A Parallel MOEA/D Generating Solutions in Minimum Overlapped Update Ranges of Solutions

Hiroyuki Sato
The University of Electro-Communications
1-5-1 Chofugaoka,
Chofu,Tokyo, 182-8585 Japan
sato@hc.uec.ac.jp

Minami Miyakawa
The University of Electro-Communications
1-5-1 Chofugaoka,
Chofu,Tokyo, 182-8585 Japan
miyakawa@hs.hc.uec.ac.jp

Elizabeth Pérez-Cortés
Universidad Autónoma Metropolitana Iztapalapa
San Rafael Atlixco No. 186,
Col. Vicentina, 09340, México
D.F., MEXICO
pece@xanum.uam.mx

ABSTRACT

This paper proposes a parallel MOEA/D which assigns the computational resources to generate solutions in the minimum overlapped update ranges of solutions. The search performance is verifie on DTLZ2 problem and a car design optimization using TORCS.

Categories and Subject Descriptors

I.2.8 [**Artificia Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

multi-objective optimization, MOEA/D, parallelization

1. INTRODUCTION

Multi-objective EAs (MOEAs) have been intensively studied and successfully applied for a number of real-world multi-objective optimization problems so far. Especially in engineering optimization problems, complicated simulations are needed to evaluate solutions, and it is often time-consuming. To evolve solutions, MOEAs generally need a huge number of solution evaluations. For solving problems involving time-consuming solution evaluations, we need to optimize solutions with a small number of solution evaluations or efficientl evaluate the generated solutions in parallel.

In this work, we focus on representative MOEA/D [1] and propose a method to efficientl evaluate solutions in parallel. The proposed parallel MOEA/D assigns the computational resources to generate solutions in the minimum overlapped update ranges of solutions. Also, the proposed parallel MOEA/D uses a tournament selection based on the scalarizing function value to strengthen the selection pressure of parents.

2. SERIAL MOEA/D

In the conventional serial MOEA/D [1], a multi-objective problem is solved by optimizing a number of single-objective problems define by scalarizing functions g using uniformly distributed weight vectors $\boldsymbol{\lambda}^i$ $(i = 1, 2, \ldots, N)$. Since one solution is assigned to each weight index i, the population size becomes N. Each weight index i has T-nearest neighbor indices $\mathcal{B}_i = \{i_1, i_2, \ldots, i_T\}$, and these indices become the selection range of parents and the update range of solutions for weight index i.

GECCO '15 July 11-15, 2015, Madrid, Spain
© 2015 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764889

Algorithm 1 Find Focus Index with the Minimum Overlap of Neighbors

```
Output: Focused index $F_{idx}$ to select parents
Note: $\boldsymbol{o} = (o^1, o^2, \ldots, o^N)$ is overlap value vector, $\boldsymbol{so} = (so^1, so^2, \ldots, so^N)$ is the vector with the sum of overlaps, and $\boldsymbol{c} = (c^1, c^2, \ldots, c^N)$ is counter vector to be selected as the focused index.
 1: $\boldsymbol{o} = (o^1, o^2, \ldots, o^N) \leftarrow (0, 0, \ldots, 0)$
 2: $\boldsymbol{so} = (so^1, so^2, \ldots, so^N) \leftarrow (0, 0, \ldots, 0)$
 3: for all $i \in \{1, 2, \ldots, N_{\text{CPU}}\}$ do
 4:     if CPU$_i$ is in use for a solution evaluation then
 5:         $j \leftarrow$ Focused index of CPU$_i$
 6:         for all $k \in \mathcal{B}_j$ do
 7:             $o^k \leftarrow o^k + 1$
 8:         end for
 9:     end if
10: end for
11: for all $i \in \{1, 2, \ldots, N\}$ do
12:     for all $k \in \mathcal{B}_i$ do
13:         $so^i \leftarrow so^i + o^k$
14:     end for
15: end for
16: $\mathcal{C}_1 \leftarrow$ Find indices with the minimum $so$ from $\{1, 2, \ldots N\}$
17: $\mathcal{C}_2 \leftarrow$ Find indices with the minimum counter $c$ from $\mathcal{C}_1$
18: $F_{idx} \leftarrow$ Find the minimum index from $\mathcal{C}_2$
19: $c^{F_{idx}} \leftarrow c^{F_{idx}} + 1$
20: return $F_{idx}$
```

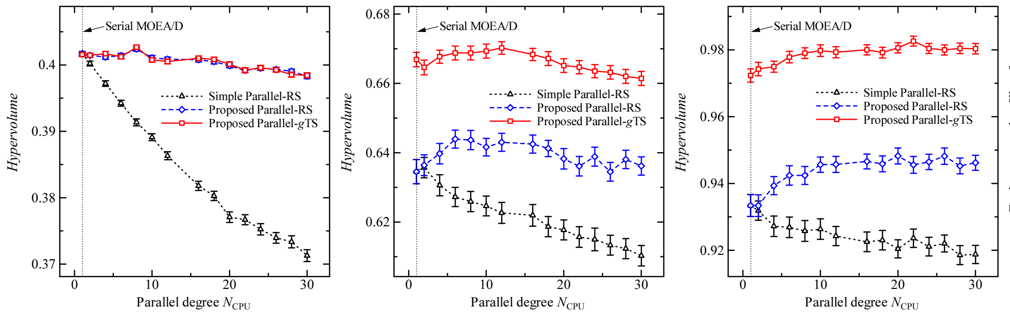
To generate one offspring, MOEA/D focuses on a weight index (*focused index*), selects parents from neighbors of the focused index, generates offspring, and updates solutions in neighbors of the focused index by the generated offspring. This process is repeated during the solutions search. Therefore, a good offspring showing better g than current solutions can become a parent immediately after it is generated. NSGA based algorithms do not have this mechanism, it contributes to enhancing the solution search of MOEA/D in several problems because good solutions are actively utilized as parents. To efficientl parallelize the evaluation of solutions in MOEA/D, it is important to maintain this advantage of MOEA/D.

3. PROPOSED PARALLEL MOEA/D

We propose a parallel MOEA/D to effectively parallelize the evaluation of solutions. When we have N_{CPU} CPUs, the proposed algorithm focuses N_{CPU} different weight indices with the minimum overlap of neighbors, generates and evaluates offspring in parallel.

3.1 Minimum Overlap Focused Index Selection

The conventional serial MOEA/D sequentially changes the focused index in the order of $1, 2, \ldots, N$. If the focused index is sequentially changed also in the parallel MOEA/D, several focused indices having overlapped neighbors (update range of solutions) are selected for a parallel evaluation of solutions. In this case, if good



(a) $m = 2$ objectives (b) $m = 3$ objectives (c) $m = 4$ objectives

Figure 1: Results of HV by varying the parallel degree N_{CPU} on DTLZ2 problems

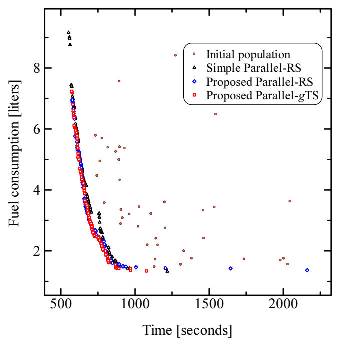


Figure 2: Obtained sets of car designs

Table 1: Three algorithms

	Focused index selection	Parent selection
Simple Parallel-RS	Sequential [1]	Random [1]
Proposed Parallel-RS	Minimum Overlap	Random [1]
Proposed Parallel-gTS	Minimum Overlap	g-Tournament

offspring are obtained around each focused index, these offspring cannot be utilized as parents at the parallel offspring generation. Thus, in the parallel evaluation of solutions, the sequential selection of the focused index cannot take the aforementioned advantage of MOEA/D. To avoid this problem, the proposed parallel MOEA/D selects focused indices with the minimum overlap of neighbors. **Algorithm 1** is used to fin such indices.

3.2 Two Options to Select Parents

After a focused index is determined, the conventional MOEA/D randomly selects two parents from neighbors of the focused index [1]. To improve the selection pressure for the search direction given by the weight vector of the focused index, we use the g-tournament selection as another option to select parents. In this method, firs two candidate solutions are randomly chosen from the neighbors of the focused index. Then, the scalarizing function values g of the two solutions for the weight vector of the focused index are compared, and the solution showing better g becomes a parent. The other parent is selected in the same way.

4. RESULTS AND DISCUSSION

In this work, we compare the three algorithms shown in **Table 1**.

4.1 Results on DTLZ2 Function Optimization

First we use DTLZ2 problems with $m = \{2, 3, 4\}$ objectives and $n = 30$ variables. DTLZ2 is a function optimization problem and each solution evaluation is not time-consuming. The purpose to use DTLZ2 is only to verify the search performances of the parallel MOEA/Ds. The population size is set to $N = \{200, 210, 455\}$ for $m = \{2, 3, 4\}$ objectives, respectively. The neighbors' size is set to $T = 20$. To generate offspring, SBX ($\eta_c = 20$ and the ratio 0.8) and PBM ($\eta_m = 20$ and the ratio $1/n$) are used. The termination criterion is set to $50N$ solution evaluations. We use the $Hypervolume$ (HV) with the reference point $\boldsymbol{r} = (1.1, \ldots, 1.1)$ as the metric. The plotted values of HV are the average of 300 independent runs. As usual, higher HV values denote better performance. In this experiment, we use a computer with Intel Xeon E7-4870 (80 cores) operated by RedHat Enterprise Linux 6.

Fig. 1 shows results of HV obtained by the three algorithms as the parallel degree N_{CPU} is increased. The error bars indicate 95% confidenc intervals. From the results of the simple parallel MOEA/D, we can see that HV is deteriorated as the parallel degree N_{CPU} increases. That is, the sequential selection of the focused index is less efficien in the parallel MOEA/D framework. In **Fig. 1 (a)**, although values of HV obtained by the two proposed parallel MOEA/Ds are slightly decreased in $m = 2$ objective problem, these algorithms maintain high HV even when the parallel degree N_{CPU} is increased. In **Fig. 1 (b)** and **(c)**, we can see that the proposed parallel MOEA/D-RS with parallelization ($N_{\mathrm{CPU}} > 1$) achieves higher HV than the one without parallelization ($N_{\mathrm{CPU}} = 1$). These results reveal that the proposed minimum overlap selection is an efficien way to select the focused index in the parallel MOEA/D framework. Next, we compare the two proposed parallel MOEA/Ds-RS and -gTS. Although their values of HV are close in $m = 2$ objective problem, HV is increased by using gTS in problems with $m = \{3, 4\}$ objectives. These results reveal that the g-tournament parent selection improves the search performance of the parallel MOEA/D in problems with large m.

4.2 Results on Car Design Optimization

We also address a racing car design optimization using the car simulator TORCS [2]. We optimize 22 kinds of design variables including the gear ratio, the strength of suspension, etc. to minimize the total time of a race and the fuel consumption. TORCS generates a car based on given design variables. The computer drives the car on Alpine 2 course build in TORCS f ve times, and we obtain the total time of the race [seconds] and the fuel consumption [liters] as the evaluation values. In this experiment, we use a computer with Intel Core i7-3770 (8 threads) operated by Windows 7. We use the same parameters specifie in the previous section.

Fig. 2 shows sets of non-dominated solutions obtained by the three algorithms with $N_{\mathrm{CPU}} = 8$. From the result, we can see that the convergence of solutions obtained by the two proposed parallel MOEA/Ds in the central area of Pareto front is higher than the simple parallel one. Also, the proposed parallel MOEA/D-gTS shows higher convergence in the central area of Pareto front than the one with RS. This results also show the effectiveness of the proposed minimum overlap selection and the tournament selection in the parallel MOEA/D.

5. CONCLUSIONS

This work proposed an efficien parallel MOEA/D using the minimum overlap selection of focused index and the tournament parent selection based on the scalarizing value g. The experiments using DTLZ2 function optimization problem and the car design optimization using TORCS showed the effectiveness of the minimum overlap selection of focused index and the g-tournament selection.

As future work, we will address to design the automatic control of the parallel degree during the solution search.

6. REFERENCES

[1] Q. Zhang and H. Li, "MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. on EC*, Vol.11, No. 6, pp.712–731, 2007.

[2] TORCS - The Open Racing Car Simulator, http://torcs.org/.