Evolutionary and Deterministic Methods for Design,
Optimization and Control with Applications to Industrial and Societal Problems
EUROGEN 2005
R. Schilling, W.Haase, J. Periaux, H. Baier, G. Bugeda (Eds)


# NEW CONSTRAINT-HANDLING METHOD FOR MULTI-OBJECTIVE MULTI-CONSTRAINT EVOLUTIONARY OPTIMIZATION AND ITS APPLICATION TO SPACE PLANE DESIGN

**Akira Oyama[*], Koji Shimoyama[†] and Kozo Fujii[*]**

[*] Department of Space Transportation Engineering
Institute of Space and Astronautical Science
Japan Aerospace Exploration Agency
3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510 Japan
Email: {oyama, fujii}@flab.eng.isas.jaxa.jp – Web page: http://flab.eng.isas.jaxa.jp

[†]Department of Aeronautics and Astronautics
University of Tokyo
3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510 Japan
Email: simoyama@flab.eng.isas.jaxa.jp

**Key words:** Constraint Handling, Space Plane.

**Abstract.** *A new constraint-handling method based on Pareto-optimality concept for multi-objective multi-constraint design optimization problems has been proposed. The proposed method does not need any constants to be tuned for constraint handling. In addition, the present method does not use weighted-sum of constraints and thus does not need tuning of weight coefficients and is efficient even when the amount of violation of each constraint is significantly different. The proposed approach is demonstrated to be remarkably robust than the dynamic penalty approach and other dominance-based approaches through the optimal design of a welded beam and conceptual design optimization of a two-stage-to-orbit space plane.*

## 1 PROBLEM STATEMENT

Without losing generality, constrained real-number optimization problems are written as:

$$\text{Find } \vec{x} \text{ that minimize } \vec{f}(\vec{x}) = (f_1(\vec{x}),..., f_m(\vec{x}),..., f_{m_{\max}}(\vec{x})) \tag{1}$$

subject to

$$\vec{g}(\vec{x}) = (g_1(\vec{x}),..., g_n(\vec{x}),..., g_{n_{\max}}(\vec{x})) \leq 0 \tag{2}$$

where $\vec{x} = (x_1,..., x_l,..., x_{l_{\max}})$ is the vector of solution that minimizes objective function(s) $\vec{f}(\vec{x})$ while satisfying the constraint(s) $\vec{g}(\vec{x}) \leq 0$. $l_{max}$, $m_{max}$ and $n_{max}$ are numbers of design parameter(s), objective function(s) and constraint(s), respectively.

## 2 INTRODUCTION

Evolutionary algorithms (EAs, see [1] for example) are robust and efficient design optimization algorithms based on the Theory of Evolution proposed by Charles Darwin, where a biological population evolves over generations to adapt to an environment by selection, recombination and mutation. One of the key features of EAs is that they search from multiple points in the design space, instead of moving from a single point like gradient-based methods do. Furthermore, these methods work on function evaluations alone (fitness) and do not require derivatives or gradients of the objective functions. These features lead to advantages over deterministic optimization approaches such as robustness, capability to uniformly capture Pareto-optimal solutions, suitability to parallel computing, and simplicity in coupling the EA code and evaluation codes. As a result, EAs have been applied to many real-world design problems in various fields (for example, see [2-3]).

However, EAs do not have any explicit mechanism to handle constraints while most of real-world design optimization problems have multiple constraints. A considerable amount of researches on constraint handing techniques that incorporate objective function(s) and constraint(s) into the fitness function of design candidates has been carried out (a good summary is given in [4]).

Traditional approach for handling design constraints of single-objective design optimization problems for evolutionary optimization is the penalty function method [1] where fitness of a design candidate is determined based on a scale function *F*, which is weighted sum of the objective function value and the amount of design constraint violations as:

$$F(\vec{x}) = f_1(\vec{x}) + \sum_{n=1}^{n_{\max}} \alpha_n \cdot \max(g_n(\vec{x}), 0) \tag{3}$$

where $\alpha_n$ are the positive penalty function coefficients. However, this method requires a careful tuning of the penalty function coefficients to obtain a satisfactory design. For example, if the penalty function coefficients are too small, the optimized design would not satisfy the constraints. On the other hand, if the penalty function coefficients are too large, the optimized design would not have satisfactory objective function value. In addition to the balance between the objective function and the constraints, the balance among the constraints is also to be carefully tuned so that the optimized design satisfies all of the constraints. Moreover, the penalty function method does not intend to deal with multiobjective design optimization problems. Application of this method to a multiobjective optimization problem raises another problem; how to combine multiple constraints with multiple objectives.

Thus a considerable number of constraint handing techniques based on multi-objective evolutionary algorithm concepts have been proposed to treat constrained design optimization problems without fine tuning of the penalty function coefficients. A simple approach is to consider constraints as additional objectives and apply any multiobjective evolutionary algorithm [1]. This approach does not need formulation of the scale function $F$ and fine tuning of the penalty function coefficients. However, This approach is not efficient when number of constraints is large because the optimized solutions scatter in the $m_{max}+n_{max}$ dimension objective function space.

Deb [5] proposed an attracting approach for constraint-handling which bases on the non-dominance concept. The constrained domination approach ranks design candidates using the following definition of domination between two design candidates:

**Definition 1:** A solution *i* is said to constrained-dominate a solution *j*, if any of the following conditions is true:
1. Solutions *i* and *j* are feasible and solution *i* dominates solution *j*.
2. Solution *i* is feasible and solution *j* is not.
3. Solutions *i* and *j* are both infeasible, but solution *i* has a smaller constraint violation.

where

**Definition 2:** A solution *i* is said to dominate a solution *j*, if both of the following conditions are true:
1. Solutions *i* is no worse than solution *j* in all objectives, *i.e.*,

$$\forall f_m(\vec{x}_i) \le f_m(\vec{x}_j) \quad \mathbf{(4)}$$

2. Solution *i* is strictly better than solution *j* in at least one objective, *i.e.*,

$$\exists f_m(\vec{x}_i) < f_m(\vec{x}_j) \quad \mathbf{(5)}$$

This approach does not need tuning of the penalty function coefficients as long as the number of constraint is one. In this sense, this approach is very useful for EA-based design optimizations. However, this approach still requires careful tuning of the weight coefficients of the constraints when multiple constraints are considered.

Another interesting approach is proposed by Coello [6]. Essence of the method is usage of the following definition of domination:

**Definition 3:** A solution *i* is said to constrained-dominate a solution *j*, if any of the following conditions is true:

1. Solutions *i* and *j* are feasible and solution *i* dominates solution *j*.
2. Solution *i* is feasible and solution *j* is not.
3. Solutions *i* and *j* are both infeasible and solution *i* violates less number of constraints than solution *j*.
4. Solutions *i* and *j* are both infeasible and solutions *i* and *j* violates the same number of constraints, but solution *i* has a total amount of constraint violation smaller than the constraint violation of solution *j* where the total amount of constraint violation for an individual $\vec{x}$ is given by

$$coef(\vec{x}) = \sum_{n=1}^{n_{\max}} \max(g_n(\vec{x}), 0) \quad \textbf{(6)}$$

Advantage of this method is that it does not use any coefficient to be tuned even if multiple constraints are considered. However, this constraint-handling technique may not be very efficient when the degrees of violation of constraints $g_n(\vec{x})$ are significantly different because the total amount of constraint violation of an individual is simple sum of $g_n(\vec{x})$ as in eq. (6).

More recently, Coello and Mezura [7] proposed a dominance-based tournament selection based on the niched-pareto genetic algorithm for single-objective constrained design optimization problems where

1. If solutions *i* and *j* are both feasible and solution *i* has better fitness value, solution *i* wins.
2. If solution *i* is feasible and solution *j* is not, solution *i* wins.
3. If solutions *i* and *j* are both infeasible and if solution *i* is a nondominated solution and solution *j* is not, solution *i* wins.
4. If solutions *i* and *j* are both infeasible and if solutions *i* and *j* are both dominated or nondominated, the solution with the lowest amount of constraint violation wins.

where dominance is defined in the space where constraints are handled as additional objectives. This approach needs tuning of only one parameter $S_r$ that controls the diversity of the population. This approach was demonstrated to be more efficient and robust than other multiobjective-evolutionary-algorithm-based constraint handling techniques [8]. However, this approach may not be very efficient when the degrees of violation of constraints $g_n(\vec{x})$ are significantly different because it also uses the total amount of constraint violation for comparison. In addition, application of the proposed method is limited to single-objective design optimization problems.

Therefore, objective of the present study is to propose a new constraint-handling method for multi-objective multi-constraint design optimization problems. The present approach does not need tuning of any coefficient and is efficient even when the amount of violation of each constraint is significantly different.

This paper is organized as follows. First, the proposed constraint-handling technique is presented in Section 3. Then, the present evolutionary algorithm and some constraint-handling techniques that are compared with the proposed approach are described in Section 4. The optimal design of a welded beam (Section 5) and conceptual design optimization of a two-stage-to-orbit space plane (Section 6) are demonstrated to compare the proposed method with the other constraint-handling methods. Finally, Section 7 summarizes the present work.

## 3 PROPOSED CONSTRAINT-HANDLING METHOD

The proposed constraint-handling method bases on the following non-dominance concept:

**Definition 4:** A solution *i* is said to constrained-dominate a solution *j*, if any of the following conditions is true:

1. Solutions *i* and *j* are both feasible and solution *i* dominates solution *j* in objective function space.
2. Solution *i* is feasible and solution *j* is not.
3. Solutions *i* and *j* are both infeasible, but solution *i* dominates solution *j* in constraint space.

where dominance in objective function space is defined as **Definition 2** while dominance in constraint space is defined as:

**Definition 5:** A solution *i* is said to dominate a solution *j* in constraint space, if both of the following conditions are true:

1. Solutions *i* is no worse than solution *j* in all constraints, *i.e.*,

$$\forall G_n(\vec{x}_i) \le G_n(\vec{x}_j) \tag{7}$$

2. Solution *i* is strictly better than solution *j* in at least one constraint, *i.e.*,

$$\exists G_n(\vec{x}_i) < G_n(\vec{x}_j) \tag{8}$$

where

$$G_n(\vec{x}) = \max(0, g_n(\vec{x})) \tag{9}$$

The proposed method simply introduces the idea of non-dominance concept in the objective function space to the constraint function space. This idea can be used for most of EAs. For example, any ranking procedure can be used for ranking among feasible designs as well as infeasible designs. Use of stochastic ranking [9] may further improve efficiency and robustness. In addition, robustness is further improved by applying any sharing mechanism in the constraint space as well as the objective space for severely constrained design optimization problems.

The proposed method has a number of advantages:

1. Application to multi-objective multi-constraint design optimization problems is straightforward.
2. It is efficient and robust even when the degree of violation of each constraint is very different because total amount of constraint violation is not used.
3. It does not need any coefficient to be tuned.
4. Number of objectives is not increased since non-dominance ranking is applied to feasible designs and infeasible designs separately.
5. Implementation is easy

## 4 PRESENT EVOLUTIONARY ALGORITHM AND THE CONSTRAINT-HANDLING TECHNIQUES

The present evolutionary algorithm and the constraint-handling techniques to be compared with the proposed constraint-handling technique are presented.

### 4.1 Present evolutionary algorithm

The present EA uses the floating-point representation [10] to represent design parameters of design candidates where an individual is characterized by a vector of real numbers. Random parental selection and the best-*N* selection [11] where the best *N* individuals are selected for the next generation among *N* parents and *N* children based on Pareto-optimality are used. The blended crossover (BLX-0.5) [12] is used for reproduction. Since the strong elitism is used, high mutation rate of 0.2 is applied and a random disturbance is added to the parameter in the amount up to ±20% of the design space. The initial population is generated randomly over the entire design space.

### 4.2 Dynamic penalty method

In the dynamic penalty method proposed by Joines and Houck [13], fitness of each solution is defined by the following function:

$$F(\vec{x}) = f_1(\vec{x}) + (C \cdot t)^{\alpha} \cdot \left( \sum_{n=1}^{n_{\max}} G_n^{\beta}(\vec{x}) \right) \quad \textbf{(10)}$$

where $t$ is generation and $C$, $\alpha$ and $\beta$ are constants defined by the user (the authors used $C = 0.5$, $\alpha = 2$ and $\beta = 2$). This dynamic function approach is considered to be efficient in the sense that number of coefficients to be tuned is small as well as the penalty function coefficient changes through generations to increase the penalty as the optimization progresses.

### 4.3 Constraint-handling method by Deb

Rank of each design candidate is defined according to **Definition 1**.To handle multiple constraints, the constraints are combined into one constraint violation function as in eq. (6).

### 4.4 Proposed constraint-handling method

Rank of each design candidate is defined according to **Definition 4**. Fonseca and Fleming's Pareto-based ranking [14] is used to rank infeasible designs. To maintain diversity in population during optimization, a standard fitness sharing [14] is applied to the infeasible designs based on their constraint violations.

## 5 EXPERIMENTAL RESULT 1: OPTIMAL DESIGN OF A WELDED BEAM

In this section, the optimal design of a welded beam [15] is demonstrated to compare the present method with the non-dominance-concept-based constraint-handling methods proposed by Deb and Coello and the dynamic penalty function approach.

### 5.1 Formulation of design optimization problem

Structure of the welded beam is shown in Fig. 1. The welded beam consists of a beam and a weld required to secure the beam to the member. The objective of the design is to find a feasible set of dimensions *h, l, t*, to and *b* (denoted by ($x_1,x_2,x_3,x_4$)) to carry a certain load (*P*) and still have a minimum total fabricating cost.

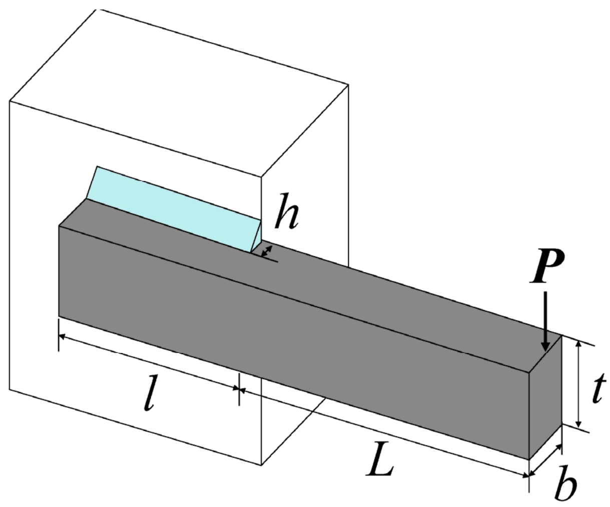


Figure 1: The welded beam structure.

The objective function, $f_1(\vec{x})$, is the total fabricating cost that mainly comprises of the set-up cost, welding labor cost, and material cost:

$$f_1(\vec{x}) = (1 + c_1)x_1^2 x_2 + c_2 x_3 x_4 (L + x_2) \tag{11}$$

where $c_1$ and $c_2$ are the cost of unit volume of weld material and bar stock, respectively. The associated functional constraints are:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0 \tag{12}$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \le 0 \tag{13}$$

$$g_3(\vec{x}) = x_1 - x_4 \le 0 \tag{14}$$

$$g_4(\vec{x}) = c_1 x_1{}^2 + c_2 x_3 x_4 (L + x_2) - 5 \le 0 \tag{15}$$

$$g_5(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \le 0 \tag{16}$$

$$g_6(\vec{x}) = P - P_c(\vec{x}) \le 0 \tag{17}$$

where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \tag{18}$$

$$\tau' = \frac{P}{\sqrt{2}x_1 x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P(L + \frac{x_2}{2}) \tag{19}$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \tag{20}$$

$$J = 2\left\{\sqrt{2}x_1 x_2 \left[\frac{x_x^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \tag{21}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}, \quad \delta(\vec{x}) = \frac{4PL^3}{E x_3^3 x_4} \tag{22}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{x_3^2 x_4^6 / 36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{23}$$

$$c_1=0.10471,\ c_2=0.04811,\ P=6{,}000lb$$
$$L=14in,\ E=30{,}000{,}000psi,\ G=12{,}000{,}000psi \tag{24}$$
$$\delta_{\max}=0.25in,\ \tau_{\max}=5{,}000psi,\ \sigma_{\max}=10{,}000psi$$

where $\sigma$, $\tau$, $P$ and $\delta$ are bar bending stress, shear stress of weld, bar buckling load and bar end deflection, respectively. The maximum bending stress $\sigma_{\max}$ and the maximum shear stress $\tau_{\max}$ of the present design problem are set to smaller than those of the original one to give severer constraints. The search space is

$$0.125 \le x_1 \le 5,\quad 0.1 \le x_2 \le 10,\quad 0.1 \le x_3 \le 10,\quad 0.1 \le x_4 \le 5 \tag{25}$$

### 5.2 Results

Population size and number of generations are set to 100 and 200, respectively. Fifty trials starting from different initial populations are demonstrated to statistically compare the constraint-handling methods. Since severe constraints are intentionally imposed on the present optimization problems, the evolutionary algorithm sometimes failed to find feasible designs. Figure 2 compares number of trials in which feasible designs are found. It is remarkable that the present constraint-handling technique found feasible designs 48 times among 50 trials.

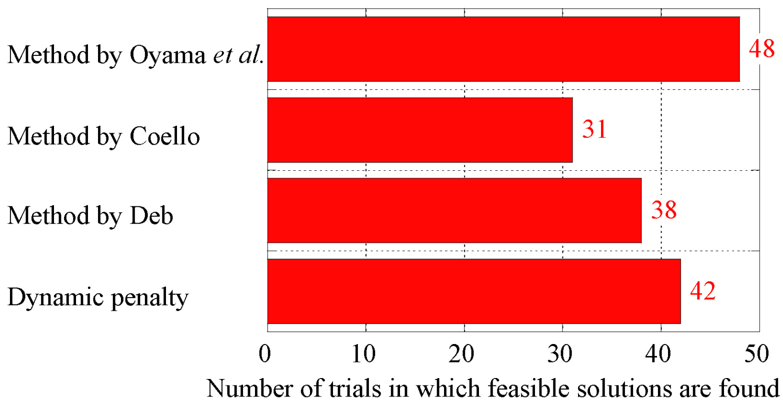


Figure 2: Number of trials in which feasible solutions are found.

Figure 3 compares average cost of the optimized feasible designs to be minimized. The average cost of the designs optimized by the evolutionary algorithm coupled with the proposed method is also the smallest among the compared constraint-handling techniques.

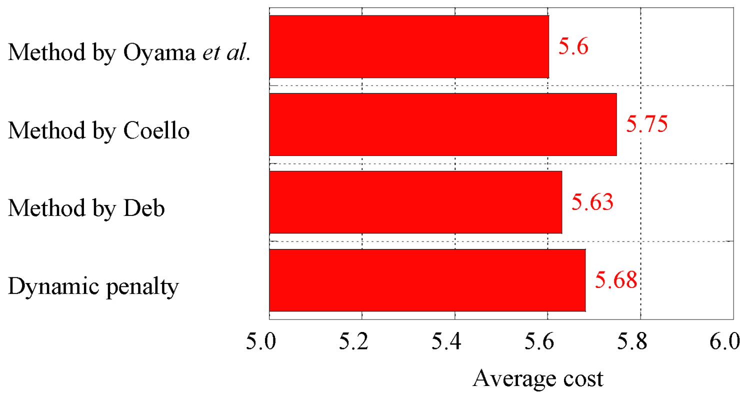


Figure 3: Average cost of optimized designs.

## 6 EXPERIMENTAL RESULT 2: CONCEPTUAL DESIGN OPTIMIZATION OF A TWO-STAGE-TO-ORBIT SPACEPLANE

In this section, conceptual design optimization of a two-stage-to-orbit (TSTO) spaceplane (Fig. 4) is demonstrated to ensure feasibility of the present approach to real-world design optimization problems. The TSTO spaceplane considered here consists of a booster with air-breathing engines and an orbiter with rocket engines. The orbiter is separated from the booster at a certain altitude to reach the low earth orbit (LEO) to bring the payload.

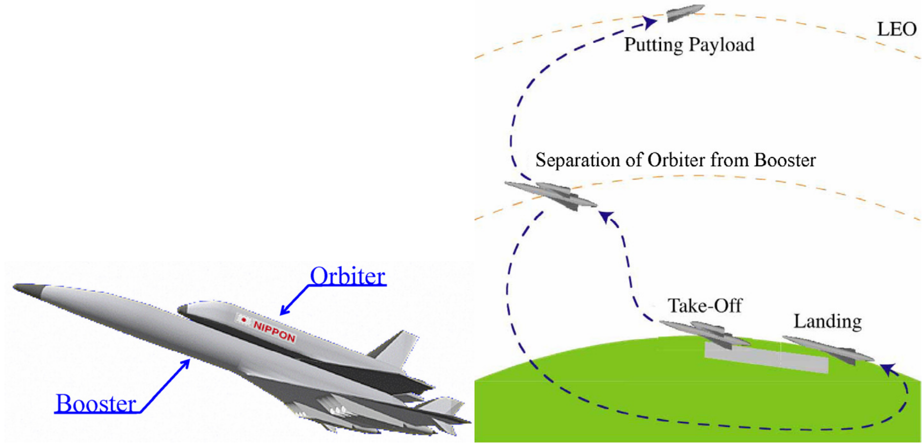


Figure 4: The TSTO Spaceplane and its mission.

### 6.1 Formulation of the present design optimization problem

The present TSTO mission is to put a payload of 10t into the equatorial orbit at the altitude of 400km. For simplicity, the take-off and landing sites are assumed to be on the equator. The engine of the Booster is assumed to be the air-turbo-ramjet engine with expander cycle (ATREX) [16], which is under development in Japan. The objective is to minimize gross take-off weight of the spaceplane. The separation time is constrained to be smaller than

550 [sec]. The maximum thrust of the booster is also constrained to be smaller than 2.5 [MN]. The gross take-off weight, separation time and maximum thrust of the booster are iteratively computed from the propulsion, aerodynamics, trajectory and structure modules [17,18] as shown in Fig. 5. Here, propulsion, trajectory and airframe configuration parameters (total ten) are considered as design variables.

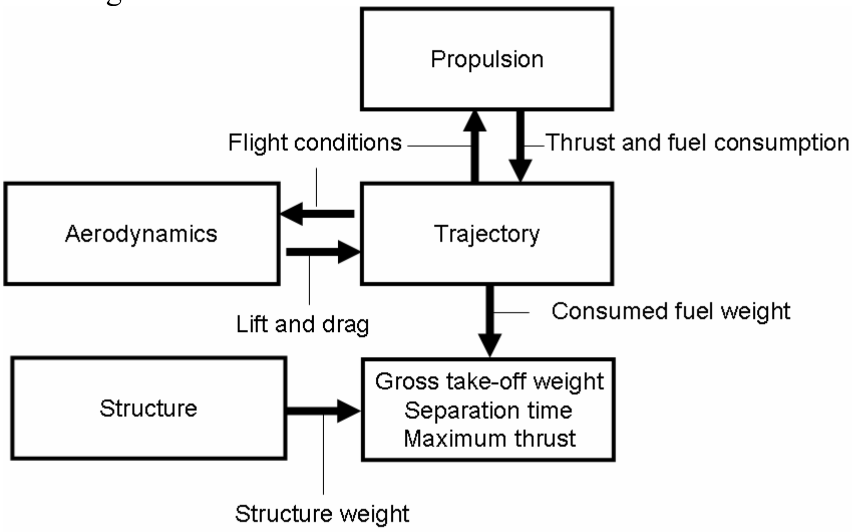


Figure 5: The TSTO simulation system

### 6.2 Results

Population size and number of generations are set to fifty. One hundred trials with different initial populations are run for each constraint-handling technique. Table 1 presents number of trials in which feasible designs are found, the average weight of the optimized designs, weight of the best optimized design and standard deviation. The dynamic penalty method and dominance-based approach by Deb failed to find feasible designs. The reason is probably that both methods adopt linear-sum of the amount of constraint violation of different order of magnitude. On the other hand, the present method and the method by Coello get good scores while the proposed method is slightly better than the method by Coello in every measure.

| | Number of Successes | Average weight [megaton] | Weight of the best design [megaton] | Standard deviation |
|---|---|---|---|---|
| Proposed method | 100 | 0.371190 | 0.369000 | 1578.7 |
| Method by Coello | 99 | 0.371285 | 0.369038 | 1623.9 |
| Method by Deb | | No feasible design is found | | |
| Dynamic Penalty | | No feasible design is found | | |

Table 1 : Comparison between the constraint-handling methods

## 7 SUMMARY

A new constraint-handling method based on Pareto-optimality concept for multi-objective multi-constraint design optimization problems has been proposed. The advantages of the proposed approach are

1. Application to multi-objective multi-constraint design optimization problems is straightforward.
2. It is efficient and robust even when the degree of violation of each constraint is very different because total amount of constraint violation is not used.
3. It does not need any coefficient to be tuned.
4. Number of objectives is not increased since non-dominance ranking is applied to feasible designs and infeasible designs separately.
5. Implementation is easy

The proposed approach was demonstrated to be more robust than the dynamic penalty approach and previous dominance-based approaches through the optimal design of a welded beam and conceptual design optimization of a two-stage-to-orbit space plane.

Although single-objective design optimizations were demonstrated in this work, application of the present method to multi-objective multi-constraint design optimization problem is straightforward.

## REFERENCES

[1] Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Chickester, England, 2001.

[2] Zitzler, E., Deb, K., Thiele, L., Coello, C. A. C., Corne, D. (Eds.): Evolutionary Multi-Criterion Optimization. Sprinter-Verlag, Berlin Heidelberg New York,2001, pp.486-710.

[3] Fonseca, C. M., Fleming, P. J., Zitzler, E, Deb, K., Thiele, L. (Eds.): Evolutionary Multi-Criterion Optimization. Sprinter-Verlag, Berlin Heidelberg New York, 2003, pp. 550-810.

[4] Coello, C. A. C.: Theoretical and Numerical Constraint-handling Techniques Used with Evolutionary Algorithms: A Survey of the State of the Art. Computer Methods in Applied Mechanics and Engineering, Vol. 191, 2002, pp.1245-1287.

[5] Deb, K.: An Efficient Constraint-handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering, Vol. 186(2-4) , 2000, pp.311-338.

[6] Coello, C. A. C.: Constraint-Handling Using an Evolutionary Multiobjective Optimization Technique. Civil Engineering and Environmental Systems Vol. 17, 2000, pp.319-346.

[7] Coello, C. A. C. and Mezura-Montes, E.: Constraint-Handling in Genetic Algorithms Through the Use of Dominance-Based Tournament Selection. Advanced Engineering Informatics, Vol.16, 2002, pp.193-203.

[8] Mezura-Montes, E. and Coello, C. A. C.: A Numerical Comparison of some Multiobjective-Based Techniques to Handle Constraints in Genetic Algorithms. Technical Report EVOCINV-03-2002, 2002.

[9] Runarsson, T. P. and Yao, X.: Stochastic Ranking for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation, Vol. 4, No. 3, 2000, pp. 284-294.
[10] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, Third, Revised and Extended Edition. Sprinter-Verlag, Berlin Heidelberg New York ,1996.
[11] Tsutsui, S., Fujimoto Y.: Forking Genetic Algorithms with Blocking and Shrinking Modes (fGA). In: Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993, pp. 206-213.
[12] Eshelman, L. J., Schaffer, J. D.: Real-Coded Genetic Algorithms and Interval Schemata. In: Foundations of Genetic Algorithms.2, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993, pp.187-202.
[13] Joines, J., Houck, C.: On the Use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with Gas. In: Fogel, D. (Ed.): Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE Press, Orlando, FL, 1994, pp.579-584.
[14] Fonseca, C. M., Fleming, P. J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993, pp.416-423.
[15] Deb, K: Optimal Design of a Welded Beam via Genetic Algorithms. American Institute of Aeronautics and Astronautics Journal, Vol. 29, No. 11, 1991, pp.2013-2015.
[16] Tanatsugu, N., Carrick, P.,: Earth-to-Orbit Combined Rocket/Airbreathing Propulsion. American Institute of Aeronautics and Astronautics Paper 2003-2586 (2003)
[17] Kobayashi, H, Tanatsugu, N.: Optimization Method on TSTO Spaceplane System Powered by Airbreather. American Institute of Aeronautics and Astronautics Paper 2001-3965, 2001.
[18] Shimoyama, K., Fujii, K., Kobayashi, H.: Improvement of the Optimization Method of the TSTO Configuration – Application of Accurate Aerodynamics. Proceedings of Third International Conference on Computational Fluid Dynamics, 2001.