Adaptive Update of Reference Point Value for Parallel and Distributed MOEA/D

Zhou, Xuan

(出版者 / Publisher)
法政大学大学院情報科学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編
(巻 / Volume)
18
(開始ページ / Start Page)
1
(終了ページ / End Page)
6
(発行年 / Year)
2023-03-24
(URL)
https://doi.org/10.15002/00026286

Adaptive Update of Reference Point Value for Parallel and Distributed MOEA/D

Zhou Xuan
Graduate School of Computer and Information Sciences
Hosei University
Tokyo 184-8584, Japan
zhouxuan8417@gmail.com

***Abstract*—**This paper proposes an updated method to determine the reference point for acceleration based on Multi-Objective Evolutionary Optimization Algorithm Parallel and Distributed Decomposition (MOEA/D) to process a multi-core environment. A method had been already proposed in switching update intervals which is not only effective for decreasing sparse regions in the initial stage, and improves multiformity and increases comparison of HV value and virtual overlap region method. However, this proposed method has limitations because the switching of the update interval needs to be manually set to find an appropriate switching value. In this paper, a method for automatically switching the update interval is proposed, which can be applied to various problems. Evaluating a newly proposed method, the convergence and diversity were compared by using a single-core MOEA/D and a parallel MOEA/D constrained knapsack matter, and prove that with a small number of generations it is not only effective for reducing the sparse regions up and also can improve diversity and increase HV value.

Keywords— MOEA/D, Parallel and Distributed Processing, Reference Points, Multi-Objective Evolutionary Algorithms；

I. INTRODUCTION

In practical problems, there are usually multiple nonlinear objectives in one model which need to be satisfied at the same time. Although these objective functions need to be optimized at the same time, they are normally conflict with each other. Those problems are called multi-objective programming issues. Decomposition is the basic tactics for traditional multi-objective optimization. Many evolutionary multi-objective optimization (EMO) algorithms had been proposed to pareto optimal solution set is obtained uniformly[1-3]. Ideally, Pareto optimal solutions should be excellent in either divergence or convergence. In EMO algorithms, in order to approximate the Pareto front line with high precision, a sufficient population size should be set. It has been studied that the parallelization of EMO algorithm attention on speed, so that the main target is to set the Pareto optimal solution in a shorter time than that of using an only core, as well as no reduced solution research accuracy [4, 5].

This study focuses on the distributed parallel processing of the multi-objective evolutionary algorithm MOEA/D. MOEA/D is a decomposition-based algorithm that uses scalar functions as a very effective solution [2]. In order to deal with this problem, two methods can be implemented. The first one called "virtual overlap method", which tries to accelerate MOEA/D in parallel from many core environments, such as graphics processing unit (GPU), without reducing the solution search accuracy [5]. This virtual overlapping way would divide the original divide into N partitions, and there are also overlapping parts between each partition for information sharing. Migrating to adjacent subgroups at specified intervals is the best solutions in these overlapping areas. Compared with the standard Island migration model, this method could increases the HV value, improves the diversity and reduce the sparse area. However, this method does not solve the problem while there is no sparse region at the partition boundary in the primary stage. The second method is to switch the update interval is used to consider the progress of the solution research to determine the reference point z^* and nadir point z^{nad}. As a result, this method effectively solves the sparsity issues in the primary stage. However, the change of the update interval proposed in this method is artificially specified. In this paper, a method of automatically changing the update interval was proposed, which sets the update interval as a variable to change with the change rate of z^* and z^{nad}. Based on results, the basic formula for the change of the update interval was defined.

II. BACKGROUND AND RELATED WORKS

A. Conventional Parallel and Distributed MOEA/D

In order to obtain a high accuracy approximation of PF, the population size should be set sufficiently. Moreover, as the population size increases, the computational complexity required to find POS increases exponentially. Therefore, execution time may be an issue when applying this approach to practical applications. In this paper, the Tchebycheff function (g^{te}) was used to handle parallelization of MOEA/D to minimize:

$$g^{te}(x, \lambda) = \max_{i \in (1,\dots,m)} \lambda_i |z_i^* - f_i(x)| \quad (1)$$

where x is a feasible solution, $\lambda = (\lambda_1, \dots, \lambda_m)$ is a positive weight vector, and $z^* = (z_1^*, \dots, z_m^*)$ is a reference point.

Currently, the simplest implementation of MOEA/D parallel acceleration is using master-slave mode, but the data transmission time between the master and slave PU would slows down the master and slave systems. Hence, this method is not suitable for large-scale parallel acceleration.

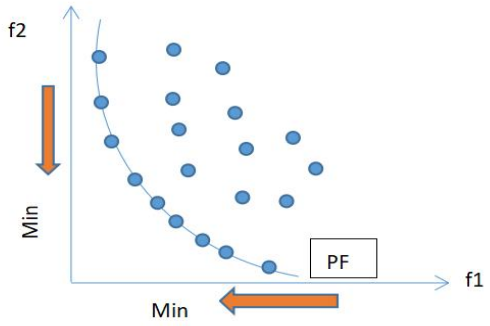

Fig. 1: Conceptual diagram of two objectives optimization problem.

B. Virtual Overlap Method

A massively parallel MOEA/D method suitable for GPU and other multi-core architectures was proposed [5]. Figure 2 shows the partition image assigned to each PU group.

In this approach, a virtual overlap region at the partition boundary was defined and evaluated exclusivity. Each partition has weight vectors and individual information, which performs well MOEA/D in parallel. Importantly, this approach defined a virtual overlap region at the partition boundary and evaluated exclusivity within the overlap region. For example, in Figure 2, the default region for partition 3 is P3A to P3B, P2BB to P3AA, and P4AA to P3BB are defined as virtual overlap regions.

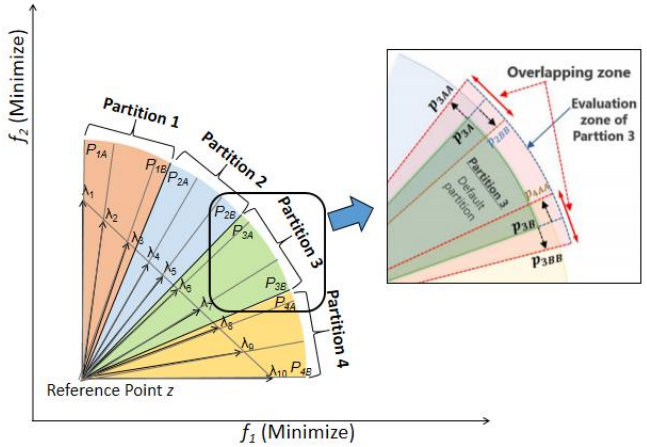

Fig. 2 The concept of the partition and the virtual overlapping zone for parallel MOEA/D. [5]

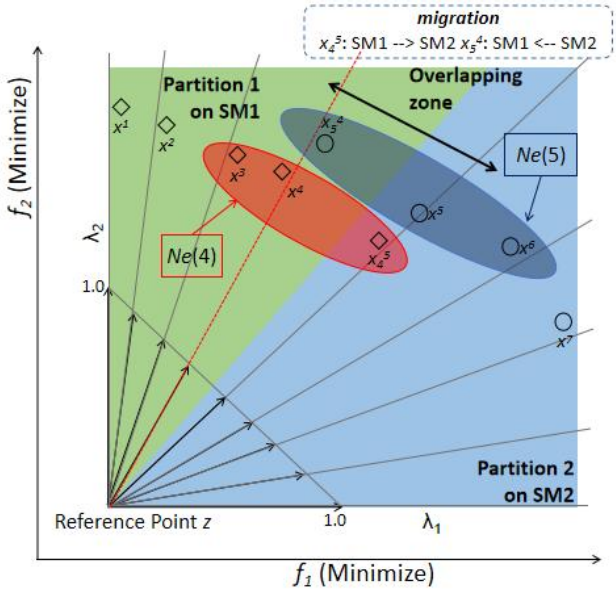

Fig. 3 The conceptual diagram of exclusively evaluating weight vectors near the boundary [5].

Figure 3 shows the concept map of shared weight vector near the boundary of two adjacent partitions in the virtual overlapping region when the size of T-neighbor is 2, where the evaluation range of SM1 is the weight vector λ^1 to λ^5, the individuals x^1 to x^4 and x_4^5, and the evaluation range of SM2 is the weight vector λ^4 to λ^9, the individuals x^5 to x^9 and x_5^4. Here, x_{ti} is a special individual in the overlapping region adjacent to T's neighbor Ne(t), evaluated using the weight vector λ^i. At appropriate intervals, perform a migration to copy x_4^5 belonging to SM1 to SM2 and x_5^4 belonging to SM2 to SM1.

C. Toggle update interval method

Another method was proposed to update datum and nadir according to the rate of variation. According to the step one of this essay, it has been investigated and discussed that when the change rate of z^* and z^{nad} decreases, the efficiency may be increased by increasing the update period of z^* and z^{nad}. As shown in figure 5, the update interval of z^* was reduced at the initial phase of solution search due to the large change rate of z^* and z^{nad} at this time. This prevented sparse solution distribution problem between partitions. After the change rate of z^* and z^{nad} slows down, a longer update interval was set for z^* and z^{nad} to save the time needed for those of updating. By taking this approach, i seek to accelerate and improve the diversity of solution distribution.

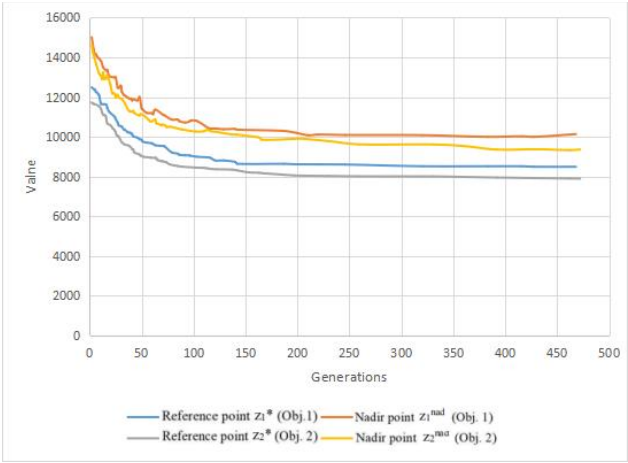

Fig. 4 Relationships between the number of generations and z^*,z^{nad} values.

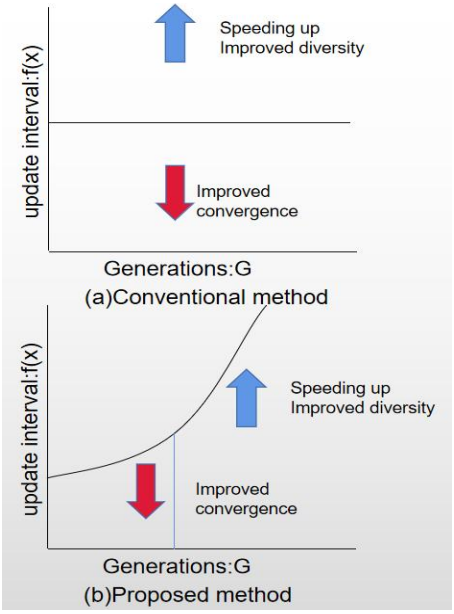

Fig. 5 The concept of update interval for reference points and nadir points.

The results show that, at the initial solution stage of search, this method not only reduces sparse areas, but also

improved the diversity of the common best prospects, increased the diversity compared to MOEA/D on a single core. However, the change of z* update interval still needs to be manually specified in this method, which is very inconvenient in practical application. Therefore, this paper proposed a design method of the update interval is automatically set according to the convergence status. I expect that the issues of sparse solution distribution between partitions can be effectively prevented and the whole process can be accelerated with the use of automatic update interval.

III. ADAPTIVE RESEARCH METHODS

A. *Update interval change method*

Here is an adaptive method used in this research. First of all, the interval at where our Z* and Z^{nad} point values are updated must be related to the speed at which Z* and Z^{nad} change. In order to predict the change of Z* and Z^{nad} accurately, an equation similar to the change curve of Z* was used.

In figure 4, it is clear that the relationship between Z_2^* and generation number G is when G approaches infinity, Z_2^* is close to 8000. Therefore, I can assume the change relationship equation of Z_2^* and G is:

$$Z = Ae^{-G} + B \qquad G \in [0, +\infty] \tag{2}$$

In formula (2), Z was used to represent Z_2^*, and G to represent the generation number G. As shown in figure 4, when $G=+\infty$, Z=8000. So B=8000 in formula (2). A point was taken according to the value of B in figure 5. When G=0, Z=11729. I set G=0, Z=11729, B=8000 in formula (2), and then got A=3729.

In summary, the change relationship equation of Z_2^* and G is:

$$Z = 3729e^{-G} + 8000 \qquad G \in [0, +\infty] \tag{3}$$

I made different calculus using the formula (3) to obtain the change relationship equation between the slope Z' and generation number of G:

$$Z' = -3729e^{-G} \tag{4}$$

As shown in figure 4 and figure 5, i can know when the slope Z' increases, the update interval x decreases, and when the slope Z' decreases, the update interval x increases. Therefore, i can assume that the relationship function between the slope Z' and the update interval x is an inverse proportional function. The specific equation is below:

$$x = \frac{C}{Z'} + D \tag{5}$$

i substituted formula (4) into formula (5) to get:

$$x = \frac{C}{-3729e^{-G}} + D \tag{6}$$

Because of the update interval $x \geq 1$, when Z' approaches infinity, x=1, then D=1. So while substituting D=1 into the formula (5)。

Regarding to the determination of the value C, i can use experimental results in the paper named *Inconstant Update of Reference Point Value For Parallel and Distributed MOEA/D* by Mr. Sato to estimate the range of value C. In this experimental result, when the generations number G<30, the update interval x=15, and when the generations number G>30, the update interval x=1000. Therefore, i can substitute G=2, x=15, and G=30, x=1000 into Formula (6) to obtain the value range of C. The calculated result is:

$$C \in (-7368, -5 \times 10^{-9})$$

Based on the value range of C, the appropriate C value can be selected by doing comparative experiments. Since the update interval x in formula (6) grows exponentially when the generation number gradually increases, the update interval x would increase as fast as result. So i set the update interval x=1000 in the program when the update interval x>1000. Finally, i compare C=-0.1/-1/-100/-3000 with constant (200) to find the most suitable C value.

Substituting D=1 and different C values into formula (6) can obtain Figure 6 and Figure 7.

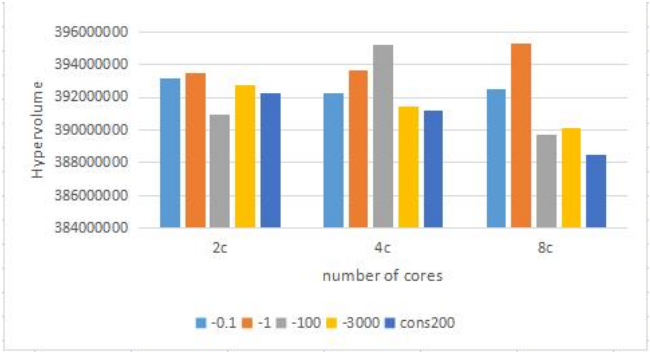

Fig. 6 Comparison of different C values and the HV value obtained by constant 200

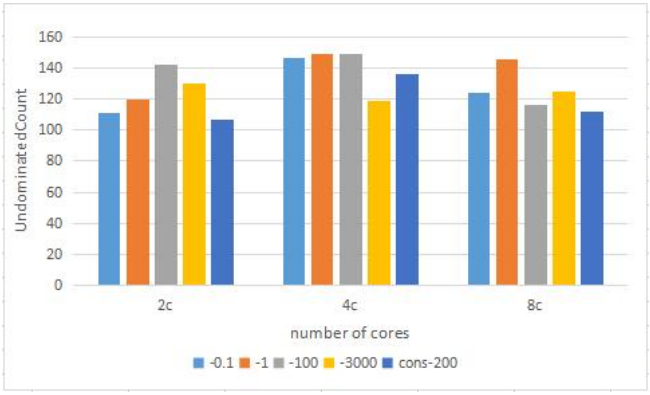

Fig.7 Comparison of different C values and the Undominated Count value obtained by constant200.

According to figure 6 and figure 7, it could be seen that when C=-1, best result will be got, so i selected C=-1 to continue the following experiment. Finally, the relationship between the update interval x and the generation number G in formula (6) can be rewritten below:

$$x = \frac{-1}{-3729e^{-G}} + 1 \tag{7}$$

B. *Algorithmic Flow*

According to the formula (7), the process of judging whether to update some algorithms is below:

Input:
x: the current update interval (x=1)
G:The current updated generation number.
1. for 0<G<5000,G++ do;
2. if(x>1000)
3. x=1000;
4. end
5. if (x != 1)
6. Not updating;
7. x--;
8. else
9. Update;
10. $x = \frac{-1}{-3729e^{-G}} + 1;$
11. end
12. end

IV. EVALUATION

A. Experimental Methods

As a feasibility study, a multi-core processor was used to verify the accuracy of the solution search. Each partition is assigned a CPU core, and each core holds individual information on its own. Communication between tasks provides the ability to report solutions in areas of partial overlap.

The constrained knapsack problem described below is used to evaluate two objective optimization problems.

In consideration, The evaluation used in this problem takes time, the average island model has too many design variables, i performed a comparative evaluation of three events: partitioning, extraordinary introduction, and sparsity between execution time.

In the case of standard MOEA/D(conventional virtual overlap method) execution on a single CPU(single-core standard MOEA/D), parallel MOEA/D with standard migration and MOEA/D parallel execution using the proposed method(modified virtual overlap method).

In order to constrain multi-objective optimization problems, i focused on mk-KPs [7]. The mk-KPs are defined in as follows.

$$\begin{cases} \text{Maximize } f_j(x) = \sum_{i=1}^{n} p_{ij} \times x_j (j = 1,2,\ldots,m) \\ \text{Subject to } \sum_{i=1}^{n} w_{il} \times x_i \le c_l (l = 1,2,\ldots,k) \end{cases} \quad (8)$$

This is a 0-1 knapsack problem, in which there are n items and k knapsacks, each item i has m profit $p_{ij}(j = 1,2,\ldots m)$ and k weight w_{il} $(l = 1,2,\ldots,k)$ task is to find a set of x to maximize the total profit m and cannot exceed k knapsack capacityc_l. The knapsacks capacity c_l is defined as follows:

$$c_l = \varphi \sum_{i=1}^{n} w_{il} (l = 1,2,\ldots,k) \quad (9)$$

φ represents the feasibility ratio of each knapsack (constraint), and the strictness of the constraint can be controlled by changing the φ. In this problem, the number of targets m and knapsack k can be determined independently.

The parameters used in the experiment and the execution environment are summarized in table 1 respectively.

Table I. TEST EXECUTION ENVIROMENT

CPU	Intel Core i7-10875H CPU, 2.3 GH
Memory	16.0 GB
OS	64-bit OS, x64-based processor
Compiler	Visual Studio 2019 (ISO C++ 17.0)

B. Experimental Results and Discussion

As an evaluation of the solution searched in the initial phase, figure 8-10 shows a comparison of the proposed method with the 5000 generation comparisons of the single-core standard MOEA/D. To be more specific, when parallelism increases, the total number of individuals is set to 200, using 2-8 cores. When searching for solutions, the single core standard MOEA/D is added to 5000 stage to compare the convergence.

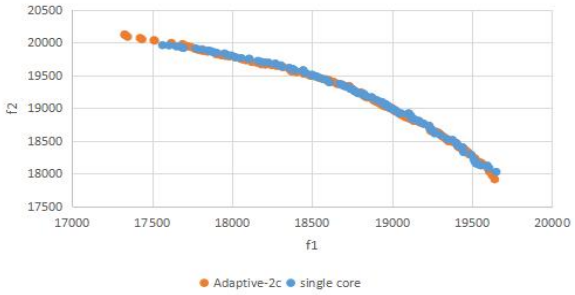

Fig. 8 Adaptive method 2 core and traditional single core results

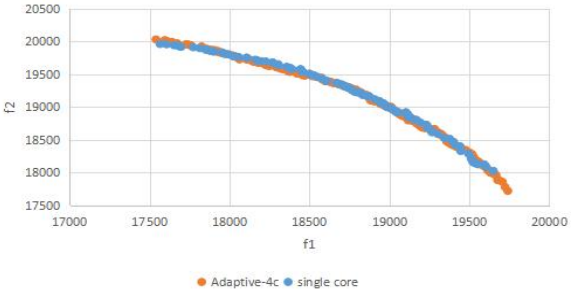

Fig. 9 Adaptive method 4 core and traditional single core results

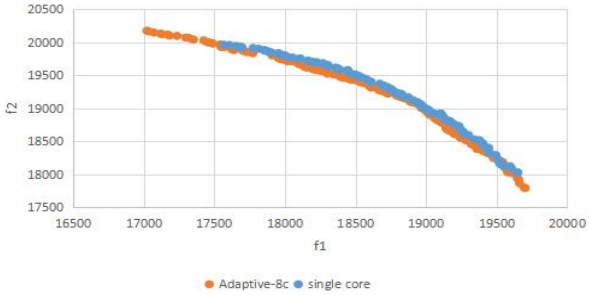

Fig. 10 Adaptive method 8 core and traditional single core results

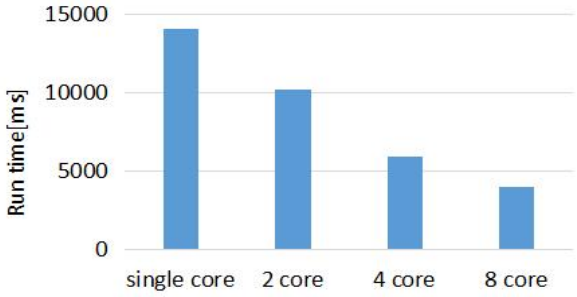

Fig. 11 Run time using adaptive method or single core

According to Figure 8-10, compared to traditional and single-core standard MOEA / D, the proposed way is not inferior to the traditional single-core method in the accuracy performance of the PF solution set, but the proposed method can get the PF solution set faster in terms of the speed of solving the problem. Subsequently, the traditional virtual overlap method was compared with the improved virtual overlap method. Different from the traditional virtual overlap method, the update intervals of reference point Z* and nadir z^{nad} are used, with fixed intervals of 100,200 and 1000.

For about the modified virtual overlap method, it toggles the update interval to determine the progress of the search by considering the solution. The reference points and lowest points show an improvement in sparsity between partitions during the initial phase of solution searching. Similar trends are used to show this contribution when other core assessments are used.

According to the comparison chart below, it can be clearly seen that the PF obtained by the method proposed in this paper is more diverse than the traditional results obtained by using fixed intervals of 100 and 200. At the same time, the adaptive method is faster. The previously proposed parallelization technology, the traditional virtual overlap method, performs updates at fixed time intervals, which helps to accelerate. Compared with the execution time of the single-core standard MOEA/D, the proposed technology could be about four times faster.

From Figure 12, i can know that compared with the traditional method constant (100/200), the proposed adaptive method is not only helpful to obtain more uniform solution distribution in early solution search, but also improves the diversity of PF and improves the HV value. In addition, i can know from Figure 13 that the execution time of the adaptive method is also shorter. However, compared with the constant (1000), the adaptive method has better performance only when there are 8 cores. Next, I did a comparative experiment between the results of the constant (1000) method and the adaptive method when the number of generations is different. The test results are shown in Figure 14.

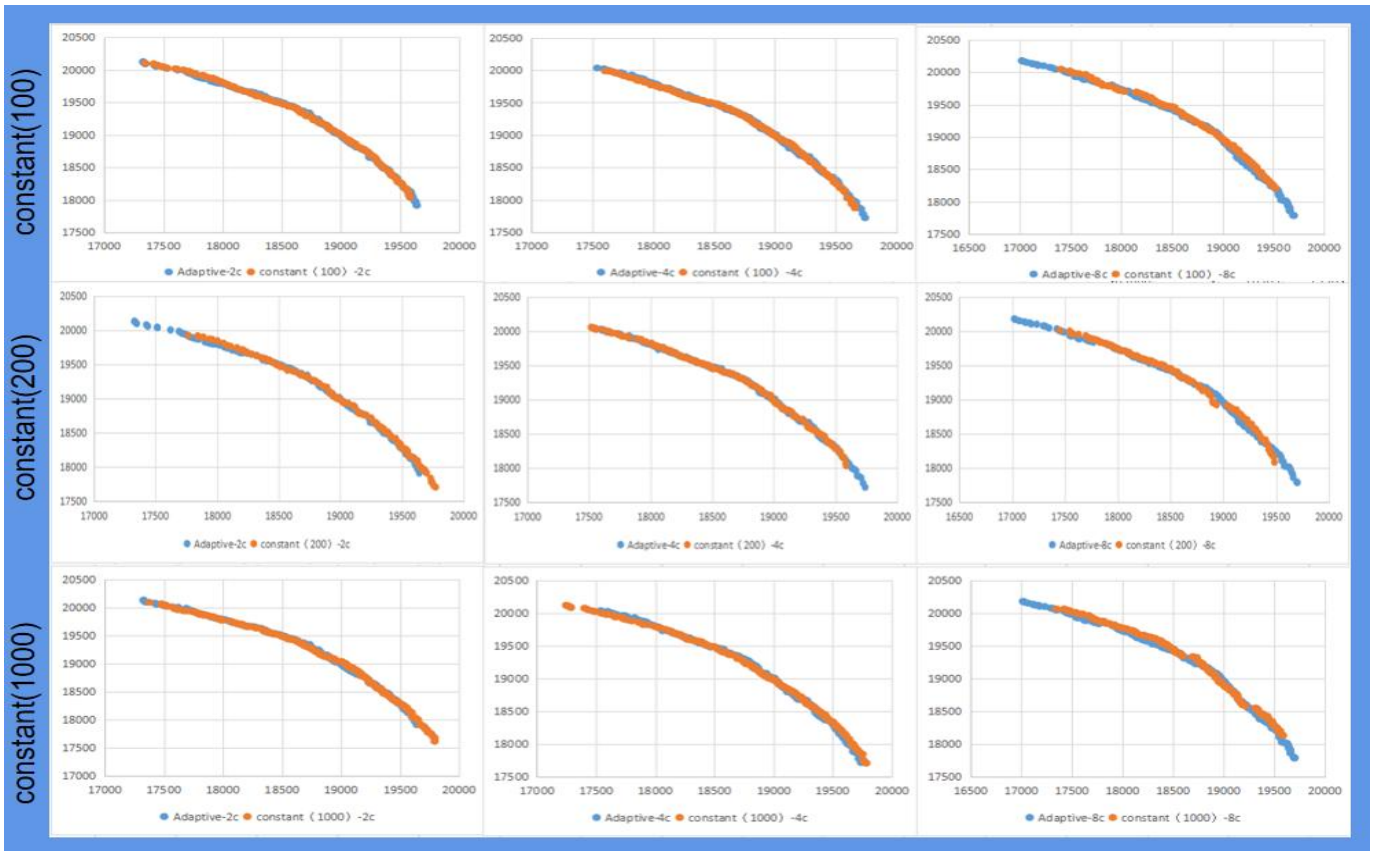

Fig. 12 The comparison between constant (100/200/1000) and adaptive mode Pareto frontier in the case of different core numbers.

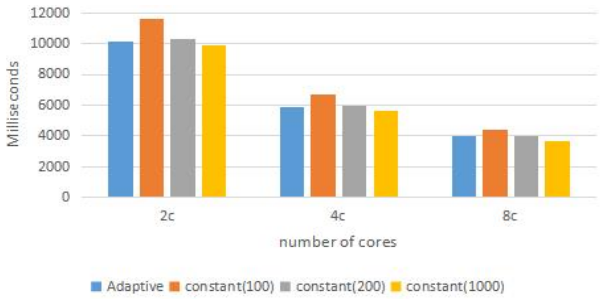

Fig. 13 Run time comparison between adaptive method and constant methods.

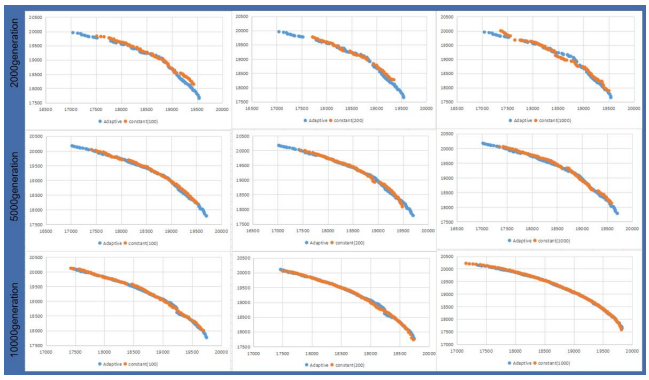

Fig. 14 Pareto frontier comparison in different generation with 8 core.

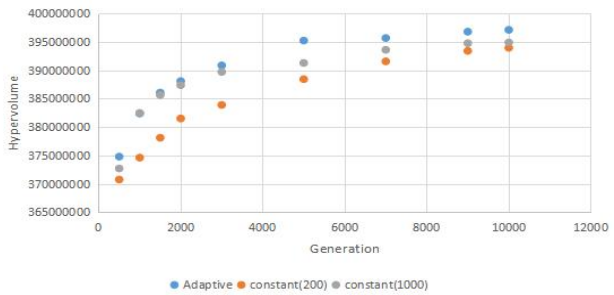

Fig. 15 The relation between HV value and generations in different methods.

Through the experimental results in Figure 14, i find that the adaptive method performs better when the current number is low. In order to verify this result, I continued to do experiments with the number of generations 500/1000/1500/2000/3000/7000/9000 using 8 cores. A comparison chart of the HV values obtained by different generations was made, as shown in Figure 15. In Figure 15, although the HV value of the 2000 generation constant (1000) is very similar to the HV value of the adaptive mode, in Fig. 15 In the Pareto frontier results of 14, i can clearly show that the results of the adaptive method are significantly better. In the 500th and 1000th generations, although neither the adaptive method nor the constant (1000) can get enough effective update points, the results can still be seen that the performance of the adaptive method is still better. Therefore, through the comparison results of Figure 15 and Figure 14, it could be seen that when the current number was low, the performance of the adaptive method was better than that of the constant method.

V. CONCLUSION

This paper proposed a new method to automatically change the ideal point update interval in the already proposed virtual overlap method, targeting MOEA/D, which is one of the effective evolutionary multi-objective optimization algorithms. In the evaluation experiments, I investigated the convergence and diversity of the solution distribution, in the case of dividing into multiple partitions and updating the ideal points at constant migration intervals, and in the case of executing his MOEA/D on a single CPU without partitioning. Using the constrained knapsack problem, two-objective optimization problems were evaluated in terms of performance, and diversity of the Pareto optimal front. As a result, it was shown that the proposed method is effective in improving the diversity of the solution distribution compared to running on a single CPU. In addition, a comparison with the virtual overlap method in which migration is performed at regular intervals shows that the proposed method is more effective in improving the diversity of the solution distribution than the method in which migration is frequently performed at regular intervals. This paper showed that it is more effective in improving convergence and diversity with a smaller number of generations than the interval migration method.

VI. ACKNOWLEDGEMENTS

During the completion of this thesis I would like to thank Prof. Yuji Sato and Prof. Mikiko Sato in particular. Under the guidance of the two professors, I found the appropriate update method and the deficiencies in the thesis writing process. In addition, I also need to thank senior Mads Midtlyng for his patience in helping me solve many problems I encountered in program modification and compilation. Without them I might not have been able to complete this thesis myself, so I would like to express my deep gratitude to them.

REFERENCES

[1] 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, Apr 2002.

[2] 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.

[3] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization,” in Proceedings of the EURO-GEN 2001 Conference, 2001.

[4] Y. Sato, M. Sato, and M. Miyakawa, “Distributed NSGA-II Sharing Extreme Non-dominated Solutions for Improving Accuracy and Achieving Speed-up,” in Proceedings. of 2019 IEEE Congress on Evolutionary Computation (CEC),pp. 3087-3094, 2019.

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

[6] M. Sato, Y. Sato, M. Midtlyng and M. Miyakawa, "Inconstant Update of Reference Point Value for Parallel and Distributed MOEA/D" in Proceedings. of 2021 IEEE Congress on Evolutionary Computation (CEC), pp. 1495-1502,2021.

[7] D. P. Hans Kellerer, Ulrich Pferschy, Knapsack Problems. Springer, 2004.