. The truss structural optimization problem is one of the typical structural and nonlinearly constrained optimization problems.

In the problems, the volume of structure is minimized within certain values of stress and displacement and buckling conditions. The design variables are the areas of each member. Therefore, the number of the design variables is same as the number of the members of the target truss structure.

There are two types of designed truss structures. One of them is the 2-stage truss structure that consists of 6 nodes and 10 members (Figure 4 : left). The other is the 3-stage truss structure that consists of 8 nodes and 15 members (Figure 4 : right). In these structures, the two nodes at the ground are fixed and several nodes are loaded as $5000N$. All the members are linear elasticity and young's module is $10GPa$. The domain of the design variables is $1.0e - 15(m^2)$ to $1e - 3(m^2)$. In the 2-stage truss structure, it is assumed that the structure whose volume is less than $5.9e - 4(m^3)$ is optimum. Similarly, in the 3-stage truss structure, it is assumed that the structure whose volume is less than $1.27e - 3(m^3)$ is optimum. There are two constraints. One constraint is where the maximum stress should be less than $4e+7Pa$ in all of the members. The other constraint is where buckling should not occur in all of the members.

4.2 Comparison of the Pulling Back Method with the Penalty Method

The penalty method and the pulling back method are applied to truss structural optimization problems for the comparison. The target truss structures are the 6-node and 10-member structure and the 8-node and 15-member structure.

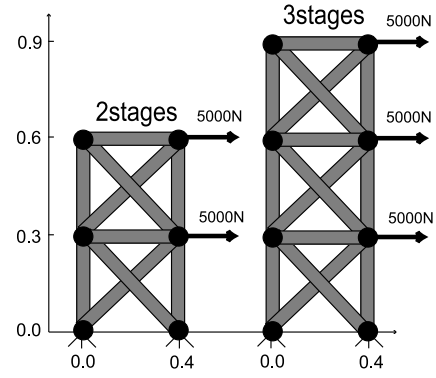


Figure 4: 2-Stage and 3-Stage Truss Structure

Table 1: Parameters of the DPMBGA

Number of Individuals(1 island)	16
Number of elites	1
Number of islands	4,8,16,32
Migration rate	0.0625
Migration interval	5
Archive size for PCA	100
Sampling rate	0.25
Amp. of Variance	2
Mutation rate	0.1/ (Dim. of function)

The used parameters of the DPMBGA are summarized in Table 1.

This example discusses the affect of the number of the islands to the solution when the number of the islands is changed from 1 to 32. The searching ability is determined by the number of trials that found the optimum and also by the average evaluation times when the optimum solution is found. The simulation is terminated when the evaluation time is over 1,000,000 for the 6-node and 10-member truss. For the 8-node and 15-member truss, the simulation is terminated when the evaluation time is over 2,000,000. When there is only one island, the PCA is performed in the island. When there are more than two islands, the distributed environment scheme is performed. The total trials are 25. The penalty parameter ρ is $1e+6$.

The number of the trials that find the optimum solution for the problem of the 6-node and 10-member truss is described in Table 2. The number of the trials that find the optimum solution for the problem of the 8-node and 15-member truss is described in Table 3. From Table 2 and Table 3, it is found that the bigger number of the islands leads the better solutions in the penalty method. On the other hand, the pulling back method can derive good solutions with the small number of the islands. In the 8-node and 15-member truss problem, the results of the pulling back method is better than those of the penalty method. From this result, it can be said that the pulling back method can

Table 2: Number of Trials that the Optimum is Found (2-Stage Truss)

Number of Island	Penalty	Pulling Back
1	0	25
2	6	25
4	14	25
8	18	25
16	22	25
32	25	25

Table 3: Number of Trials that the Optimum is Found (3-Stage Truss)

Number of islands	Penalty	Pulling Back
1	0	24
2	1	19
4	4	24
8	4	19
16	5	5
32	12	0

derive better solutions than the penalty method in the difficult problems.

The average evaluation times when the optimum solution is derived for the 6-node and 10-member truss structure are shown in Figure 5. In the same way, the results of the 8-node and 15-member truss is shown in Figure 6. From the results of the Figure 5 and Figure 6, it is found that the penalty method can find solutions with smaller evaluation times. For the pulling back method, the bigger evaluation times are needed when the number of the islands becomes bigger.

Generally, huge population size maintains the diversity of solutions. This leads to increased searching capability. In the penalty method, the increase in the number of islands and the total population size leads to a higher searching capability. On the other hand, when the number of islands is small, an early convergence may happen and it prevents an effective search. In the pulling back method, the individual who is out of the feasible field is pulled back to the boundary. This new point is different from the parent points. Therefore, this operation can keep the diversity even with a small number of islands. However, because the pulling back method requires the differentiation of constraints, a lot of evaluations are needed. Because of this, when the number of the total population is increased, the total calculation for evaluation is also increased. Therefore, the effectiveness of searching is decreased.

From these results, we can conclude as follows. For simple problems, the penalty method is effective. On the other hand, for difficult problems, the pulling back method has higher possibility to find the optimum than the penalty

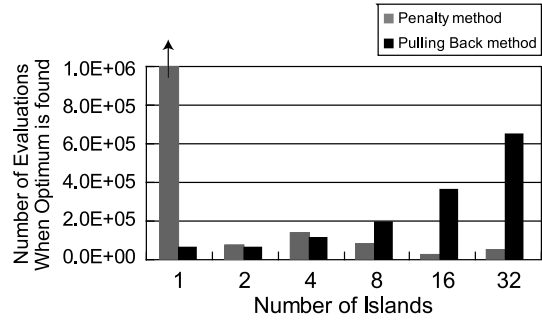


Figure 5: Average Evaluation Times When the Optimum is Found (2-Stage Truss)

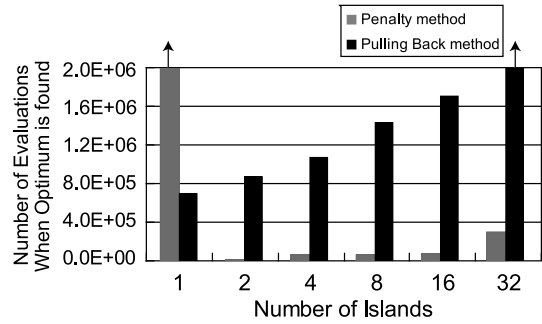


Figure 6: Average Evaluation Times When the Optimum is Found (3-Stage Truss)

method. In that case, the smaller number of the islands is better for the evaluation times.

4.3 Number of Individuals that Violate Constraints

In this section, the number of the individuals that do not satisfy the constraints is checked by both the penalty method and the pulling back method.

The rate of the individuals in each generation that violates the constraints for the 6-node and 10-member truss structure problem is shown in Figure 7. The average evaluation value is also shown in Figure 8. Both results are the median value of 25 trials with 32 islands and 512 individuals.

It is found from Figure 7 that 60% of population does not satisfy the constraints in the penalty method. On the other hand, only 30% of population does not satisfy the constraints in the pulling back method.

In Figure 8, the evaluation values of individuals that violate constraints are very high in the penalty method. The evaluation values of individuals that violate constraints are rather low in the pulling back method.

From these results, it is found that most of the searching points in the penalty method do not satisfy the con-

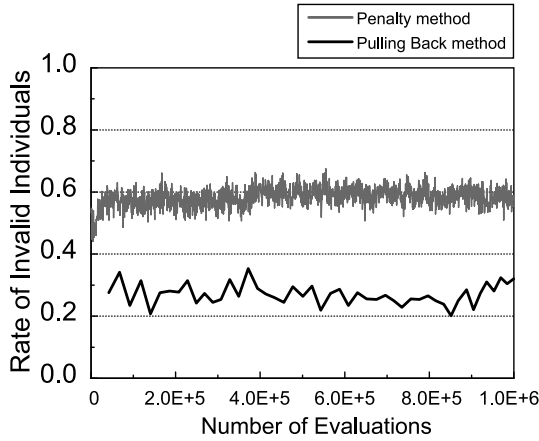


Figure 7: History of Rate of Invalid Individuals

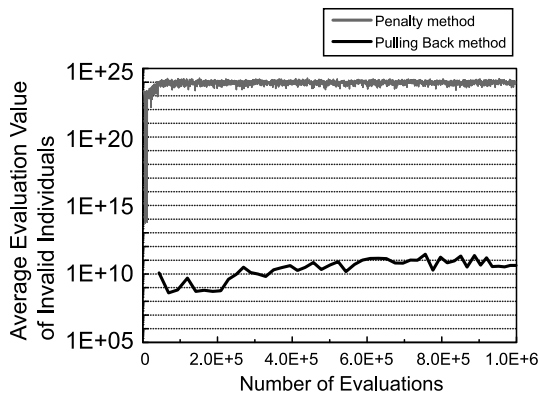


Figure 8: History of Average Evaluation Value of Invalid Individuals

straints. Therefore, an efficient search cannot be expected. To overcome this problem, many individuals are needed for obtaining the results. Therefore, when difficult problems are solved, huge populations should be prepared.

On the other hand, in the pulling back method, the individuals that do not satisfy the constraints are pulled back into the feasible region. Because of this mechanism, an effective search can be expected even for difficult problems.

4.4 Comparison of the Distribution of Individuals

In this section, the transition of the distribution of the population is compared between the pulling back method and the penalty method. The target problem is the 6-node and 10-member truss structure.

Figure 9 shows the transition of the distribution of the individuals derived by the penalty method that satisfy the constraints. Figure 10 is the transition of the individuals derived by the pulling back method. In these figures, the

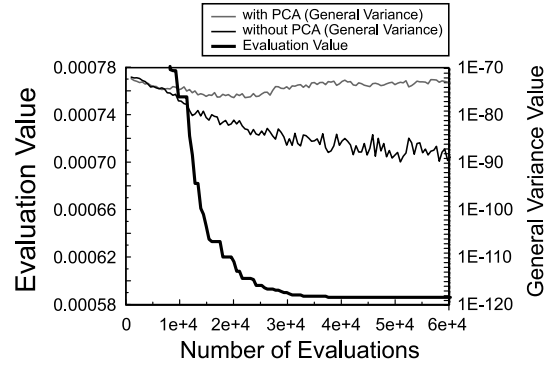


Figure 9: History of General Variance Value in the Penalty Method (32 islands)

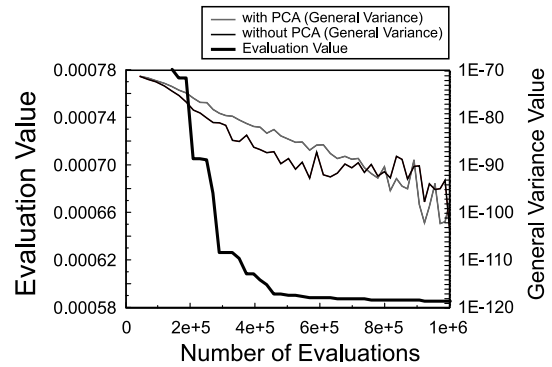


Figure 10: History of General Variance Value in the Pulling Back Method (32 islands)

transition of the average evaluation value is also illustrated. These results are the median of 25 trials and are derived with 32 islands and 512 individuals. In Figure 11, the results of the penalty method with 8 islands and 128 individuals are shown.

In the results of the penalty method, the population is converged in the island where the PCA is not performed. On the other hand, the population is not converged in the island where the PCA is performed. Therefore, in the latter stage of the search, the islands where the PCA is performed may not help to find the solution.

In the results of the pulling back method, the population is converged in both islands where the PCA is performed or not.

In Figure 11, when the number of the islands becomes small in the penalty method, the general variance value becomes small. This means that the diversity of the solutions is lost.

From these results, it can be concluded that more individuals of the pulling back method are used for searching than those of the penalty method while some individuals are not used when searching the solution in the penalty method.

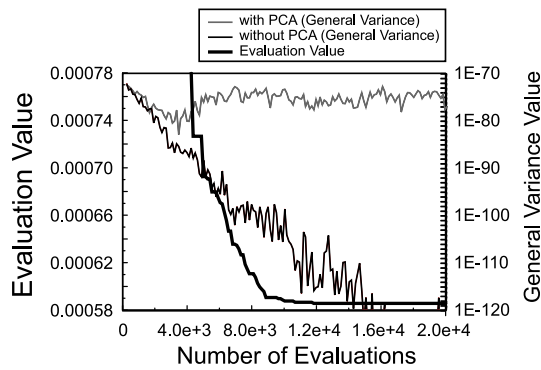


Figure 11: History of General Variance Value in the Penalty Function Method (8 islands)

When the number of the islands becomes small, the diversity of the solutions is lost and the searching ability of the DPMBGA with the penalty method decreases.

5 Conclusions

For complex optimization problems with constraints, GAs that have high searching capabilities and that have mechanisms that can treat constraints are necessary. In this paper, DPMBGA is a new extended algorithm of PMBGA is discussed. At the same time, the operations of the penalty method and the pulling back method that can treat constraints for the GAs are compared.

In the DPMBGA, in order to maintain the diversity of the solutions during the search, the distributed GA model is applied. At the same time, since the probabilistic model is developed using the PCA, an effective search can be expected.

In the pulling back method, the new individual that violates the constraints is moved to the nearest point that satisfies the constraints. The pulling back method is compared with the penalty method. This comparison is performed through truss structural optimization problems. Truss structural optimization problems are real-world problems that have several types of constraints. Through experiments, the following results are obtained. The algorithm using the penalty method and the DPMBGA with many islands is effective for small problems. On the other hand, for big problems, the algorithm using the pulling back method and the DPMBGA with a small number of the islands is effective.

Through simulations, the number of individuals that are out of the feasible region is checked. Compared to the penalty method, the pulling back method creates very few individuals that are out of the feasible region. Because of this tendency, the pulling back method can perform an effective search.

Bibliography

- [Goldberg89] Goldberg,D.E. *Genetic Algorithms in Search Optimization and Machine Learnig*, Addison-Wesley, 1989.
- [Holland75] Holland,J.H. *Adaptation In Natural and Artificial Systems*, University of Michigan Press,1975.
- [Pelikan99] Pelikan,M., Goldberg,D.E., and Lobo,F. *A Survey of Optimization by Building and Using Probabilistic Models*, Technical Report 99018, IlliGAL,1999.
- [Larranaga01] Larranaga,P., Lozano,J.A. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, Kluwer Academic Publishers,2001.
- [Tomoyuki03] Tomoyuki,H., Miki,M., Sano,M., Shimosaka,H., Tsutsui,S. and Dongarra,J. *Distributed Probabilistic Model-Building Genetic Algorithm*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2003), pp.1015 - 1028, 2003.
- [Tanese89] Tanese,R. *Distributed Genetic Algorithms*, Proc. 3rd International Conference on Genetic Algorithms,pp.434 - 439,1989.
- [Jolliffe86] Jolliffe,I.T. *Principal Component Analysis*, Springer-Verlog,New York,1986.
- [Coello99] Coello Coello,A.C. *A Survey of Constraint Handling Techniques used with Evolutionary Algorithms*, Technical Report Lania-RI-99-04,Laboratorio Nacional de Informática Avanzada,1999.
- [Richardson89] Richardson,J.T., Palmer,M.R., Liepins,G. and Hilliard,M. *Some Guidelines for Genetic Algorithms with Penalty Functions*, In J. David Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, pp.191 - 197,1989.
- [Mimura02] Mimura,Y., Yoshimura,S., Hiroyasu,T. and Miki,M. *Development of Multi-Stage and Hybrid Real-Coded Genetic Algorithms with Active Constraints*, Proceedings of 2002 ASME Design Engineering Technical Conferences (DETC) & The Computers and Information in Engineering Conference, DETC2002/DAC-34059(CD-ROM).2002.
- [Paz98] Cantú-Paz,E. *A survey of parallel genetic algorithms*, Calculateurs Paralleles,Vol.2,No.10,2001.
- [Miki99] Miki,M., Hiroyasu,T., Kaneko,M. and Hatanaka,K. *A Parallel Genetic Algorithm with Distributed Environment Scheme*, IEEE Proceedings of Systems, Man and Cybernetics Conference SMC'99,1999.

SOURCE :

Hisashi Shimosaka, Tomoyuki Hiroyasu, Mitsunori Miki
Comparison of Pulling Back and Penalty Methods for Constraints in DPMBGA

The IEEE International Congress on Evolutionary Computation (CEC2003)

Canberra - Australia, December 8-12, 2003