Proposal of Guidance Markers to Reduce the Time to Reach the Pareto-optimal Front

1st Yuji Sato
Faculty of Computer and Information Sciences
Hosei University
Tokyo, Japan
yuji@k.hosei.ac.jp

2nd Mikiko Sato
School of Information and Telecommunication Engineering
Tokai University
Tokyo, Japan
mikiko.sato@tokai.ac.jp

3rd Ruhul Sarker
School of Systems and Computing
University of New South Wales
Canberra, Australia
r.sarker@unsw.edu.au

***Abstract*—Companies often develop products with similar trends, such as improving existing products or creating product series. This paper proposes using non-dominated solutions in an objective function space corresponding to the design variables of past similar products as a Guidance Marker (GUM) to rapidly guide a population of individuals toward their desired destination. More specifically, we propose a method that uses difference vectors derived from the GUM to correct the population's search positions as a supplement to the genetic operations of the underlying evolutionary multi-objective optimization algorithm. Using the standard benchmark problem, the ZDT series, and a Human-Powered Aircraft (HPA) problem, we compare the solution search accuracy of the original NSGA-II with that of Hyper Volume, IGD, and IGD+, as well as the convergence speed to the Pareto-optimal front (PF), and the process of the population's convergence to the PF. The results show that the proposed method improves convergence speed and solution search accuracy compared to the original NSGA-II, without reducing the diversity of the solution distribution.**

Keywords—multi-objective evolutionary algorithms, NSGA-II, Improved convergence, Guidance Marker, Difference Vector

I. INTRODUCTION

In research into evolutionary multi-objective optimization (EMO) problems, evaluation experiments using standard benchmark problems are common. Since program execution is completed quickly, execution time is not an issue. On the other hand, when applying major EMO algorithms such as NSGA-II [1] and MOEA/D [2] to complex real-world problems, execution time often becomes a big issue. To address this issue, research has already begun on speeding up algorithms such as NSGA-II and MOEA/D by distributing and parallelizing them [3 - 7], as well as on surrogate models, which use machine learning to generate approximate models of objective functions that would otherwise require enormous execution times [8 - 12]. Both approaches are important for speeding up EMO algorithms. However, the former suffers from the problem of complex program implementation when the number of objective functions is three or more, while the latter suffers from the enormous training time required to generate surrogate models.

Meanwhile, companies often develop products with similar characteristics, such as improvements to existing products or product series. For this reason, a method has been proposed [13] for developing similar products using an EMO algorithm, in which design data from previous products is archived and included in the initial solution. In this paper [13], the authors conducted experiments in which they archived design-variable solutions corresponding to optimal solutions near the center of the Pareto optimal front (POF) obtained by solving a benchmark problem and included these solutions in the initial solution for other similar problems. The results showed that convergence to the POF improved for essentially all benchmark problems. On the other hand, some problems, such as ZDT3, have been reported to decrease the diversity of the solution distribution, leading to partial omission of the Pareto-optimal front. Furthermore, the authors reported that including solutions at both ends of the POF of similar products, which are unlikely to be selected as the final product, in the initial solution can potentially avoid the problem of partial omission of the POF.

However, if an EMO algorithm was not used in the design of past products, there would be no information about the POF and, therefore, no information about the solutions at either end. Furthermore, depending on the problem or the initial solution, it might be impossible to avoid the problem of reduced diversity, even if solutions at either end of the Pareto front of similar products are included in the initial solution. We therefore propose a method that uses non-dominated solutions corresponding to the design variables of previous similar products as guidance markers (GUMs) to guide the population of individuals toward the destination area quickly. Next, Section 2 discusses related research, and Section 3 describes the basic concepts and implementation of the proposed method. Section 4 presents evaluation experiments and analysis, concluding with a summary.

II. BACKGROUND AND RELATED WORKS

A. Constrained Multi-objective Optimization Problem

Multi-objective optimization is a method that simultaneously optimizes multiple objective functions in a trade-off relationship. The constrained multi-objective optimization problem (CMOP) is defined by the following equation.

$$\begin{cases} Minimize\ f_i(\boldsymbol{x}) & (i = 1,\ 2,\ \dots,\ m) \\ subject\ to\ g_j(\boldsymbol{x}) \geq 0 & (j = 1,\ 2,\ \dots,\ k) \end{cases}$$

where f is an objective function vector consisting of m conflicting objective functions and g is a constrained function.

The task is to find a set of $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ that minimizes/maximizes the m objective functions under the constraint functions.

Fig. 1 shows a conceptual diagram of a two-objective optimization problem. In multi-objective optimization, a currently known solution that has a good evaluation value for any particular objective function is called a non-dominated solution and is regarded as an optimal solution. As shown in Fig. 1, there are usually multiple non-dominated solutions. The set of non-dominated solutions in the objective space is called the Pareto optimal solution. The surface formed by the Pareto-optimal solutions is called the Pareto-optimal front (POF), and the purpose of a search algorithm, such as an MOEA, is to find the POF in a short time accurately.

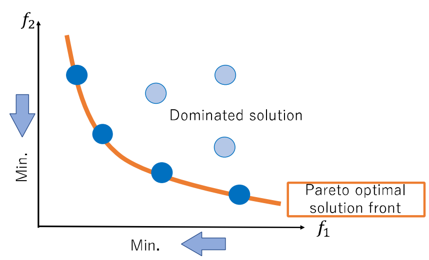


Fig. 1. Conceptual illustration of the objective space for two objective optimization problems.

B. *Previous Research Examples for Accelerating EMO*

When applying major EMO algorithms such as NSGA-II and MOEA/D to complex real-world problems, execution time is often a major issue as large population sizes are required to approximate the Pareto front with high accuracy. To address this issue, research has already begun on speeding up algorithms such as NSGA-II and MOEA/D by distributing and parallelizing them [3 - 7], and on surrogate models that use machine learning to generate approximate models of objective functions [8 - 12].

For example, as a distributed parallelization method for the EMO algorithm, Nebro et al. [3] proposed a parallel MOEA/D algorithm (pMOEA/D) that maintains solution diversity. pMOEA/D can reduce the number of data transfers compared to standard master-slave architectures. However, in many-core environments such as GPUs, aggregating all individual information on the master Processing Unit (PU) incurs significant overhead. Mambrini et al. [4] proposed the parallel decomposition algorithm (PaDe), which assigns different slave PUs to each subgroup of single-objective problems. At regular intervals, the worst T solutions from each PU (island) are replaced by the best T solutions from neighboring islands through migration. This method can also be implemented on GPUs, achieving faster speeds. However, because individuals in each subgroup are evaluated only using a function that aggregates the weight vectors of each PU, the distribution of Pareto-optimal solutions across subgroups is likely to be sparse. Derbel et al. [5] proposed MP-MOEA/D, which brings copies of populations of solutions belonging to neighboring weight vectors to each weight vector to mate solutions within that weight vector. When implemented on multi-core processors, this algorithm is well-suited for improving the speed of thread-based parallelization using shared memory, and a GPU implementation is also possible. However, because the groups of individuals are stored in global memory, the rate of performance improvement does not increase greatly relative to the increase in the number of CPU cores.

Next, research has begun using surrogate models to reduce the time required to evaluate the objective function. This process typically consists of three steps: (1) Sampling design variables within the feasible region and measuring their responses in the objective space. These serve as input data for training and their expected values, respectively. (2) Using this set of input data and expected values, a surrogate model is trained and validated. (3) The EMO algorithm is run using the surrogate model instead of the objective function. For example, Zhang et al. [10] proposed a CNN-based surrogate model for indoor airflow simulation, achieving a speedup of over 2000-fold compared to the CFD method. Similarly, Tang et al. introduced a hybrid CNN-ConvLSTM model [11] for predicting subsurface fluid evolution, achieving inference approximately 1000 times faster than CFD simulations with low prediction error. These studies demonstrate the potential of DNN-based surrogate models to achieve both high accuracy and significantly reduced computational time. Also, Liu et al. used transfer learning to shorten the training time of a surrogate model [12]. However, building training datasets for these models still requires significant training time.

Both approaches are important to speed up the EMO algorithm. However, the former has the problem that the program implementation becomes complicated when there are three or more objective functions, and the latter has the problem that generating a highly accurate surrogate model requires a huge amount of training time.

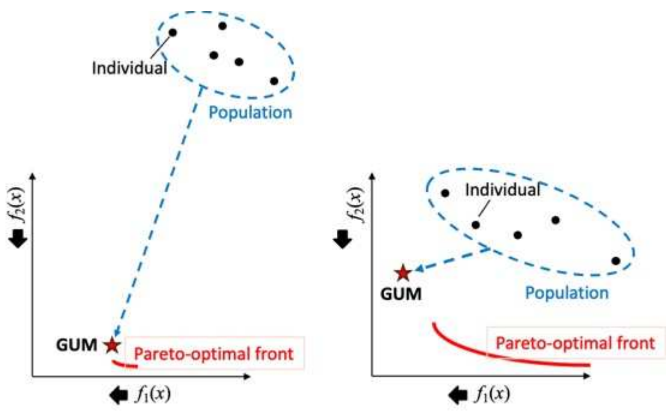
Taking these issues into consideration, a method for reducing execution time by leveraging design variables from similar products has been proposed [13]. Since companies often develop products with similar trends, such as improving existing products or creating product series, the idea is to improve the speed of convergence to the Pareto-optimal front by including individuals (non-dominated solutions) corresponding to the design variables of existing products in the initial solution when developing similar products using the EMO algorithm. This proposed method is worthy of praise, demonstrating improved convergence toward the POF across all benchmark problems used in the evaluation experiments. However, it does have issues with some benchmark problems, such as ZDT3, which lead to reduced diversity, such as the absence of several non-dominated solutions on the right end. To address this issue, it has been shown that the problems encountered with ZDT3 can be alleviated by including, in the initial solution for the next product development, not only non-dominated solutions corresponding to the design variables of existing similar products, but also non-dominated solutions at both ends of the POF developed using the EMO algorithm for existing similar products.

However, this improvement requires that the existing product be developed using the MO algorithm, and that non-dominated solutions at both ends of the POF be recorded in addition to the product's design variables. In other words, if the EMO algorithm has not been applied in the development of an existing product, this proposed method cannot be used.

III. Proposal of Guidance Markers (GUM)

A. Basic Concept of GUM

To address the issues of related research [13], we propose using non-dominated solutions in the objective function space corresponding to the design variables of similar products from the past as guidance markers (GUM) to roughly guide the population toward its desired destination. Figure 2 shows the relationship between the population, GUM, and the Pareto-optimal front. As shown in Figure 2-(a), when the population (a collection of candidate solutions) is far from the destination, the GUM indicates the direction the population should head, like a lighthouse guiding a ship in the dark. However, because the GUM deviates from the true destination, its guidance accuracy decreases as the population approaches the destination, as shown in Figure 2 (b). Therefore, the influence of the GUM must be reduced as the population evolves.



(a) Initial stage of POF search (b) Late stage of POF search

Fig. 2. Diagram of the relationship between populations, the guidance marker (GUM), and Pareto optimal fronts.

B. Correcting individuals by difference vectors using GUM

To realize this basic concept, we perform the ensemble correction process using the difference vector created based on the GUM in the following steps (a) to (d). The following explanation corresponds to (a) through (d) in Figure 3, which shows the shift factor α = 0.5 and the number of the correction process $N = 1$ in Equation (2).

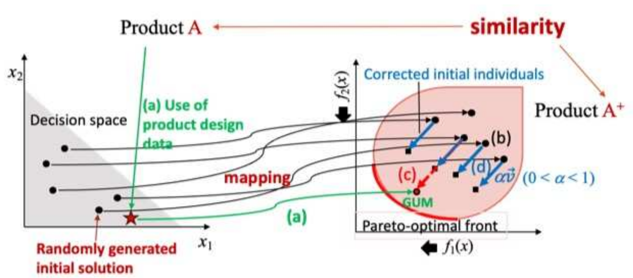


Fig. 3. A diagram explaining the general flow of the proposed method.

First, (a) define the GUM as a non-dominated solution in the objective function space corresponding to the design variables of past similar products. Next, (b) perform non-dominated sorting on the population of individuals to obtain a rank 1 population. Then, (c) obtain the difference vector from the (nearly) central individual of the rank 1 population to the GUM. Let $\boldsymbol{Vg} = (f_{1\text{gum}}, f_{2\text{gum}})$ be the coordinates of the GUM in the two objective function space, and $\boldsymbol{Vc} = (f_{1\text{c}}, f_{2\text{c}})$ be the coordinates of the non-dominated solution near the center of rank 1. The difference vector ΔV is defined as follows:

$$\Delta \boldsymbol{V} = \boldsymbol{Vg} - \boldsymbol{Vc} = (f_{1\text{gum}} - f_{1\text{c}}, f_{2\text{gum}} - f_{2\text{c}}) \quad (1)$$

Finally, (d) perform a correction process by shifting the rank 1 individuals by G(α) times the length of this difference vector. Here, G(α) is the decay function, and α is the shift coefficient. For example, $G(\alpha) = \alpha^N$ $(0<\alpha<1)$, where N is the number of iterations of the correction process. The shift amount for a non-dominated solution is defined as follows:

$$G(\alpha)\,\Delta \mathbf{V} = \alpha^N(f_{1\text{gum}} - f_{1\text{c}}, f_{2\text{gum}} - f_{2\text{c}}) \quad (2)$$

The above processes (b) to (d) are applied N times at regular generation intervals. However, when the approximate position of the POF is known in advance, such as when using a standard benchmark problem that converges quickly to the POF, or a GUM obtained using the EMO algorithm, the value of α is determined using this knowledge and applied only once. On the other hand, for real-world problems that do not converge to the Pareto-optimal front within a practical amount of time and for problems where there is no prior knowledge of the search space, the processes are applied repeatedly (N times) at regular generation intervals, taking into account the diversity of the population.

It is conceivable that shifting individual positions using the difference vector could be applied to all individuals, but to avoid early convergence and maintain population diversity, we will limit its application here to individuals with rank 1, for example. A comparative experiment with the case where all individuals are shifted is future work.

Figure 4 shows an example of the change in the population when the proposed method is actually applied to a combination of NSGA-II and the standard benchmark problem ZDT1. In Figure 4, the number of individuals is 50, α = 0.5, and $N = 1$. The blue circles are individuals with rank 1 among the initial individuals (non-dominated solutions), and the red circles are individuals resulting from shifting (correcting) the rank 1 individuals using the proposed method.

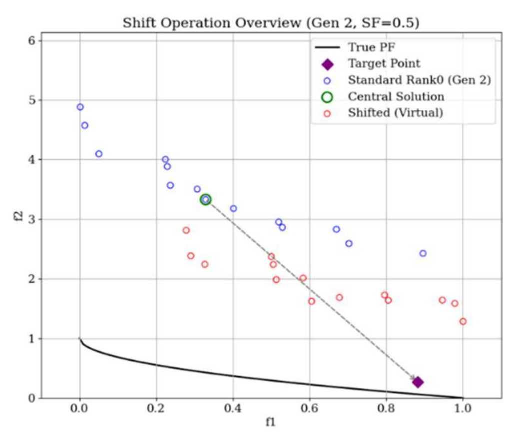


Fig. 4. The change in the population when the proposed method is actually applied to a combination of NSGA-II and the standard benchmark problem ZDT1.

IV. EVALUATION

A. Experimental method

a) Benchmark Problems: To compare our approach with related research [13], we use NSGA-II as the EMO algorithm, as in the research, and conduct evaluation experiments using the ZDT series (ZDT1, ZDT2, ZDT3, ZDT4) [14], a set of representative two-objective optimization problems, as benchmark problems. ZDT1 and ZDT2 are problems for which solutions are relatively easy to find when using NSGA-II. ZDT3 is a problem for which the research [13] observed partial loss of POFs (reducing the diversity of the solution distribution). ZDT4 is a problem that is relatively difficult to find using NSGA-II. Additionally, to evaluate its effectiveness for problems that require a long time to converge, a Human-Powered Aircraft (HPA) problem, HPA204 [15], was added to the evaluation problems. NSGA-II was run in advance with initial values different from those used in the evaluation experiments, and one individual was randomly selected from the population of individuals in the middle of the search that had come close to POF to a certain extent and set as the GUM. The test execution environment is summarized in Table I.

TABLE I. EXPERIMENTAL ENVIRONMENT

Category	Library Name	Version
Programming Language	Python	3.10.12
Optimization Algorithm	pymoo	0.6.1.6
Numerical Computation	numpy	2.2.6
Scientific Computing	scipy	1.15.3

b) Evaluation Items: The effectiveness of the proposed method will be evaluated based on the convergence speed to POF and solution accuracy. The convergence speed to POF will be evaluated based on the relationship between the number of generations and the hypervolume (HV) value [14]. The solution accuracy will be evaluated by observing the HV value, IGD [16], and IGD+ [17] from the final POF, as well as whether there is a lack of a part of the POF, as observed in related research. Consistent with the research [13], the population size will be set to 50.

B. Experimental results and discussion

a) Experiments Using the ZDT Series: Figures 5 through 8 show the relationship between the number of generations and the HV value for ZDT1 through ZDT4, respectively, for the original NSGA-II and the proposed method. Similarly, Tables 1 through 4 show the HV, IGD, and IGD+ values calculated from the final POF for the original NSGA-II and the proposed method. In each case, 31 trials were performed with different initial values, and the HV value evolution graph shows the experimental results for the median HV value. The other results are the experimental results for the average values for the 31 trials. In all cases, the value of the difference vector coefficient α was set to 0.5, and population correction using the proposed method was applied only once at the beginning of the initial population.

Figures 5-8 and Tables 1-4 show that for all benchmark problems, ZDT1-ZDT4, the proposed method improved the convergence speed of NSGA-II to POF and also improved the diversity of the solution distribution compared to standard NSGA-II, which randomly generates initial solutions. In particular, for ZDT4, where solution search is relatively difficult, the speed of convergence to POF was significantly faster than standard NSGA-II. Furthermore, the phenomenon of missing parts of the POF for ZDT3, which was observed in related research [13], was not observed in any of the 31 trials. Furthermore, compared to standard NSGA-II, the HV values using the proposed method tend to improve both in terms of mean and variance. As an example, Figure 9 shows a box-and-whisker plot of the HV values for the original NSGA-II and the proposed method using ZDT1.

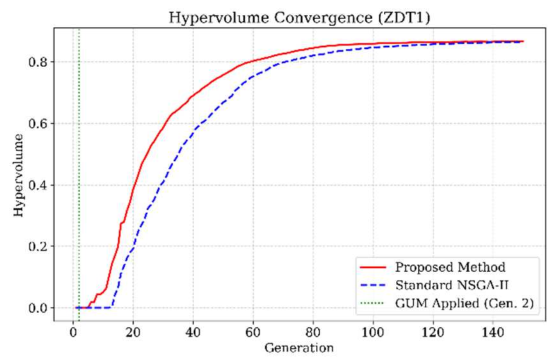


Fig. 5. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for ZDT1. (α = 0.5)

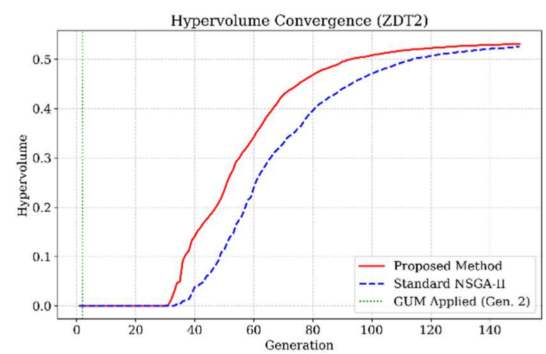


Fig. 6. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for ZDT2. (α = 0.5)

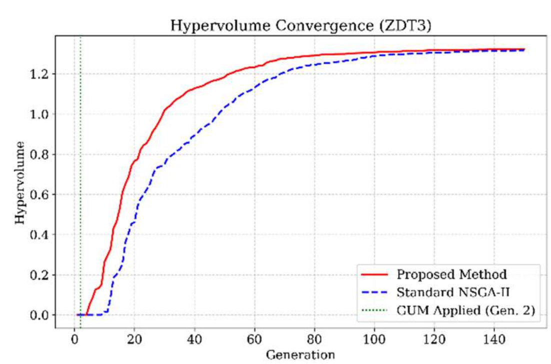


Fig. 7. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for ZDT3. (α = 0.5)

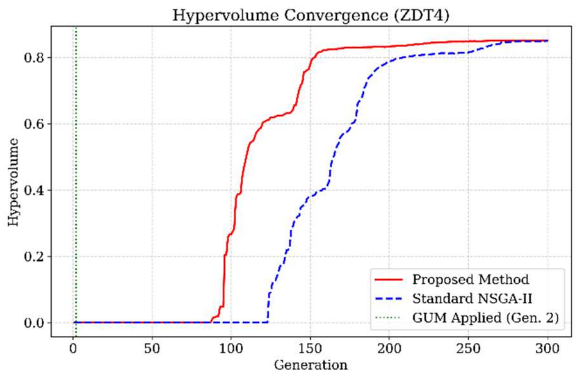


Fig. 8. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for ZDT4. (α = 0.5)

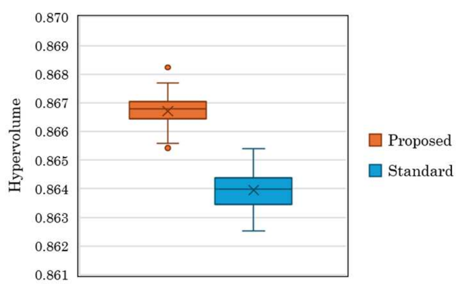


Fig. 9. The box-and-whisker plot of the HV values for the original NSGA-II and the proposed method using ZDT1.

TABLE II. HV, IGD, AND IGD+ VALUES FOR ZDT1.

Method (ShiftGen)	HV	IGD	IGD+
Proposed method (2)	0.8669	0.0056	0.0050
Standard NSGA-II	0.8639	0.0070	0.0066

TABLE III. HV, IGD, AND IGD+ VALUES FOR ZDT2.

Method (ShiftGen)	HV	IGD	IGD+
Proposed method (2)	0.5310	0.0066	0.0060
Standard NSGA-II	0.5275	0.0083	0.0078

TABLE IV. HV, IGD, AND IGD+ VALUES FOR ZDT3.

Method (ShiftGen)	HV	IGD	IGD+
Proposed method (2)	1.3204	0.0074	0.0041
Standard NSGA-II	1.3163	0.0083	0.0052

TABLE V. HV, IGD, AND IGD+ VALUES FOR ZDT4.

Method (ShiftGen)	HV	IGD	IGD+
Proposed method (2)	0.8557	0.0125	0.0111
Standard NSGA-II	0.8473	0.1612	0.1546

Experiments Using HPA204: HPA is an aircraft that obtains propulsion by pedaling like a bicycle, rotating the propeller. The HPA problem [15] deals with the optimization of the HPA's wing design, including the wing shape, carbon fiber reinforced plastic (CFRP) pipe lamination, wing root dihedral angle, wire tension, and payload. The benchmark problems provided in [15] are single-objective or multi-objective minimization problems in a normalized design space.

Here, we specifically selected the HPA204 problem provided by the Python library pymoo. HPA204 is a two-objective unconstrained optimization problem, but it has a high degree of freedom in design variables and is a difficult problem, formulated in the following format:

$$\textit{Minimize}\ \boldsymbol{F}(\boldsymbol{x}_l)=(f_1(\boldsymbol{x}_l), f_2(\boldsymbol{x}_l))$$

$$\boldsymbol{x}_l \in [0,1]^{D_l}$$

Where the number of wing divisions "n" is 4, the maximum number of layers "m" is 20, and the dimensions parameter l is 1. The dimensions D_l of the design variables for HPA204 are set to $D_0 = 18$, $D_1 = 33$, and $D_2 = 189$, respectively. For details, see reference [15].

Figure 10 shows the relationship between the number of generations and the HV value in HPA204 for the original NSGA-II and the proposed method. Similarly, Table 5 shows the HV, IGD, and IGD+ values calculated from 300 generations of HPA204 for the original NSGA-II and the proposed method. Thirty-one trials were performed using different initial values, and the HV value evolution graph shows the experimental results for the median HV value. The IGD and other values shown in the table are the average values for the 31 trials. As with the experiments using the ZDT series, the value of the difference vector coefficient α was set to 0.5, and the correction using the proposed method was applied only once at the beginning of the initial population.

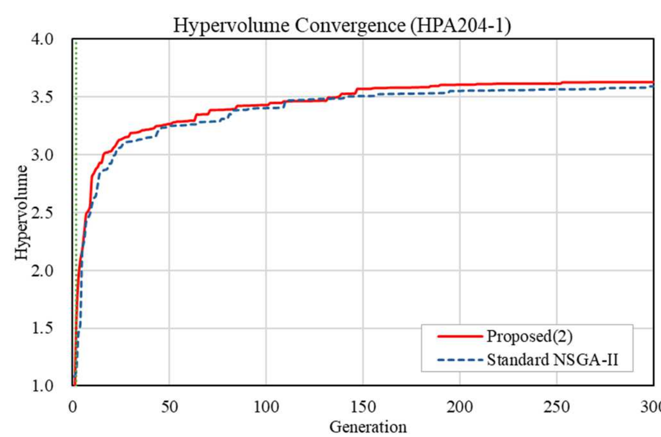


Fig. 10. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for HPA204. (Applied to 2nd gens)

TABLE VI. HV, IGD, AND IGD+ VALUES FOR HPA204.

Method (ShiftGen)	HV	IGD	IGD+
Proposed method (2)	3.6260	0.3300	0.2212
Standard NSGA-II	3.5770	0.3231	0.2456

As can be seen from Figure 10 and Table 6, in the case of the HPA204 problem, which requires many generations to converge to POF, applying the proposed method only once to the initial population, resulted in only a slight improvement in HV value and IGD+, and no significant effect was observed. Therefore, below we present experiments in which the initial value of the difference vector coefficient α is set to 0.5, and the results of applying the proposed method's population correction three times using equation (2) are shown. The difference vector coefficient is $(0.5)^1 = 0.5$ for the first application of the proposed method, $(0.5)^2 = 0.25$ for the second application, and $(0.5)^3 = 0.125$ for the third application. In other words, the influence of the difference vector decreases as the number of generations increases.

Figures 11 to 13 show the relationship between the number of generations and HV values for the original NSGA-II and the proposed method when the proposed method, which performs population correction processing using difference vectors created with GUM, is applied three times to the HPA204 problem. In this experiment, 31 trials were conducted with different initial values, and the HV value transition graph shows the median value over the 31 trials. Using Equation (2), Figure 11 shows the results of applying the proposed method's population correction to 2, 10, and 20 generations. Figure 12 shows the results of applying the proposed method's population correction to 2, 30, and 60 generations. Figure 13 shows the results of applying the proposed method's population correction to 2, 100, and 200 generations. Table 6 shows the HV, IGD, and IGD+ values for the original NSGA-II and the proposed method, calculated from the POF obtained at the 300th generation in Figures 11-13. Figure 14 shows the box plots of the HV values for the original NSGA-II and the proposed method using HPA204. In the box plot, the bottom of the whisker represents the minimum value, the bottom edge of the box represents the first quartile, the middle line represents the second quartile (median), the top edge represents the third quartile, and the top of the whisker represents the maximum value. The x marks inside the box represent the mean, and points plotted away from the whiskers represent outliers. Figure 15 shows the population distribution during the solution search process, corresponding to Figure 11.

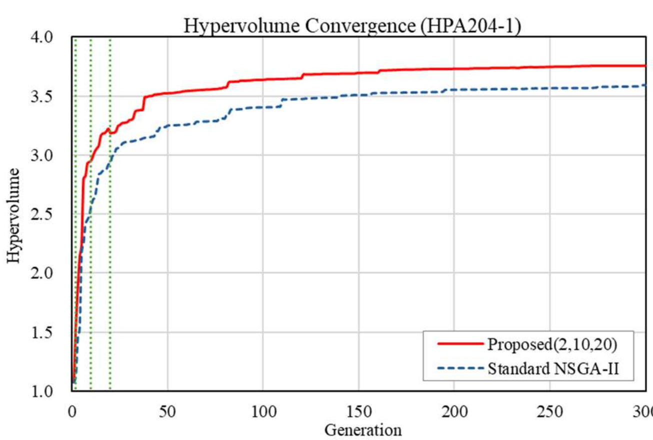


Fig. 11. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for HPA204. (Applied to 2, 10, and 20 generations)

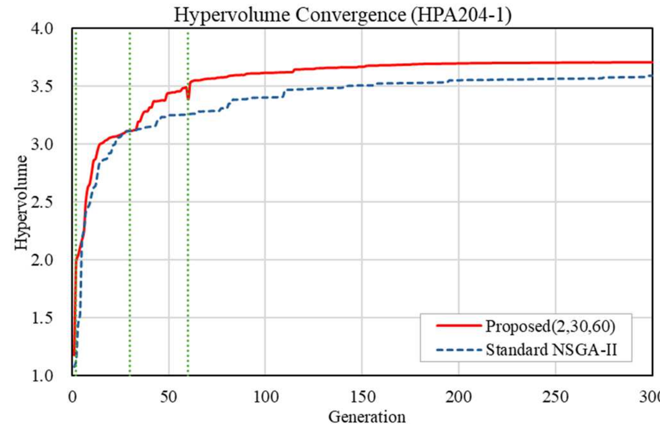


Fig. 12. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for HPA204. (Applied to 2, 30, and 60 generations)

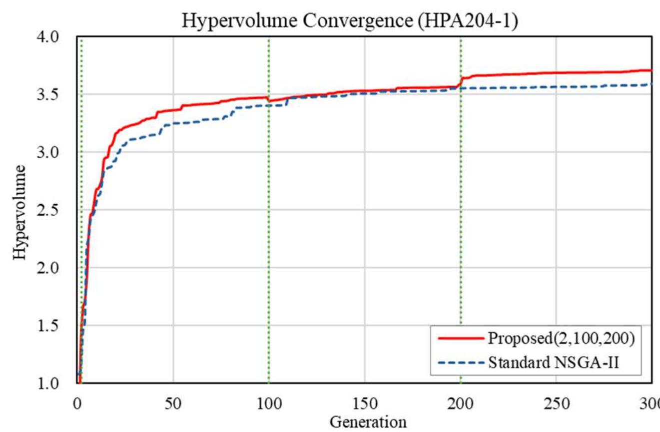


Fig. 13. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for HPA204. (Applied to 2, 100, and 200 generations)

TABLE VII. HV, IGD, AND IGD+ VALUES FOR HPA204.

Method (ShiftGen)	HV	IGD	IGD+
Proposed (2,10,20)	3.7175	0.2910	0.1769
Proposed (2,30,60)	3.6994	0.3000	0.1844
Proposed (2,100,200)	3.6894	0.3315	0.1838
Standard NSGA-II	3.5770	0.3231	0.2456

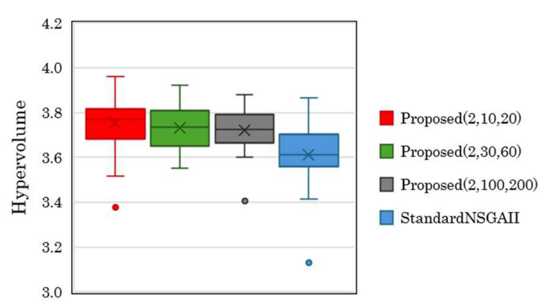


Fig. 14. The box-and-whisker plot of the HV values for the original NSGA-II and the proposed method using HPA204. (Comparison of applying generation intervals)

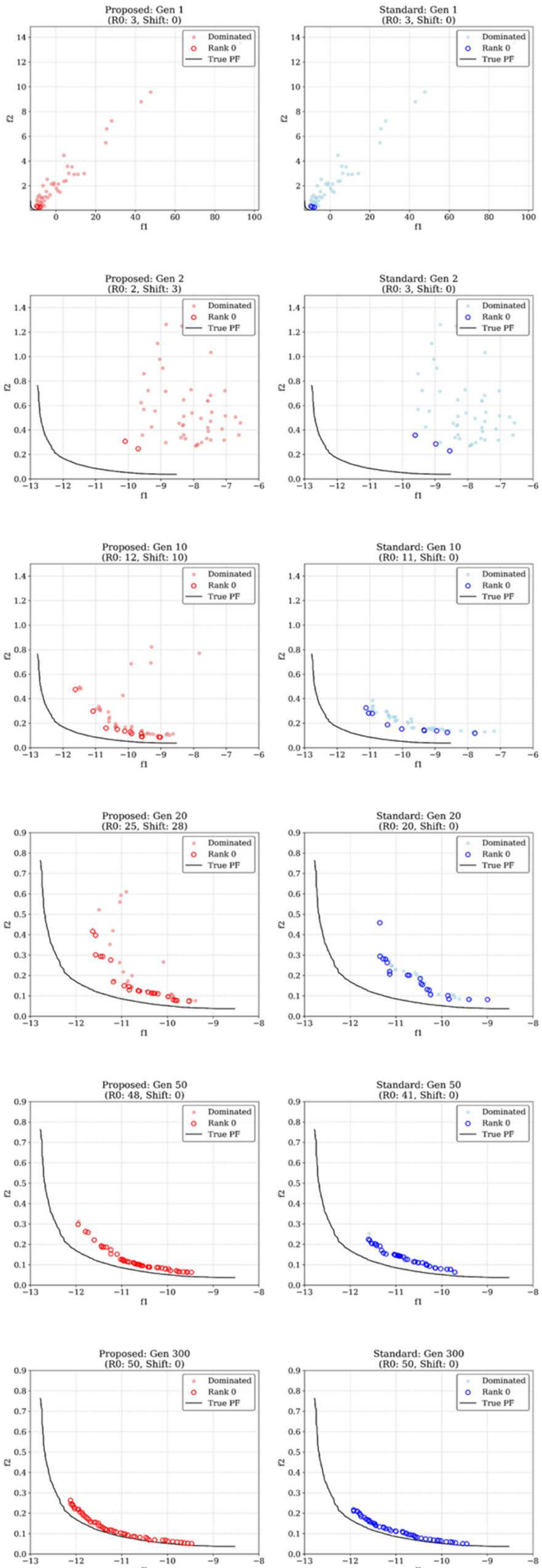


Fig. 15. The distribution of individuals during the solution search process for HPA204, corresponding to Figure 10.

First, based on the comparison of HV, IGD, and IGD+ values in Table 7 and the box-and-whisker plot of the HV values in Figure 14, the proposed method appears to be effective in improving HV and increasing the diversity of the solution distribution compared to standard NSGA-II, which randomly generates initial solutions. Next, Figures 11 to 14 and 15 show that applying the proposed method, especially in the early stages of the search, significantly improves convergence to POF and increases the diversity of the solution distribution during the search, compared to standard NSGA-II, which randomly generates initial solutions. In particular, Figure 11 shows that the HV value achieved by standard NSGA-II in 300 generations is achieved by the proposed method in approximately 80 generations. Furthermore, the increase in HV value in generations 2, 10, and 20 when the proposed method is applied suggests that the proposed method contributes to improving solution search capabilities.

Next, to examine the effect of the number of times the proposed method was applied, Figure 16 shows the relationship between the number of generations and the HV value when the proposed method was applied two more times (at 30 and 40 generations) using Equation (2) to the results shown in Figure 11, which showed the best results. Table 8 compares the HV value, IGD, and IGD+ depending on the number of applications. These results show that applying the proposed method five times resulted in slight improvements in the HV value, IGD, and IGD+ compared to applying it three times, indicating that further improvements may be possible by considering the definition of the decay function and the generation interval for applying the proposed method.

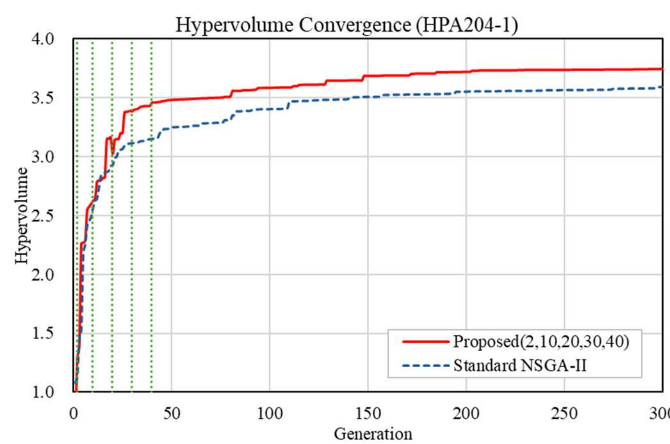


Fig. 16. Relationship between the number of generations and HV value in the original NSGA-II and the proposed method for HPA204. (Applied to 2, 10, 20, 30, and 40 generations)

TABLE VIII. HV, IGD, AND IGD+ VALUES FOR HPA204.

Method (ShiftGen)	HV	IGD	IGD+
Proposed (2)	3.6260	0.3300	0.2212
Proposed (2,10, 20)	3.7175	0.2910	0.1769
Proposed (2,10, 20, 30, 40)	3.7219	0.2794	0.1746
Standard NSGA-II	3.5770	0.3231	0.2456

Based on these experimental results, we think this proposal is effective for developing similar products such as version upgrades and product series. In particular, given that actual aircraft design using EMO has begun [18], we believe that the effective results demonstrated in evaluation experiments using human-powered aircraft represent an example of a practical application area where the proposed method can be effectively utilized. On the other hand, in the development of entirely new products, we believe that the proposed method is not in competition with approaches such as speeding up development through distributed parallelization or reducing the processing time of objective functions using surrogate models, but rather that a synergistic effect can be achieved by combining them effectively. However, defining the damping function, determining the generation interval to which the proposed method is applied, and evaluating the proposed method using a wider range of problems such as three-objective problems and real-world problems remain as future challenges.

V. CONCLUSION

In this paper, we proposed a method for quickly guiding a population toward its desired destination by using non-dominated solutions in the objective function space corresponding to the design variables of past similar products as guidance markers (GUMs). Specifically, we proposed a method for correcting the population's search position using difference vectors derived from the GUM as a complement to the genetic operations of the underlying EMO algorithm. Using the standard benchmark problems, the ZDT series, and the Human-Powered Aircraft problem HPA204, we compared the solution search accuracy with the original NSGA-II in terms of the hypervolume, IGD, and IGD+. Results show that the proposed method can improve convergence speed and solution search accuracy compared to the original NSGA-II without compromising the diversity of the solution distribution.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Numbers JP25K15279.

REFERENCES

[1] Q. Zhang, H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition", in IEEE Transactions on Evolutionary Computation, Volume 11, Issue 6, pp. 712-731, 2007.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[3] A. J. Nebro and J. J. Durillo, "A Study of the Parallelization of the Multi-Objective Metaheuristic MOEA/D", in International Conference on Learning and Inteligent Optimization (LION4), pp. 303–317, 2010.

[4] A. Mambrini and D. Izzo, "PaDe: A parallel algorithm based on the MOEA/D framework and the island model", in Parallel Problem Solving from Nature (PPSN XIII), pp. 711–720, 2014.

[5] B. Derbel, A. Liefooghe, G. Marquet, and E. Talbi, "A fine-grained message passing MOEA/D", in Proceedings of 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1837–1844, 2015.

[6] Y. Sato, M. Sato, M. Midtlyng, M. Miyakawa, "Parallel and Distributed MOEA/D with Exclusively Evaluated Mating and Migration", Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC 2020). pp. 1-8 (CD-ROM), 2020.

[7] S. G. Machicao-Mollocondo, E. Hinojosa-Cardenas, "Parallel Multi-objective Evolutionary Algorithms: A Systematic Literature Review", Proceedings of the 11th Annual International Conference of Information Management and Big Data (SIMBig 2024), pp. 93 – 109, 2024. https://doi.org/10.1007/978-3-031-91428-7_7

[8] Alan Díaz-Manríquez, Gregorio Toscano, Jose Hugo Barron-Zambrano, Edgar Tello-Leal, "A Review of Surrogate Assisted Multiobjective Evolutionary Algorithms", WILEY Online Library, pp. 1- 14 (Online), 2016. https://doi.org/10.1155/2016/9420460

[9] Tao Zhang, Fuzhang Li, Xin Zhao, Wang Q, Tianwei Liu, "A Convolutional Neural Network-Based Surrogate Model for Multi-objective Optimization Evolutionary Algorithm Based on Decomposition", Swarm and Evolutionary Computation, Vol. 72 (3):101081, 2022. https://doi.org/10.1016/j.swevo.2022.101081

[10] W. Zhang, C. Zhang, Y. Zhao, Z. Wang, Y. Liu, C. Zhou, Y. Hu, "Convolutional neural networks-based surrogate model for fast computational fluid dynamics simulations of indoor airflow distribution", Energy and Buildings p. 115020, 2024.

[11] M. Tang, Y. Liu, L.J. Durlofsky, "A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems", Journal of Computational Physics 413, 109456, 2020.

[12] J. Liu, A. Gupta, C. Ooi, Y. Ong, "ExTrEMO: Transfer Evolutionary Multiobjective Optimization With Proof of Faster Convergence", *IEEE Transactions on Evolutionary Computation*, Vol. 29, Issue 1, pp. 102-116, 2024.

[13] Y. Sato, H. Yamada, M. Sato, "Initial Population of NSGA-II for Solving Similar Multi-Objective Optimization Problems", Proceedings of the 2025 ACM/SIGEVO Genetic and Evolutionary Computation Conference (GECCO) Companion, pp. 109-110, 2025. DOI: 10.1145/3712255.3734264

[14] K. Deb, "Multi-Objective Optimization using Evolutionary Algorithms", John Wiley & Sons, 2001.

[15] N. Nobuo, "Single and multi-objective optimization benchmark problems focusing on human-powered aircraft design", *International Conference on Evolutionary Multi-Criterion Optimization*. Singapore: Springer Nature Singapore, pp. 195-210, 2025.

[16] D.A. Van Veldhuizen, Gary B. Lamont, "On measuring multiobjective evolutionary algorithm performance", in Proceedings of 2000 IEEE Congress on Evolutionary Computation (CEC), pp. 204–211, 2000.

[17] H. Ishibuchi, R. Imada, N. Masuyama, Y. Nojima, "IGD+ from the Viewpoint of Optimal Distributions of Solutions", in Proceedings of 2019 Evolutionary Multi-Criterion Optimization (EMO), pp. 332–345, 2019.

[18] S. Obayashi; D. Sasaki; Y. Takeguchi; N. Hirose、" Multiobjective evolutionary computation for supersonic wing-shape optimization", IEEE Transactions on Evolutionary Computation, Volume: 4, Issue: 2, pp. 182 – 187, 2000. DOI: 10.1109/4235.850658