Robust Multiple Optimization for Adjustable Mechanisms

Masatoshi MURAMATSU*[1], Keigo YAMAMOTO*[2] and Takeo KATO*[3]
*1, 2, 3 Department of Mechanical Engineering, Tokai University
4-1-1 Kitakaname, Hiratsuka-shi, Kanagawa 259-1292, JAPAN
m.muramatsu@fuji.tokai-u.jp, 0bem2103@mail.tokai-u.jp, t.kato@tokai-u.jp

Abstract
Robust design that enables the products to ensure robust performance in diverse users and the surroundings has received much attention. The previous research proposed a robust design method derived the optimal adjustable ranges of the adjustable control factors by using the optimization method which considers only robustness. However, deriving the small adjustable ranges is difficult using this method. Consequently, this research proposed a robust multiple optimization method using the Vector evaluated particle swarm optimization in order to consider both robustness and the adjustable range. Design examples are presented to demonstrate the applicability of the proposed method.
Keywords: robust design, design methodology, particle swarm optimization, seat design

1 Introduction
Recently, globalization of markets has diversified both the users and the surroundings. Consequently, a robust design is crucial to maintain the function of the design object [1] because it ensures function stability (hereinafter called robustness). Thus, the fluctuation of objective characteristics y, which expresses a design objective, due to the fluctuations of control factors $\boldsymbol{x}$, which designers can control, and noise factors $\boldsymbol{z}$, which designers cannot control, must be decreased. In most conventional methods, designers set control factors to fixed values to maximize the robustness. In cases where the objective characteristic distribution is smaller than the tolerance (**Fig.1a**), these methods can derive a design solution (optimized control factor values) x_0 with sufficient robustness. However, in cases where the objective characteristic distribution is larger than the tolerance **(Fig.1b**), a solution to sufficiently maximize robustness cannot be obtained. In such cases, the control factors must be adjusted to ensure robustness. In other words, as the values of the control factors are varied, the whole of the objective characteristic distribution should be located within the tolerance (**Fig.1c**). Unfortunately, securing sufficient robustness is difficult using these methods when the distributions of the objective characteristics are significantly larger than their tolerance.

In a previous research, a robust design method for adjustable mechanism was proposed to overcome the problem where a sufficient function cannot be ensured for diverse users or surroundings [2, 3]. In this method, control factors whose values are adjusted by adjustable mechanisms are defined as adjustable control factors $\boldsymbol{t}$ (hereinafter called ACFs). Because this method properly evaluates the robustness, it allows designers to judge the employment of the adjustable mechanisms as well as to set the ranges where the ACF values are adjustable. These ranges, called adjustable ranges, are represented as $[\boldsymbol{t}_\mathrm{l}, \boldsymbol{t}_\mathrm{u}]$ where $\boldsymbol{t}_\mathrm{l}$ and $\boldsymbol{t}_\mathrm{u}$ denote the lower and upper limit of the ACF values, respectively.

The concept of robustness in this research is defined below. If the tolerance of objective characteristic $[y_\mathrm{l}, y_\mathrm{u}]$ exists as shown in **Fig. 1**, then ACFs can be adjusted to locate each fluctuation value of the objective characteristic within the tolerance. Hence, the robustness index for ACFs (R_A) is defined as the feasibility that the objective characteristic values are within the tolerance at least once, by the adjustment of ACFs. Using ACFs and R_A, design problems in this research are expressed as shown in the following equation.

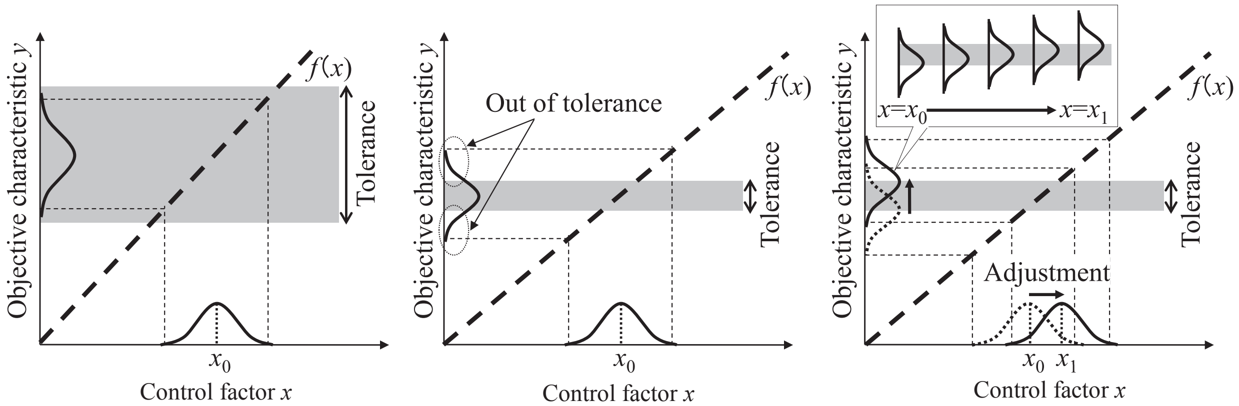
$$\begin{aligned}&\text{Find } [\boldsymbol{t}_\mathrm{l}, \boldsymbol{t}_\mathrm{u}],\ \boldsymbol{x}\\ &\text{to maximize } R_\mathrm{A}\left(y\left(= f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{t})\right) \right)\\ &\text{to minimize } \left|\boldsymbol{t}_\mathrm{u} - \boldsymbol{t}_\mathrm{l}\right|\end{aligned} \quad , \qquad (1)$$

where f is the objective function. To prevent an unnecessary expansion of the ACF ranges, which increases the production costs and failure rate, this formulation does not only maximize R_A. Minimizing the size of the range described in eq. (1) is an example of preventing an unnecessary expansion because other factors (e.g., the form and location of the range) can lead to the aforementioned issues.

Additionally, R_A is calculated as the ratio of the sum of the sets of fluctuant combinations of $\boldsymbol{x}$ and $\boldsymbol{z}$ where at least one of the objective characteristic values y_j derived from $\boldsymbol{t}_j$ is within the tolerance as shown in the following equation.

$$R_\mathrm{A} = P\left[\bigcup_{j=1}^{n_\mathrm{ap}} \left\{ C(\boldsymbol{x}, \boldsymbol{z}) \middle| \, y_\mathrm{l} \le f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{t}_j) \le y_\mathrm{u} \right\} \right] , \qquad (2)$$

where the square bracket expresses a set of $C(\boldsymbol{x}, \boldsymbol{z})$ where the objective characteristic is located within the tolerance by adjusting ACFs. This means R_A is the rate of the set and the entire set (**Fig. 2**). The assignable point values are expressed as a finite number of discontinuous values $\boldsymbol{t}_j$ because R_A is calculated using the Monte Carlo method. The number of the assignable (discontinuous) values



(a) Objective characteristic with wide tolerance and control factor with unique value
(b) Objective characteristic with narrow tolerance and control factor with nique value
(c) Objective characteristic with narrow tolerance and control factor with adjustable value

Fig. 1 Conceptual illustration of a design problem that includes a factor whose value is adjustable

should be sufficient to assume the ACF is continuous. However, the number should be decreased if the calculation amount is too large. To calculate R_A, first, s random combinations of the control and noise factors are generated based on their probability density functions. Second, objective characteristic y_i is calculated using the generated random combinations $\{\boldsymbol{x}_i, \boldsymbol{z}_i\}$ (i=1, 2,..., s) and all the assignable point values. That is, the number of calculating objective characteristic values is the product of the random combination number s and the assignable points numbers of ACFs. Finally, the values calculated from each random combinations of $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ are assessed to determine whether at least one of the calculated values is within the tolerance (i.e., at least one assignable point which consists an objective characteristic value that satisfies the tolerance). Then R_A is calculated as:

$$R_{\mathrm{A}} = \frac{1}{s}\sum_{i=1}^{s} M_i$$
$$\left(\begin{array}{l} M_i = \begin{cases} 1 & \left(\exists \boldsymbol{t} \in \{\boldsymbol{t}_j\}\ ;\ y_{\mathrm{l}} \le f(\boldsymbol{x}_i, \boldsymbol{z}_i, \boldsymbol{t}) \le y_{\mathrm{u}}\right) \\ 0 & (\text{otherwise}) \end{cases} \\ j = 1,2,\ldots,n_{\mathrm{ap}} \end{array} \right) . \quad (3)$$

The previous method derived the combination of the ACFs using the genetic algorithm (GA) whose fitness is assigned R_A. However, deriving the small adjustable ranges is difficult using this method because it is not considers those.

Below are definitions and descriptions of the terminologies used in this paper.

Objective characteristic ($y = f(x, z, t)$): The characteristic to express the function of the design objective, and is calculated by objective function f.

Control factors ($\boldsymbol{x} = \{x_i\}$, i=1, 2,..., n_x): Factors whose nominal values are set by the designer, but fluctuate the objective characteristic. n_x represents the total number of control factors.

Noise factors ($\boldsymbol{z} = \{z_i\}$, i=1, 2,..., n_z): Factors that fluctuate the objective characteristic, but their nominal

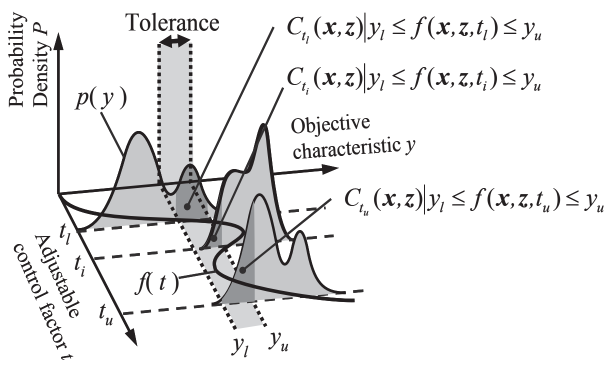


Fig. 2 Concept of a robustness index

values cannot be set by designers. n_z expresses the total number of noise factors.

ACFs ($\boldsymbol{t} = \{t_i\}$, i=1, 2,..., n_t): Control factors with nominal values that can be adjusted within their adjustable ranges. n_t denotes the total number of ACFs.

Adjustable range of ACFs ([$\boldsymbol{t}_u$, $\boldsymbol{t}_l$]): The range, defined by the designer, where the ACFs are adjustable.

Assignable points of ACFs ($\{\boldsymbol{t}_j\}$, j=1, 2,..., n_{ap}): The combinations of ACFs' values that can be varied to be within the adjustable range. n_{ap} denotes the number of ACFs' assignable points.

Robustness index (R_A): Index to evaluate the robustness of the objective characteristics with regards to the ACFs adjustment.

This research applies the Vector evaluated particle swarm optimization in order to maximize R_A and minimize the adjustable range, and aims to improving the above issue.

2 Type and comparison of Multi-objective optimization

This research selected multi-objective optimization method which simultaneous optimize maximizing R_A and minimizing the adjustable range. Multi-objective optimization method derives the pareto optimal solution in the minimization problem which is minimized multi-

Table 1 Characteristic of Multi-Objective Optimization

Characteristic of Multi-Objective Optimization \ Multi-Objective Optimization	Weighted Aggregation Method	ε-constrained Method	Surrogate Worth Trade-off Method	Objective Function Ordering Method	Weighted Tchebycheff Norm Method	Weighted lp Norm Method	Goal Program Method	Step Method	Satisficing Trade-off Method	Compromise Programming Method	Tchebycheff Method	GUESS Method	NIMBUS Method
Using aspiration level	×	×	×	×	×	○	○	×	○	×	×	○	○
Using constraint value	×	○	○	×	×	×	×	×	×	×	×	×	○
Using ability and Knowledge of a designer	×	×	○	×	×	×	○	○	○	×	○	○	○
Using quantitative weight	○	×	×	×	○	○	○	×	○	○	○	×	○
Using ideal point	×	×	×	×	×	×	×	○	○	○	○	○	○
Using worst point	×	×	×	×	×	×	×	×	○	×	×	○	×

objective function which has trade off relationship, where he pareto optimal solution is the feasible solution that cannot minimize all objective function simultaneously.

Multi-objective optimization methods are divided into two types: scalar methods and evolutionary multi-objective optimization methods.

Scalar methods singulate the objective functions and derive an optimal solution from one the pareto optimal set. The characteristics of the scalar approaches are summarized in **Table 1**. Some of the scalar methods are described as follows: 1) weighted aggregation (all objective functions are summed up to a single scalar with a prescribed weight); 2) ε-constrained (objective functions are transformed into constraints exception of the one emphasized by user); 3) Goal Program (minimize the distance between the value of each objective function and the aspiration level). These methods can derive an accurate solution of the monomodal function using the mathematical techniques (e.g. steepest descent method, newton method, etc) based on the gradient of the objective function. Additionally, they can also derive a global optimal solution of the multimodal function using the metaheuristics such as GA, etc. However, there are certain drawbacks: 1) the parameters (e.g. the weight coefficient, the aspiration level, etc) are required to be set; 2) only one pareto optimal solution can be derived by once optimization trial.

Evolutionary multi-objective optimization methods are proposed on the basis of applied the metaheuristics (e.g. GA, particle swarm optimization (PSO), etc). These methods can derive many pareto optimal solutions using only a few parameters (without weighted coefficient, the aspiration level, etc) However, there are certain drawbacks compared to the scalar methods: 1) accuracy of the derived pareto optimal solution is less; 2) it has larger amount of the calculation. In the robust design problem of this study (eq. (1)), it is difficult to set the proper parameters of R_A and the adjustable range, such as the weighted coefficient and the aspiration level. Hence, this study applied evolutionary multi-objective optimization to the proposed method, and focused the PSO. It is described in the following reason: the design problem of eq. (1) seems to include both of the continuous values (e.g. the adjustment amount of a reclining seat angle) and the discrete values (e.g. the variation of the manufacture). The discrete values can be described as the combination of the continuous values. Therefore, this study used evolutionary multi-objective optimization method based on PSO which can derive a global optimal solution on continuous function. The characteristics of each multi-objective optimization based on PSO are shown in **Table 2**. In this table, Vector Evaluated Particle Swarm Optimization (VEPSO) has low computational complexity compared to the other. This is beneficial to the design problem (eq. (1)) because the computational complexity of R_A is high.

Table 2 Characteristic of Multi-Objective Optimization

Characteristic of Multi-ObjectiveOptimization \ Multi-Objective Optimization	VEPSO	NSPSO	SigmaPSO	AMOPSO	OMOPSO
Using Non-dominated Sorting	×	○	×	×	×
Using Crowding Distance	×	○	×	×	○
Using Sigma Value	×	×	○	×	×
Using Mutation	×	○	○	×	○
Using Clustering	×	×	×	○	×
Complexity in terms of the number of evaluations	$O(mN)$	$O(mN^2)$	$O(mN^2)$	$O(m^2N^2)$	$O(mN^2)$

This study is proposed an optimization algorithm using the VEPSO in order to solve the design problem.

3 Outline of VEPSO

The VEPSO [5, 6] is an improved method of the PSO [7] that is one of the representative metaheuristics, in order to handle the multi objective optimization problems. The PSO imitates the movement of organisms in a bird flock or fish school and searches a solution using the information both from the individuals (particles) and their swarm. The VEPSO assigns an objective to each of swarms and searches a solution using the information inside or between swarms. The location vector (i.e. design variables) of the *i* th particle in the *j* th swarm $\boldsymbol{x}_i^{[j]}$ is updated as follows:

$$\boldsymbol{x}_i^{[j]}(T+1) = \boldsymbol{x}_i^{[j]}(T) + \boldsymbol{v}_i^{[j]}(T) \quad , \quad (4)$$

where, T is the number of iterations. $\boldsymbol{v}$ is the velocity vector to direct the particles to the updated locations and is calculated as:

$$\boldsymbol{v}_i^{[j]}(T+1) = k[\, w\boldsymbol{v}_i^{[j]}(T) + c_1 r_1 \{\boldsymbol{x}_{\mathrm{pb},i}^{[j]}(T) - \boldsymbol{x}_i^{[j]}(T)\} + c_2 r_2 \{\boldsymbol{x}_{\mathrm{gb},i}^{[p]}(T) - \boldsymbol{x}_i^{[j]}(T)\}\,]$$
$$\left(p = \begin{cases} M & \text{if } j = 1, \\ j-1 & \text{if } j = 2,3,\ldots,M \end{cases} \right) \quad , (5)$$

where, M is the number of swarms; c_1 and c_2 are the parameters to express the degree of incidence of the private best location of each particle $\boldsymbol{x}_{\mathrm{pb}}$ and the global best location $\boldsymbol{x}_{\mathrm{gb}}$, respectively; r_1 and r_2 denote the random numbers uniformly distributed in [0, 1]. w is the parameter to define the effect of the current velocity vector and decreases based on T as shown in the following equation:

$$w(T) = w_{\max} - \frac{w_{\max} - w_{\min}}{T_{\max}} \cdot T \quad , \quad (6)$$

where, $w_{\max}$ and $w_{\min}$ are the maximum and minimum value of w, respectively. $T_{\max}$ is the maximum number of the iterations. k denotes the parameter relating the convergence performance and expressed as the following equation:

$$k = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \quad (\varphi = c_1 + c_2) \quad . \quad (7)$$

As shown in eq. (5), the velocity vectors are defined using the global best locations of the different swarms. This enables the solution search based on the information from the other swarms and the global locations (solutions) of the swarms to approach each other.

4 Proposal robust design method using VEPSO

This study assigned each swarms to the robustness index and ACF range that is the Euclidean distance between assignable points.

The optimization algorithm using the VEPSO is described in **Fig.3**. In this algorithm, the parameters of the VEPSO (e.g. c, w, $T_{\max}$, etc) are firstly set. Next, the number of the assignable points is decided and the same number of the swarms is set. The locations of the particles are updated based on the objective (robustness) R_{A} and ACFs range. The derived pareto optimal solutions is memorized in an external archive [8]. The update of the locations iterates until $T = T_{\max}$, and the solution candidates in an external archive are derived and an optimal design solution (adjustable range is selected by the designers).

5 Illustrative Example

5.1 Problem Description

Similar to previous studies [2, 4], the design objective in this design example is to reduce the hip-sliding force for a public seat. The ACFs are defined as seat the cushion angle θ_{C}, seat back angle θ_{B} and forward tilt angle θ_{F}.

Additionally, in the PSO, the parameters (e.g. c, w, etc) are important for the convergence and the computational efficiency. Specifically, c denotes weight of exchanging information between swarms, and affect the search of the pareto optimal solutions. Therefore, this study focused on c_1 and c_2. To clarify the proper values of them, this study implemented the analyses regarding several values of the parameter combinations similar to the conventional studies by Carlisle [9]. Carisle varied

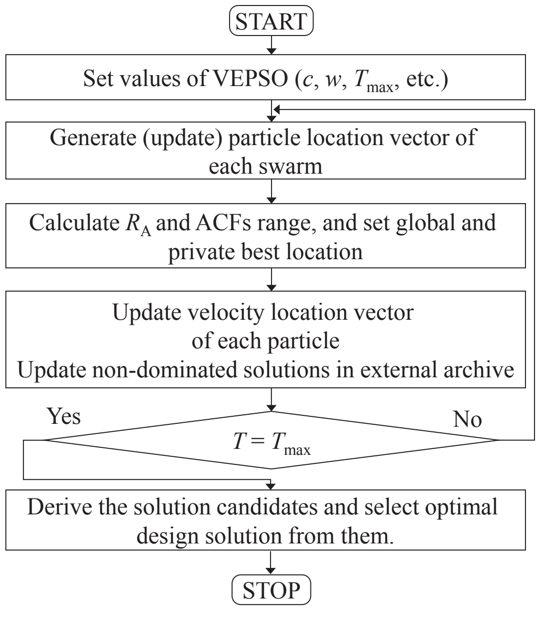


Fig. 3 Proposed algorithm of robust design method

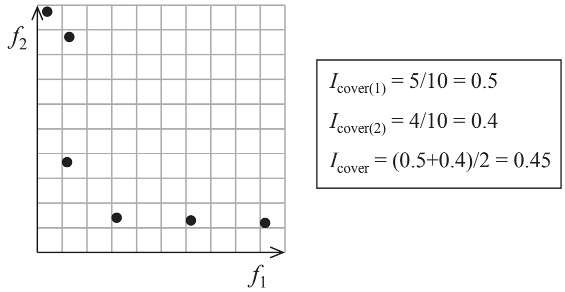


Fig. 4 Cover rate

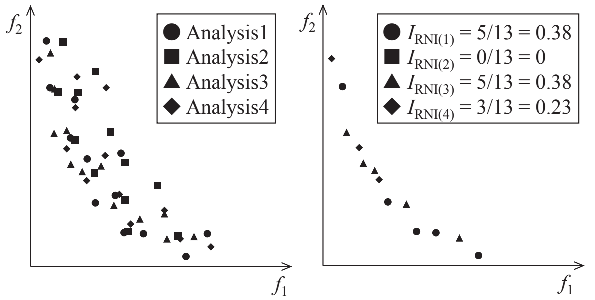
Table 3 Definition of parameters

Items	Set value	
	GA	Proposed method (Analyses)
Tolerance of y	$-10 \leq y \leq 20$	
Feasible area of t_1	$10 \leq \theta_C \leq 25$	
Feasible area of t_2	$20 \leq \theta_B \leq 35$, $\theta_B \geq \theta_C+10$	
Feasible area of t_3	$0 \leq \theta_F \leq 30$	
Max iteration number T_{max}	10000	100
c_1	—	(0.00 - 4.10, 4.10 - 0.00)
c_2	—	
w_{min}	—	0.4
w_{max}	—	0.9
Number of objective	1	2
Swarm size	—	30

the c_1 from 0.0 to 4.1 (i.e., $c_2 = 4.1-c_1$) and ran a test function for 20 times in each parameter combinations. Then, he evaluated the combinations using the median number of the function call until an optimal solution is derived. The derived pareto optimal solutions are expected to be diversity and accuracy. Therefore, this study applied the median value of cover rate [10] and Ratio of Non-dominated Individuals (RNI) [11], which can be compared the accuracy and the diversity of the pareto optimal solutions. The cover rate is the average value of the number of a region including the pareto optimal solutions divided by the number of all regions, where regions are generated dividing the objective space. Therefore, higher cover rate value means more diverse pareto optimal solutions (**Fig. 4**). RNI is the ratio of the number of the non-dominated solutions derived in each analysis divided by that of the entire set of the non-dominated solutions derived in all analyses. Non-dominated solution is described as the pareto optimal solution which dominates all others. Therefore, higher value means more similar to the true pareto optimal solutions (**Fig. 5**). The definition of the parameters in the analysis is summarized in **Table 3**.

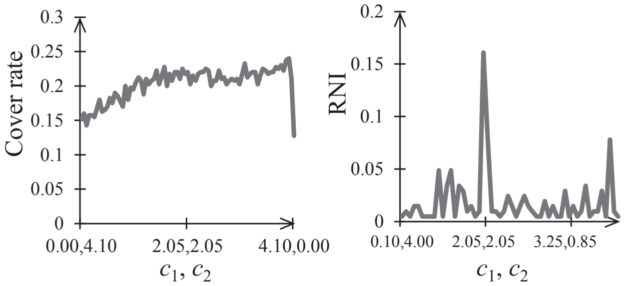
5.2 Result and discussion

The analysis of this study derived the pareto optimal solutions for each parameter combination, and these Cover Rate and the RNI are shown in **Fig. 6**. **Figure 6 (a)** shows the cover rate is minimum at c_1=4.10, c_2=0.00. This means the smaller c_2 causes the search considering each individual objective function (not considering both function). Additionally, **Figure 6 (b)** shows the RNI is maximum at $c_1 = 2.05$, $c_2 = 2.05$. This seems to be caused by the two reasons: the smaller c_2 caused the issue described above and the larger c_2 caused the local search. The latter is also pointed out by Carlisle [23]. Hence, the combination of the parameter values ($c_1 = 2.05$, $c_2 = 2.05$) is suitable in this design problem. The comparison of the pareto optimal solution (c_1=2.05, c_2=2.05) with the GA (conventional method) is shown in **Fig. 7**. This figure shows the proposed method derives the optimal design solution with higher robustness and smaller adjustable range compared to those by the conventional method.



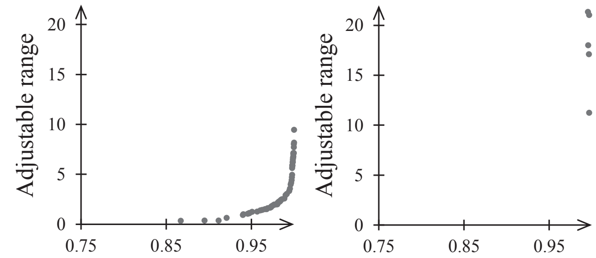
(a) Sum set of each analyses (b) Non dominated solutions of each analyses

Fig. 5 RNI



(a) Cover rate (b) RNI

Fig. 6 Evaluation of cover rate and RNI based on each c_1(c_2=4.1−c_1) parameters



(a) VEPSO(c_1 = 2.05, c_2 = 2.05) (b) GA

Fig. 7 Solution candidates of conventional and proposed method

6 Conclusions

The research proposed the robust multiple optimization method which optimize R_A and the adjustable range in order to effectively derive the optimal adjustable ranges for adjustable mechanisms, focused evolutionary multi-objective optimization methods, and selected VEPSO. It is described in the following reason: those methods can derive many pareto optimal solutions using only a few parameters (without weighted coefficient, the aspiration level, etc.). Additionally, this. The proposed method was applied to seat design problem. In this application, it was confirmed that the proposed method can derive the design optimal solution with high robustness, small adjustable range and the appropriate parameters.

Acknowledgement

This work was supported by the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research.

References

[1] Matsuoka, Y., Design Science. Maruzen, Tokyo, (2010).

[2] Watai, A., Nakatsuka, S., Kato, T., Ujiie, Y. and Matsuoka, Y., “Robust Design Method for Diverse Conditions”, Proceedings of the ASME DETC (DETC/CIE2009-87108), (2009), pp. 1131-1138.

[3] Kimura, S., Kato, T., and Matsuoka, Y., A proposed “Robust Design Method for Deriving Adjustable Range”, Advanced Materials Research, Vol. 311-313, (2011), pp. 2332-2335.

[4] Kato, T., Watai, A., Ujiie, Y., Matsuoka, Y., Robust Design Method for Objective Characteristics with Nonnormal Distributions and Control Factors with Adjustable Range, UMTIK 2010 International Conference on Machine Design and Production, (2010).

[5] Vlachogiannis, J.G. and Lee, K.Y., “Determining Generator Contributions to Transmission System Using Parallel Vector Evaluated Particle Swarm Optimization”, IEEE Transactions on Power Systems, Vol. 20, No. 4, (2005), pp. 1765-1774.

[6] Vlachogiannis, J.G. and Lee, K.Y., “Multi-objective based on parallel vector evaluated particle swarm optimization for optimal steady-state performance of power systems”, Expert Systems with Applications, Vol. 36, No. 8, (2009), pp. 10802-10808.

[7] Kennedy, J. and Eberhart, R.C., “Particle swarm optimization”, Proceedings of IEEE International Conference on Neural Networks, Vol. 4, (1995), pp. 1942-1948.

[8] Yaochu, J., Markus, Olhofer.: Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization, Proceedings of the Genetic and Evolutionary Computation Conference, (2001), 1042.

[9] Carlisle, A. and Doizier, G., “An off- the-shelf PSO”, Indianapolis, Proceedings of Workshop Particle Swarm Optimization, (2001).

[10] Hiyane, K., “Generation of a Set of Pareto-Optimal Solutions for Multiobjective Optimization by Parallel Genetic Algorithms and its Quantitative Evaluation”, SICE Symposium on Decentralized Autonomous Systems, (1997), pp. 295-300.

[11] Tan, K.C., Lee, T.H., and Khor,E.F.. “Increamenting Multi-objective Evolutionary Algorithms: Performance Studies and Comparisons. In First International Conference on Evolutionary Multi-Criterion Optimization”, (2001), pp. 111-125.

Received on December 31, 2013

Accepted on February 7, 2014