

# A METHOD FOR THE MIXED DISCRETE NON-LINEAR PROBLEMS BY PARTICLE SWARM OPTIMIZATION

S. Kitayama,<sup>1</sup> M. Arakawa,<sup>2</sup> and K. Yamazaki<sup>1</sup>

<sup>1</sup>Kanazawa University, Kakuma-machi, Kanazawa, 920-1192, JAPAN, [kitagon@t.kanazawa-u.ac.jp](mailto:kitagon@t.kanazawa-u.ac.jp) <sup>2</sup>Kagawa University, 2217-20 Hayashi-cho, Takamatsu, Kagawa, 761-0396, JAPAN

**Abstract** An approach for the Mixed Discrete Non-Linear Problems (MDNLP) by Particle Swarm Optimization is proposed. The penalty function to handle the discrete design variables is employed, in which the discrete design variables are treated as the continuous design variables by penalizing at the intervals. By using the penalty function, it is possible to handle all design variables as the continuous design variables. Through typical benchmark problem, the validity of proposed approach for MDNLP is examined.

**keywords:** Global Optimization, Particle Swarm Optimization, Mixed Discrete Non-Linear Problems

## 1. Introduction

Particle Swarm Optimization (PSO), which mimics the social behavior, is an optimization technique developed by Kennedy et. al. [1]. It has been reported that PSO is suitable for the minimization of the non-convex function of the continuous design variables through many numerical examples. Few researches of PSO have been reported about the discrete optimization [2]. These researches handle the discrete design variables as the continuous ones, directly. That is, firstly all design variables may be handled as the continuous ones, and optimized. Finally, the round-off or cut-off are applied to get the discrete optimum. These approaches may be valid in a sense, but some problems are included as shown in Fig.1(a), (b).

Fig.1(a) shows a case.  $x_L$  represents the optimum of the continuous design variables. Point A and B represent the discrete design variables close to  $x_L$ . In this case, Point B is chosen as the neighborhood of  $x_L$  by the round-off. However, the objective function at Point B makes a change of the function value worse, when compared with the objective function at Point A. [3] Another

Please use the following format when citing this chapter:

Author(s) [insert Last name, First-name initial(s)], 2006, in IFIP International Federation for Information Processing, Volume 199, System Modeling and Optimization, eds. Ceragioli F., Dontchev A., Furuta H., Marti K., Pandolfi L., (Boston: Springer), pp. [insert page numbers].

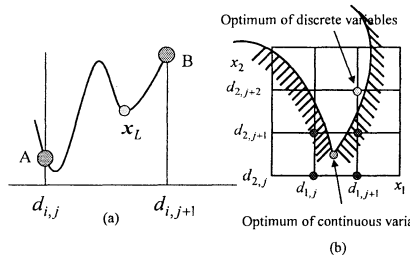


Figure 1. The nature of discrete optimization

case shown in Fig.1(b) is well known. That is, the optimum obtained by the round-off or the cut-off does not satisfy all feasibilities [4].

PSO is suitable for the global optimization of the non-convex function of the continuous design variables. Therefore, all design variables should be handled as the continuous ones whenever PSO is applied to the mixed or discrete design variables problems.

In this paper, the penalty function approach to handle the discrete design variables is proposed. By using the penalty function for the discrete design variables, it is possible to handle the discrete design variables as the continuous ones. Through typical MDNLP, the validity of proposed approach is examined.

## 2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is one of the global optimization methods for the continuous design variables [6]. PSO does not utilize the gradient information of function like Genetic Algorithm. In PSO, each particle has the position and velocity, and they are updated by a simple addition and subtraction of vectors during search process.

The position and velocity of particle  $d$  are represented by  $x_d^k$  and  $v_d^k$ , respectively.  $k$  represents iteration. The position and velocity of particle  $d$  at  $k+1$  th iteration are calculated by following equations.

$$x_d^{k+1} = x_d^k + v_d^{k+1} \tag{1}$$

$$v_d^{k+1} = w \times v_d^k + c_1 r_1 \times (p_d^k - x_d^k) + c_2 r_2 \times (p_g^k - x_d^k) \tag{2}$$

in which the coefficient  $w$  is called as inertia term, and  $r_1$  and  $r_2$  denote random number between  $[0,1)$ . The weighting coefficients  $c_1$  and  $c_2$  are parameters. In general,  $c_1 = c_2 = 2$  is often used.  $p_d^k$  called as  $pbest$ , represents the best position of particle  $d$  till  $k$  th iteration, and  $p_g^k$  called as  $gbest$ , represents the best position in the swarm till  $k$  th iteration. That is,  $p_g^k$  is chosen among  $p_d^k$ .

The inertia term in Eq.(2) gradually decreases during the search iteration.

$$w = w_{max} - \frac{w_{max} - w_{min}}{k_{max}} \times k \quad (3)$$

in which,  $w_{max}$  and  $w_{min}$  represent the maximum and minimum value of inertia.  $k_{max}$  represents the maximum number of search iteration. In general,  $w_{max} = 0.9$  and  $w_{min}=0.4$  are recommended [7].

## 2.1 PSO as an Optimization Technique

From Eq.(1) and Eq.(2), the following equation can be obtained.

$$\mathbf{x}_d^{k+1} = \mathbf{x}_d^k + w \times \mathbf{v}_d^k + \alpha(\mathbf{q} - \mathbf{x}_d^k) \quad (4)$$

in which  $\alpha$  and  $\mathbf{q}$  are represented as follows, respectively.

$$\alpha = c_1 r_1 + c_2 r_2 \quad (5)$$

$$\mathbf{q} = \frac{c_1 r_1 \mathbf{p}_d^k + c_2 r_2 \mathbf{p}_g^k}{c_1 r_1 + c_2 r_2} \quad (6)$$

From Eq.(4), it is possible to interpret that  $\mathbf{q} - \mathbf{x}_d^k$  represents the search direction when we imagine the similarity to the gradient methods.  $\alpha$  in Eq.(4) also may be regarded as stochastic step-size, in which its lower and upper bounds are 0 and  $c_1+c_2$ , and the mean value is  $(c_1 + c_2)/2$ . From these relationships, it is possible to consider that PSO has a search direction vector and stochastic step-size even though PSO does not utilize the gradient information of function.

## 3. Penalty Function Approach for MDNLP by PSO

### 3.1 Problem Definition

In general, the Mixed Discrete Non-Linear Problem (MDNLP) is described as follows [5]:

$$f(\mathbf{x}) \rightarrow \min \quad (7)$$

$$x_i^L \leq x_i \leq x_i^U \quad i = 1, 2, \dots, m \quad (8)$$

$$x_{m+i} \in D_i \quad D_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,q}\} \quad i = 1, 2, \dots, n \quad (9)$$

$$g_k(\mathbf{x}) \leq 0 \quad k = 1, 2, \dots, ncon \quad (10)$$

where  $\mathbf{x}$  represents the design variables, which consist of the continuous and discrete design variables.  $f(\mathbf{x})$  is the objective function, and  $g_k(\mathbf{x})$  is the behavior constraints.  $ncon$  represents the number of behavior constraints.  $x_i$  denotes

the continuous design variables, and  $m$  is the total number of continuous design variables.  $x_i^L$  and  $x_i^U$  denote the lower and upper bounds of continuous design variables, respectively. On the other hand,  $n$  is the total number of discrete design variables.  $D_i$  is the set of discrete values for the  $i$ -th discrete design variable.  $d_{i,j}$  is the  $j$ -th discrete value for the  $i$ -th discrete design variables.  $q$  represents the number of discrete values.

### 3.2 Penalty Function

In this paper, the following penalty function suggested by [8] is adopted.

$$\phi(\mathbf{x}) = \sum_{i=1}^n \frac{1}{2} \sin \left[ \frac{2\pi \times \{x_{m+i}^c - 0.25(d_{i,j+1} + 3d_{i,j})\}}{d_{i,j+1} - d_{i,j}} + 1 \right] \quad (11)$$

where  $x_{m+i}^c$  is the continuous design variables between  $d_{i,j}$  and  $d_{i,j+1}$ . Then the augmented objective function  $F(\mathbf{x})$  is constructed by using above penalty function as follows:

$$F(\mathbf{x}) = f(\mathbf{x}) + s \times \phi(\mathbf{x}) + r \sum_{k=1}^{ncon} \max[0, g_k(\mathbf{x})] \quad (12)$$

in which  $s$  and  $r$  denote the penalty parameters for Eq.(11) and Eq.(10), respectively. Finally, MDNLP is transformed into the following continuous design variables problem.

$$F(\mathbf{x}) \rightarrow \min \quad (13)$$

$$x_i^L \leq x_i \leq x_i^U \quad i = 1, 2, \dots, m \quad (14)$$

$$d_{i,1} \leq x_{m+i}^c \leq d_{i,q} \quad i = 1, 2, \dots, n \quad (15)$$

For the simplicity, the design variables are supposed as the discrete design variables in the following discussion. In the case of the mixed design variables, we discuss at section 3.7 separately.

### 3.3 Characteristics of Penalty Function

The value of Eq.(11) becomes small around the neighborhood of discrete value. On the other hand, the value of Eq.(11) becomes large, turning from discrete value. When  $p_g^k$  satisfies the following equation, the discrete value resides around the neighborhood of  $p_g^k$ .

$$\phi(p_g^k) \leq \varepsilon \quad (16)$$

$\varepsilon$  in Eq.(16) represents small positive value. As a result, the penalty parameter  $s$  in Eq.(12) must be updated so that Eq.(16) is satisfied. In order to examine the effect of the penalty parameter  $s$ , let us consider a following simple problem.

$$f(x) = x^4 - \frac{8}{3} - 2x^2 + 8x \rightarrow \min \quad (17)$$

$$x \in \{-1, 0, 1, 2\} \quad (18)$$

In this simple example, the objective  $f(x)$  and the augmented objective function  $F(x)$  are shown in Fig.2.

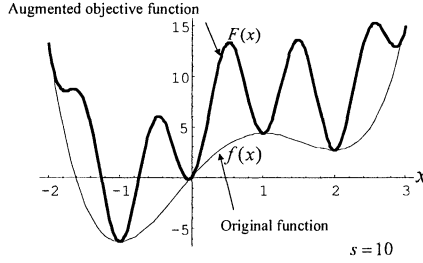


Figure 2. Behavior of the augmented objective function

From Fig.2, it is apparent that  $F(x)$  becomes non-convex and continuous. Additionally many local minima are generated around the neighborhood of the discrete values. As a result, the problem to find the discrete optimum is transformed into finding global minimum of  $F(x)$ . Additionally, the discrete values are given at the point, where the relative error between  $f(x)$  and  $F(x)$  becomes small. The following equation is utilized as terminal criteria.

$$\frac{|F(\mathbf{p}_g^k) - f(\mathbf{p}_g^k)|}{|F(\mathbf{p}_g^k)|} \leq \varepsilon \quad (19)$$

PSO does not use the gradient information of function, so that it is difficult to satisfy Eq.(16) strictly. Then, Eq.(19) is used instead of that.

Behaviors of  $F(x)$  for various penalty parameter  $s$  are shown in Fig.3. From Fig.3, it is found that to determine a penalty parameter  $s$  in advance is very difficult.

### 3.4 Initial Penalty Parameter $s$

An initial search point  $\mathbf{x}_d$  of particle  $d$  is determined randomly. Then the value of penalty function in Eq.(11) is calculated for each particle. The penalty parameter  $s$  is determined as follows.

$$s_d = 1 + \phi(\mathbf{x}_d) \quad d = 1, 2, \dots, agent \quad (20)$$

where  $s_d$  represents the penalty parameter of particle  $d$ .  $agent$  in Eq.(20) is the total number of particles. And initial penalty parameter  $s_{initial}$  is determined

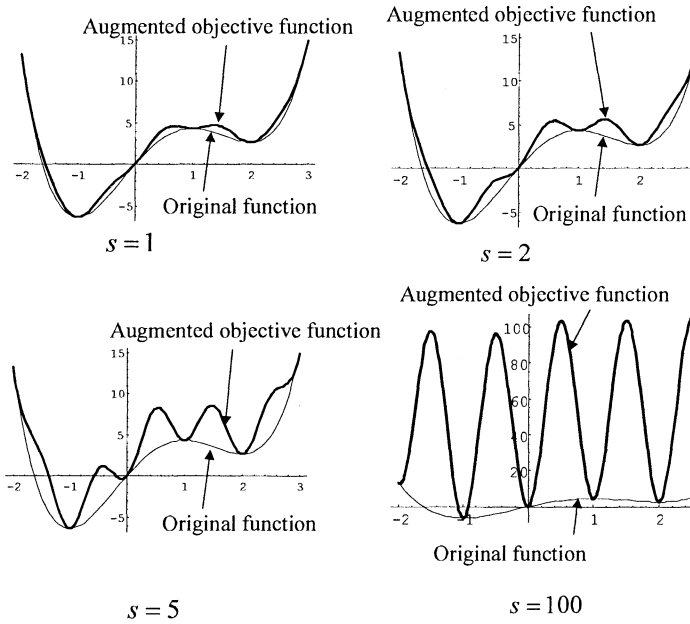


Figure 3. Behavior of the augmented objective function for some parameter

as follows.

$$s_{initial} = \min\{s_1, s_2, \dots, s_{agnet}\} \tag{21}$$

At the initial stage to search the optimum,  $F(x)$  is actively transformed into non-convex and continuous function, and many local minima are enenerated around the neighborhood of the discrete value.

### 3.5 Update Scheme of Penalty Parameter $s$

The following equation is used to update the penalty parameter  $s$ .

$$s = s \times \exp(1 + \phi(p_g^k)) \tag{22}$$

The behavior of  $F(x)$  by updating the penalty parameter  $s$  is shown schematically in Fig.4.

In Fig.4, solid line shows  $F(x)$  at  $k$  th iteration, and dotted line shows  $F(x)$  at  $k+1$  th iteration. As shown in Fig.4,  $F(x)$  at  $k+1$  th iteration becomes highly non-convex function in comparison with  $F(x)$  at  $k$  th iteration. For example, point A in Fig.4 corresponds to the point  $p_g^k$  at  $k$  th iteration. By updating

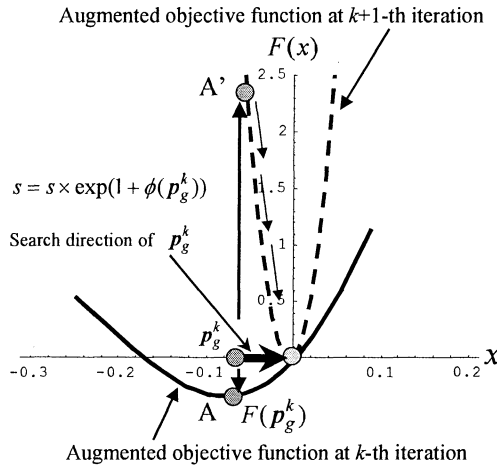


Figure 4. Update of penalty parameter  $s$

penalty parameter  $s$ ,  $p_g^k$  corresponds to the point A' on the dotted line. As discussed in section 2.1, PSO has similar structure to the gradient methods, so that it is expected that  $p_g^k$  moves to the direction in Fig.4. Finally, it is also expected that  $p_g^k$  will satisfy Eq.(19).

### 3.6 Initialization of Penalty Parameter $s$

When Eq.(19) is satisfied, the discrete value around the neighborhood of  $p_g^k$  resides. Then an initial penalty parameter by Eq.(21) is utilized in order to find another discrete value, because  $F(x)$  becomes highly non-convex function by updating the penalty parameter  $s$ . It is assumed that  $p_g^k$  fails to escape from local minimum. In such occasions,  $F(x)$  is relaxed by using an initial penalty parameter when Eq.(19) is satisfied. As a result, it is expected that  $p_g^k$  can escape from local minimum.

### 3.7 In the Case of Mixed Design Variables

The component of  $p_g^k$  can be expressed as follows:

$$p_g^k = (x^{cont}, x^{discret})^T \tag{23}$$

where  $x^{cont}$  and  $x^{discret}$  represent the components of the continuous and discrete design variables, respectively. Then, the components of the continuous design variables  $x^{cont}$  in  $p_g^k$  are neglected when the terminal criteria by Eq.(19) is

applied. That is, only the components of the discrete design variables  $x^{discrt}$  in  $p_g^k$  are checked when the terminal criteria by Eq.(19) is utilized.

### 3.8 Difference between Traditional and Proposed Method

The penalty function of Eq.(11) is the same as [8]. However, its approach is very different from each other. Shin et. al. have searched an optimum by regarding all design variables as the continuous at the initial stage, the penalty parameter  $s$  in Eq.(12) has been set as zero. After the optimum obtained by regarding all design variables as the continuous has been found, the penalty function of Eq.(11) has been introduced to avoid the search procedure of global minimum among many local minima of  $F(x)$ .

On the other hand, the penalty parameter  $s$  is actively introduced at the initial stage in our approach. Obviously  $F(x)$  becomes non-convex and continuous. However, this is not serious problem because PSO is applied to  $F(x)$ . The new update scheme of penalty parameter  $s$  by Eq.(22) is proposed. In the past reports [4, 8], the constant positive number is used to update the penalty parameter. However, the constant positive number depends on the problems. On the other hand, the penalty parameter  $s$  may always changes in our approach because the value of  $\phi(p_g^k)$  is utilized. It may be expected that flexible applications may be possible. Finally, the initialization of the penalty parameter  $s$  is also introduced in order to relax  $F(x)$ . As a result, it is expected that  $p_g^k$  can escape from local minimum.

Binary PSO is also easy to handle the discrete design variables [9, 10]. However, the search process of binary PSO is stochastic. Additionally, no proof that the objective or design domain is continuous is given. On the other hand, our approach adopted here utilizes the characteristics of PSO and the penalty function of Eq.(11), in order to find optimum. That is, our approach may be deterministic, when compared with binary PSO.

### 3.9 Algorithm

The proposed algorithm for MDNLP by PSO is shown in Fig.5.

## 4. Numerical Example

To examine the validity of proposed approach, let us consider the optimum design of pressure vessel as shown in Fig.6.

This problem is one of the most famous benchmark for MDNLP [9, 11–14]. Several results are shown in table 1. From table 1, it is very difficult to find optimum solutions even though this problem consists of only 4 design variables. The design variables are 1) Radius  $R$  (continuous design variables:  $x_1$ ), 2) Length  $L$  (continuous design variables:  $x_2$ ), 3) Thickness  $T_s$  (discrete



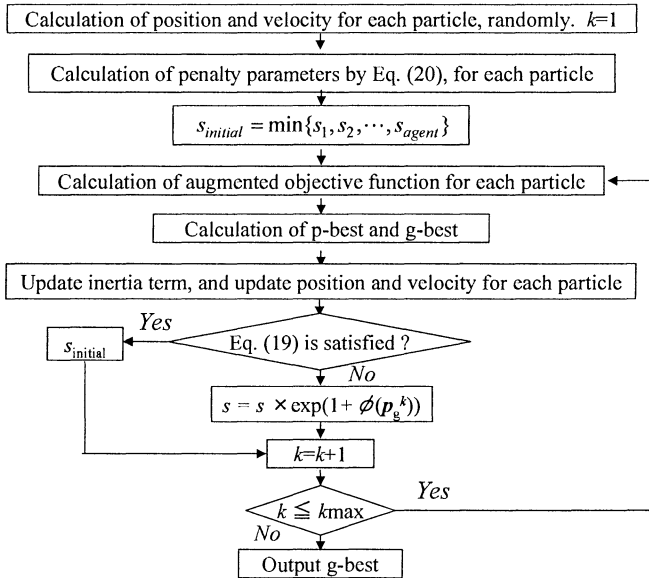


Figure 5. The algorithm for MDNLP by PSO

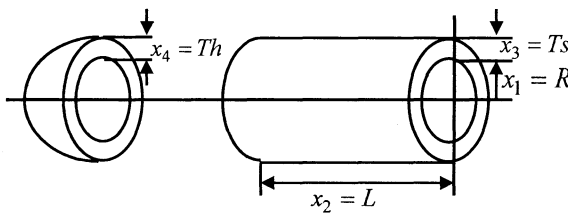


Figure 6. Optimum Design of Pressure Vessel

design variables:  $x_3$ ), and 4) Thickness  $T_h$  (discrete design variables:  $x_4$ ). An optimization problem is defined as follows.

$$f(\mathbf{x}) = 0.6224x_1x_2x_3 + 1.7781x_1^2x_4 + 3.1661x_2x_3^2 + 19.84x_1x_3^2 \rightarrow \min \tag{24}$$

$$10 \leq x_1, x_2 \leq 200 \tag{25}$$

$$0.0625 \leq x_3, x_4 \leq 6.1875 \tag{26}$$

$$g_1(\mathbf{x}) = \frac{0.0193x_1}{x_3} - 1 \leq 0 \tag{27}$$

$$g_2(\mathbf{x}) = \frac{0.00954x_1}{x_4} - 1 \leq 0 \quad (28)$$

$$g_3(\mathbf{x}) = \frac{x_2}{240} - 1 \leq 0 \quad (29)$$

$$g_4(\mathbf{x}) = \frac{1296000 - \frac{4}{3}\pi x_1^3}{\pi x_1^2 x_2} - 1 \leq 0 \quad (30)$$

in which  $x_3$  and  $x_4$  are the discrete design variables, and are integer multiples of 0.0625 inch.

Behavior constraints from Eq.(27) to Eq.(30) are handled as penalty function by using Eq.(12). The penalty parameter  $r$  is set as  $1.0 \times 10^8$ . The number of particle is set as 50, and the maximum number of search iteration  $k_{max}$  is also set as 500. 10 trials are performed with different random seed. The best result during 10 trials is shown in the last column "Kitayama" in table 1. From table 1, best result could be obtained by our proposed method. The average of function calls through 10 trials is 22500.

Table 1. Comparison of results

	Sandgren	Qian	Kannan	Hsu	He	Kitayama
$R$ [inch]: $x_1$	47.00	58.31	58.29	N/A	42.10	42.37
$L$ [inch]: $x_2$	117.70	44.52	43.69	N/A	176.64	173.42
$T_s$ [inch]: $x_3$	1.125	1.125	1.125	N/A	0.8125	0.8125
$T_h$ [inch]: $x_4$	0.625	0.625	0.625	N/A	0.4375	0.4375
$g_1(\mathbf{x})$	-0.19	0.00	0.00	N/A	0.00	0.00
$g_2(\mathbf{x})$	-0.28	-0.11	-0.11	N/A	-0.08	-0.08
$g_3(\mathbf{x})$	-0.51	-0.81	-0.82	N/A	-0.26	-0.28
$g_4(\mathbf{x})$	0.05	-0.02	-1.11	N/A	0.00	0.00
<i>Objective</i>	8129.80	7238.83	7198.20	7021.67	6059.71	6029.87

## 5. Conclusions

In this paper, PSO has been applied to MDNLP. The penalty function for the discrete design variables is introduced, in order to handle as the continuous design variables. The augmented objective function becomes non-convex function of continuous design variables, by introducing penalty function. As considered that PSO is naturally suitable for the global search of non-convex function of the continuous design variables, our proposed approach may be valid. A method to determine the penalty parameter  $s$  and new update scheme of penalty parameter  $s$  have been also proposed. Through typical benchmark problem, the validity has been examined.

## References

- [1] J.Kennedy, R.C.Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.
- [2] G.Venter, J.S.Sobieski. Particle Swarm Optimization. *9-th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference*, AIAA2002-1235, 2002.
- [3] S.S.Rao. *Engineering Optimization: Theory and Practice*. Wiley InterScience, 1996.
- [4] J.F.Fu, R.G.Fenton, W.L.Cleghorn. A Mixed Integer-Discrete Continuous Programming and its Application to Engineering Design Optimization. *Eng. Opt.* 17:263-280, 1996.
- [5] J.S.Arora, M.W.Huang. Methods for Optimization of Nonlinear Problems with Discrete Variables: A Review. *Struc. Opt.* 8:69-85, 1994.
- [6] K.E.Parsopoulos, M.N.Vrahatis. Recent Approaches to Global Optimization Problems through Particle Swarm Optimization. *Neural Computing.* 1:235-306, 2002.
- [7] Y.Fukuyama. A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment. *IEEE Trans. Power Syst.* 15-4:1232-1239. 2000.
- [8] D.K.Shin, Z.Gurdal, O.H.Griffin. A Penalty Approach for Nonlinear Optimization with Discrete Design Variables. *Eng. Opt.* 16:29-42, 1990.
- [9] S.He, E.Prempan, Q.H.Wu. An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems. *Eng. Opt.* 35-5:585-605, 2004.
- [10] J.Kennedy, R.C.Eberhart. A Discrete Binary Version of the Particle Swarm Optimization Algorithm. *Proc. of the Conference on Systems, Man, and Cybernetics.* 4104-4109, 1997.
- [11] E.Sandgren. Nonlinear and Discrete Programming in Mechanical Design Optimization. *Trans. of ASME, J. of Mech. Des.* 112:pp.223-229, 1990.
- [12] B.K.Kannan, S.N.Kramer. An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design. *Trans. of ASME, J. of Mech. Des.* 116:405-411, 1994.
- [13] Z.Qian, J.Yu, J.Zhou. A Genetic Algorithm for Solving Mixed Discrete Optimization Problems. *DE-Vol.65-1, Advances in Design Automation.* 1:499-503, 1993.
- [14] Y.H.Hsu, L.H.Leu. A Two Stage Sequential Approximation Method for Non-linear Discrete Variable Optimization. *ASME/DETC/DAC MA* 197-202, 1995.