Investigating the Effect of Parallelism in Decomposition Based Evolutionary Many-Objective Optimization Algorithms

Lei Chen[1,2], Kalyanmoy Deb[2], and Hai-Lin Liu[1]

[1] Guangdong University of Technology, Guangzhou, China
chenaction@126.com, hlliu@gdut.edu.cn.
[2] Michigan State University, East Lansing, MI 48824, USA
kdeb@egr.msu.edu, http://www.coin-laboratory.com

COIN Report Number 2018004

Abstract. One of the main reasons for evolutionary multi-objective and many-objective optimization (EMO) algorithms to find and maintain multiple trade-off solutions is that their operators are capable of establishing an implicit search parallely to multiple regions of the search space. A recent direction in EMO algorithm development is to use decomposition-based methods which make a compromise on achieving the full advantage of the implicit parallelism aspect of evolutionary algorithms. In this paper, we investigate a specific decomposition-based algorithm, MOEA/D-M2M, and investigate the effect of an user-controlled explicit parallelism method in its search operators. For this purpose, we apply a number of M2M variants on a number of standard many-objective test problems (DTLZ and WFG problems) and compare their performances with more implicitly parallel EMO algorithms – NSGA-III and MOEA/D. Results from our extensive study indicate that by relaxing the decomposition effect, thereby re-establishing parallel search within M2M operators, the performance of the resulting MOEA/D-M2M variants can be improved. Despite our study being restricted to a single EMO algorithm, it provides interesting insights to the working of EMO algorithms in solving many-objective optimization problems.

Keywords: Implicit parallelism · Decomposition · Evolutionary algorithms · Many-objective optimization.

1 Introduction

Ever since the demonstration of evolutionary algorithms (EAs) to solve multiple conflicting objectives to find multiple Pareto-optimal solutions simultaneously in early nineties, EAs have emerged as popular means for solving multi- and many-objective optimization problems. Unlike single-objective optimization problems, multi-objective optimization problems (MOPs) give rise to multiple trade-off optimal solutions, which must be found as a representative set before a single preferred solution is chosen. Evolutionary multi-objective and many-objective

optimization (EMO) algorithms constitute a population based search such that their operators conduct a parallel search in different areas of the search space at the same time. This kind of implicit parallelism among EMO population members makes it possible to obtain a set of Pareto-optimal solutions for an MOP in a single run. For this reason, population based EMO algorithms are more popular than classical methods in dealing with MOPs. The past few decades have seen the dramatic increase in the number of evolutionary algorithms for multi- and many-objective optimization.

For the first time since evolutionary algorithm was utilized for MOPs by by Schaffer in 1980s [2], the development of evolutionary multi-objective (EMO) algorithms has been greatly improved. The Non-dominated Sorting Genetic Algorithm (NSGA) and its successor NSGA-II used the non-dominated sorting to classify a population into several non-dominated layers [3]. By providing uniform selection pressure to the entire non-dominated set of solutions, a parallel search was established. Additional pressure was introduced by its crowding distance operator to search for better extremes and filling the gaps between non-dominated solutions, as and when needed. The Strength Pareto Evolutionary Algorithm (SPEA) and the following SPEA-II used an external archive containing non-dominated solutions [4]. The Pareto Archived Evolution Strategy (PAES) used a (1+1) evolution selection strategy together with an external archive to record all the non-dominated solutions. The Pareto Envelope-based Selection Algorithm (PESA) and PESA-II evolves a small internal population and a larger external population [5] by emphasizing non-dominated and more diverse solutions. All these early methods allowed a population-wide search and applied their recombination operators on the entire population without any restriction, thereby introducing a flexible and parallel search.

The above-mentioned EMO methods for multi-objective optimization were all based on the Pareto dominance, and they have achieved a great success in dealing with multi-objective optimization problems with two or three objectives. However, when it came to solving MOPs with more than three objectives, i.e., many-objective optimization problems (MaOPs), these methods encountered difficulties for the so-called dominance resistance phenomenon [6]. This led to the development of new EMO algorithms following the idea of *decomposition* of the search process. Decomposition0-based EMO algorithms have been reported to be more advantageous than the Pareto and indicator-based EMO algorithms in dealing with MaOPs [7, 8]. In decomposition based EMO algorithms, a set of predefined decomposition vectors is predefined, either for objective aggregation [9–11] or for diversity and convergence enhancement [12–15]. This way, despite the large dimensionality of the objective space, the search is restricted within an island of solutions in the variable space dictated by a decomposition vector. Although such decomposition enabled to negotiate the challenges of dimensionality increase, the proposed EMO algorithms vary in the way the extent of decomposition has been introduced. One extreme with a zero parallelism is the classical "generating methods" [16], which attempts to find a single Pareto-optimal solution in a single optimization task. In the context of EMO methods,

MSOPS (multiple single-objective Pareto sampling) method ranks the individuals according to the vector angle distance scaling and the weighted Tchebycheff aggregation [11]. MOEA/D (multi-objective evolutionary algorithm based on decomposition) decomposes an MOP into a number of single-objective optimization subproblems using objective aggregation method. NSGA-III [12] is the third generation non-dominated sorting genetic algorithm, but it uses a decomposition-based niching method to enhance the convergence and diversity. MOEA/D-M2M [17] is a new variant of MOEA/D for population decomposition, and it decomposes a many-objective optimization problems into a set of many-objective optimization subproblems. The parallelism that can be exploited from such algorithms depends on how each subproblem search has been interlinked and parallelized to each other.

The above discussion raises an important distinction between the well-known "implicit" parallelism associated with a population-based EA and a possible "explicit" parallelism which can be introduced by the algorithm designer. The implicit parallelism comes from the effect of a selecto-recombination search operator applied to the entire population of solutions. Depending the "designed" extent of parallelism allowed to the operators, a highly fit population member can be recombined with another such population from two different parts of the search space and produce a child having good characteristics of both parents. In problems where such a child would be evaluated to be fitter than its parents, such implicitly parallel evolutionary algorithms will flourish and perform well. The designed aspect of introducing parallelism to an algorithm is called here as the explicit parallelism. The developer of an algorithm restricts the implicit parallelism to be extended to subpopulation, thereby controlling the extent of implicit parallelism can be imposed to the search process. Figure 1 presents a sketch explaining the extent of explicit parallelism introduced by different multi- and many-objective optimization algorithms.

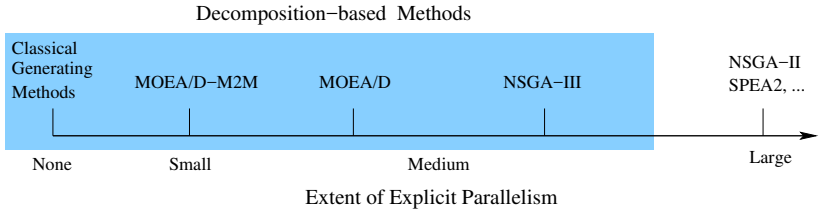

Fig. 1. Extent of explicit parallelism in different decomposition-based multi- and many-objective optimization algorithms.

In this paper, we use MOEA/D-M2M as a representative algorithm to investigate how the explicit parallelism affects the performance of decomposition based EMO algorithms. To be specific, we modify the original MOEA/D-M2M to introduce different degrees of parallelism, and then conduct extensive experiments of the variants of MOEA/D-M2M, NSGA-III and MOEA/D on two sets of widely used benchmark problems. Performance comparison and parallelism

analysis are done to show that the degree of parallelism makes a large effect on the performance of decomposition-based EMOs, and a better performance of MOEA/D-M2M can be achieved by properly adjusting the degree of explicit parallelism. The explicit parallelism studies conducted in this paper can help to facilitate the design and practical application of decomposition-based EMO algorithms.

The remainder of paper is organized as follows: Section II describes the basic idea of implicit and explicit parallelism involved in EMO algorithms. In Section III, experiments are conducted, and the effect of parallelism is demonstrated through simulation results and analyzed. Section IV concludes this paper by proposing a number of promising issues for future studies.

2 Two Modes of Parallelism

In this section, we introduce explanations of parallelism related to evolutionary algorithms in detail. As we have mentioned, there are two modes of parallelism involved in decomposition-based EMO algorithms. The first is the well-known implicit parallelism, which is an inherent quality of all population-based evolutionary algorithms. The second aspect is an explicit parallelism mechanism, which can be introduced explicitly by the algorithm developer in the hope of controlling the implicit parallelism which can be incorporated through an explicit decomposition of the search process. An efficient optimization algorithm needs to make a good balance of these two parallelism modes in which one can be explicitly controlled by the developer and the other gets established accordingly in the ensuing algorithm. These two modes of parallelism are explained more in the following:

- **Implicit parallelism:** The parallelism comes from operator interactions in population based search and the algorithm itself. When two solutions are recombined, certain schemata are preserved between parents and offsprings. Selection, niching, recombination, mating restriction, and any other genetic operators employ such a parallelism within an evolutionary algorithm. The degree of implicit parallelism gets established by the ability and flexibility of population members to be integrated in creating new solutions. The ability of one population member to influence another population member in finding new and better child solutions is referred as the implicit parallelism of the search algorithm.
- **Explicit parallelism:** Parallelism is introduced explicitly by directly controlling how two or more individuals can participate in an evolving population in creating new solutions. Such an external control can reduce the generic search effect introduced by the implicit parallelism to make a focused search, if and when needed. In problems where an implicit parallelism effect between two disparate solutions can result in not-so-good solutions, an explicit control of parallelism can be beneficial. In a sense, an explicit parallelism controls the effect of implicit parallelism in order to constitute a better search algorithm.

In the context of multi- and many-objective optimization, implicit parallelism is actually an inner feature of EMO algorithms, and the nature of population-based search makes it possible to approximate a set of optimal solutions simultaneously. Explicit parallelism can be introduced by the recombination, selection and migration procedures of evolutionary algorithms. In decomposition-based EMO algorithms, the population is divided into different subpopulations and they can either evolve independently or collaboratively within an algorithm. A coupling can be used to roughly describe the degree of explicit parallelism. For example, in MOEA/D-M2M, the population is decomposed into a number of subpopulations. If the explicit parallelism is too restrictive, each subpopulation is expected to evolve independently with implicit parallelism is restricted only to one subproblem. We can easily relax the explicit parallelism of MOEA/D-M2M by introducing a certain kind of interactions among subpopulations. MOEA/D-M2M framework allows us to control the effect of explicit parallelism by relaxing the original restrictions of its operators. NSGA-III has two different selection operators: (i) Non-domination based selection, which is a key operation in introducing implicit parallelism, and (ii) Niching based selection, which can be controlled explicitly to act along each reference vector independently or as a whole. In MOEA/D, the selection procedure is confined within a neighborhood of a reference vector, thereby making the process explicitly parallel. Overall, we observe that the above three popular decomposition-based EMO algorithms have similar different extents of explicit parallelism, but importantly they can be controlled to vary their overall search power.

Table 1 shows the coupling of MOEA/D-M2M, NSGA-III, and MOEA/D in detail. In this table, we can intuitively explain the parallelism by means of selection, recombination, and migration procedures. To be specific, the selection and recombination of original MOEA/D-M2M are restricted to take place independently within each subpopulation, but a migration of a solution created by one subproblem is allowed to place the new solution in its own subproblem. The non-dominated sorting process in NSGA-III is applied to the whole population, whereas the niching procedure compares solutions within the same niche (or subpopulation). While the second part allows a controlled explicit parallelism, we do not study its effect in this paper. MOEA/D conducts neighbor-based (or subpopulation-based) selection and recombination, but the newly generated offsprings do not have to compared with its competitors within the same neighborhood. Due to the lack of migration, an offspring is only used to update the subproblems within a specified neighborhood. When viewed through the lens of explicit and implicit parallelism modes discussed above, each of these EMO algorithms must ideally be investigated for their control of explicit parallelism introducing operators.

2.1 Variants of MOEA/D-M2M

In this paper, we restrict our study to MOEA/D-M2M algorithm only. Because of the population decomposition strategy in MOEA/D-M2M, we can easily control the degree of explicit parallelism by varying the interactions among different

Table 1. Illustration of the coupling in MOEA/D-M2M, NSGA-III, and MOEA/D.

1	MOEA/D-M2M	NSGA-III	MOEA/D
Selection	Subpopulation	Yes	Neighbor
Recombination	Subpopulation	Yes	Neighbor
Migration	Yes	Yes	No

subpopulations. Therefore, we vary the probabilities of selection and recombination within each subpopulation of the original MOEA/D-M2M to make it have different degrees of parallelism, and then compare the performance of MOEA/D-M2M variants with NSGA-III and MOEA/D algorithms. There are totally 14 MOEA/D-M2M variants including the original one (Table 2). For simplicity, we use $v_1, \ldots, v_{14}$ and their corresponding (P_1, P_2) to represent the 14 variants of MOEA/D-M2M, where P_1 and P_2 indicate the probability of selection within the subpopulation and the probability of crossover within the subpopulation. For example, MOEA/D-M2M(1,1) represents the original MOEA/D-M2M, where both the selection and recombination are restricted within each subpopulation. MOEA/D-M2M(1,0) allows recombination to take place among the entire population. MOEA/D-M2M(0.67,0.67) restricts selection and recombination within a subpopulation 67% of the time and applies the operators among the entire population (including other subpopulations) 33% of the time. MOEA/D-M2M(0,0) is the most parallel algorithm in which any two population members can be compared within the selection operator and any two members can participate in its recombination operator.

Table 2. Variants of MOEA/D-M2M and their respective parameter values.

Variant	v-1	v-2	v-3	v-4	v-5	v-6	v-7
(P_1,P_2)	(1,1)	(1,0.67)	(1,0.33))	(1,0)	(0.67,0.67)	(0.67,0.33)	(0.67,0)
Variant	v-8	v-9	v-10	v-11	v-12	v-13	v-14
(P_1,P_2)	(0.33,0.67)	(0.33,0.33)	(0.33,0)	(0,0)	(0,0.33)	(0,0.67)	(0,1)

3 Experimental Studies and Results

In this section, we present results from an extensive experimental study to demonstrate the effect of the explicit parallelism on the performance of decomposition-based MOEA/D-M2M algorithm. Parallelism can be introduced in selection and crossover operators either implicitly or explicitly. In this paper, we study the explicit parallelism introduced by selection and crossover only. Mutation operator is applied as usual. In our experimental study, two sets of widely used benchmark problems from DTLZ [22] and WFG [21] families with three to 10 objectives are tested. These two kinds of benchmark problems are very popular among the EMO community, because they cover a number of problem characteristics that may bring challenges for evolutionary algorithms to obtain a set of well representative solutions along the Pareto Front. We study a total

of 16 experimental algorithms including 14 variants of MOEA/D-M2M. Each of them is run 15 times independently for each test problem. The IGD-metric [24] is used to measure the quality of obtained results for each test instance. Due to the space limitation, the experimental results are shown by means of Wilcoxon Rank-Sum Test.

3.1 Parameter Settings

In the experimental studies, the population size N and divisions H for generating initial decomposition vectors for all the experimental algorithms are kept the same and shown in Table 3. Problems DTLZ1-DTLZ4 and WFG5-WFG8 with the number of objectives $m = 3$, 5, 8 and 10 are tested here. The number of decision variables is set as $n = m + 4$ for DTLZ1, $n = m + 9$ for DTLZ2-4, and $n = m + 19$ for WFG5-8. The number of generations for DTLZ1-4 and WFG5-8 are shown in Table 4. The SBX [20] operator with $p_c = 1$ and $\eta_c = 30$, and polynomial mutation [1] with $p_m = 1/n$ and $\eta_m = 20$ are used in each algorithm.

Table 3. Population size and number of divisions in each objective axis for generating decomposition vectors.

♯Obj.	♯Div.	PopSize
3	12	91
5	6	210
8	3,2	156
10	3,2	275

Table 4. Number of generations for DTLZ1-4 and WFG5-8 problems.

♯Obj.	DTLZ1	DTLZ2	DTLZ3	DTLZ4	WFG5-8
3	400	250	1000	600	400
5	600	350	1000	1000	750
8	750	500	1000	1250	1500
10	1000	750	1500	2000	2000

Note that ♯Div. values in Table 3 for 8 and 10 objectives, indicate them for two layers.

3.2 Results

In this subsection, we make performance comparison and analyze the variants of MOEA/D-M2M with NSGA-III and MOEA/D on DTLZ1-4 and WFG5-8 with 3, 5, 8, and 10-objective versions. For each test instance, we first find the best performance of all algorithms according to the mean IGD-metric values, and then compare the best with each of the rest by using the Wilcoxon Rank-Sum test to see if they are significantly different from each other. Since the performance of an algorithm can be different for relatively low and high number of objectives, we group 3 and 5-objective problems in one class and 8 and 10-objective problems as another class.

Simulation Results on DTLZ Problems Median IGD values on DTLZ benchmark problems are presented in Table 5 for one of the M2M variants (v-2) and other original algorithms. It is interesting to note that some M2M variants (v-2 and v-3, v-12, for example) perform much better than the original M2M algorithm.

Table 5. IGD values MOEA/D-M2M v-2 and other methods on DTLZ problems.

Method	DTLZ1-3	DTLZ1-5	DTLZ2-3	DTLZ2-5	DTLZ3-3	DTLZ3-5	DTLZ4-3	DTLZ4-5
NSGA-III	0.00245	**0.00337**	0.00188	0.00689	0.00446	0.00992	0.00084	0.00118
MOEA/D	0.00153	0.00049	**0.00054**	**0.00092**	0.00430	**0.00182**	0.00013	**0.00008**
M2M	0.00705	0.00101	0.00108	0.00186	0.03076	0.01758	0.00013	0.00035
M2M-v2	**0.00113**	0.00039	0.00064	0.00155	**0.00305**	0.00232	**0.00011**	0.00025
Method	DTLZ1-8	DTLZ1-10	DTLZ2-8	DTLZ2-10	DTLZ3-8	DTLZ3-10	DTLZ4-8	DTLZ4-10
NSGA-III	0.00644	0.00401	0.00217	0.01762	0.03420	0.01954	0.00418	0.00480
MOEA/D	0.00395	0.00512	**0.00204**	**0.00288**	**0.00868**	**0.00394**	0.22141	**0.00117**
M2M	0.01144	0.01151	0.00769	0.00690	0.03105	0.01365	0.00452	0.00439
M2M-v2	**0.00367**	**0.00359**	0.00576	0.00523	0.00887	0.00516	**0.00253**	0.00225

Table 6. Wilcoxon Rank-Sum test score for the variants of MOEA/D-M2M, NSGA-III, and MOEA/D on 3 and 5-objective DTLZ problems.

Instance	v-1	v-2	v-3	v-4	v-5	v-6	v-7	v-8	v-9	v-10	v-11	v-12	v-13	v-14	NSGA-III	MOEA/D
DTLZ1-3	1	0	0	0	0	**0**	0	0	0	0	0	0	0	1	1	0
DTLZ1-5	1	0	**0**	0	1	1	1	1	1	1	1	1	0	1	1	0
DTLZ2-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
DTLZ2-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
DTLZ3-3	1	0	0	**0**	1	0	0	0	1	1	1	0	1	1	1	0
DTLZ3-5	1	0	**0**	0	1	1	1	1	1	1	1	1	1	1	1	0
DTLZ4-3	1	**0**	0	1	0	1	1	1	1	1	1	0	0	1	1	1
DTLZ4-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
Sum	8	3	3	4	6	6	6	6	7	7	7	5	5	8	8	1

Table 7. Wilcoxon Rank-Sum test score for the variants of MOEA/D-M2M, NSGA-III, and MOEA/D on 8 and 10-objective DTLZ problems.

Instance	v-1	v-2	v-3	v-4	v-5	v-6	v-7	v-8	v-9	v-10	v-11	v-12	v-13	v-14	NSGA-III	MOEA/D
DTLZ1-8	1	1	0	0	1	1	1	1	1	1	**0**	0	1	1	1	1
DTLZ1-10	1	1	1	1	1	1	1	1	1	1	1	**0**	1	1	1	1
DTLZ2-8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
DTLZ2-10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
DTLZ3-8	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	**0**
DTLZ3-10	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
DTLZ4-8	1	0	1	1	1	1	1	1	1	1	0	**0**	0	1	1	0
DTLZ4-10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
Sum	7	4	5	6	7	7	7	7	7	7	5	4	6	7	7	2

Tables 6 and 7 show the Wilcoxon Rank-Sum test results of the 15 sets of obtained solutions for each algorithm on DTLZ series test problems. The best performances for each test instance of all the algorithms in terms of the median IGD values of the 15 independent runs are shown in bold. In these tables, a score of one for an algorithm for a problem means that the algorithm is significantly outperformed by the best performing algorithm on the same problem.

A zero score means that the algorithm performs similar (with 95% confidence) to the best performing algorithm on the same problem. The final row calculates the sum of all Wilcoxon scores on all problems. Thus, a smaller sum-score is considered to have performed better, in general. For DTLZ1-4 problems with 3 and 5 objectives, we observe from Table 6 that the M2M variants 2 and 3 achieve the best performance among all variants. Also, their performances are much better than the original MOEA/D-M2M (variant 1) and NSGA-III. However, the performances of variants 2 and 3 are comparable to that of MOEA/D. Similarly, we also observe from Table 7 that variants 2 and 12 achieve the best performance among all M2M variants for DTLZ problems having 8 and 10 objectives. While it is much better than the original MOEA/D-M2M (variant 1) and NSGA-III, its performance is almost comparable to MOEA/D. MOEA/D-M2M variant 2 employs a restricted selection within each subpopulation, whereas the recombination is allowed to take place between different subpopulations with a probability of 33%. Allowing recombination to take place among different subpopulations, implicit parallelism works better in creating useful child solutions and the 33% probability event makes a good balance between exploiting parallelism and negotiating generality needed to solve DTLZ problems, in general. It is also interesting to note that variant 12 (with no restriction on selection and recombination probability of 33%, meaning 33% recombination among different subpopulations) also performs well on low and high-dimensional DTLZ problems.

Simulation Results on WFG Problems One distinction between DTLZ and WFG problems is that WFG problems have a non-uniform scaling of objectives and more complexities. Table 8 shows the IGD values of M2M variant 2 and other original methods on WFG problems. In most problems, M2M v-2 performs better than the original M2M. Wilcoxon scores on WFG problems are presented

Table 8. Mean IGD values of original MOEA/D-M2M, MOEA/D, NSGA-III and the MOEA/D-M2M variant 2 on WFG problems.

Method	WFG5-3	WFG5-5	WFG6-3	WFG6-5	WFG7-3	WFG7-5	WFG8-3	WFG8-5
NSGA-III	**0.03031**	**0.03227**	**0.03255**	**0.03434**	**0.00770**	**0.00910**	0.05567	0.08832
MOEA/D	0.03699	0.03248	0.04715	0.03439	0.03809	0.00938	**0.05245**	**0.06503**
M2M	0.03931	0.03277	0.05627	0.03598	0.05321	0.01284	0.06752	0.06808
M2M-v2	0.03659	0.03288	0.04913	0.03657	0.02871	0.00919	0.07278	0.07385
Method	WFG5-8	WFG5-10	WFG6-8	WFG6-10	WFG7-8	WFG7-10	WFG8-8	WFG8-10
NSGA-III	**0.03178**	0.03326	**0.03018**	**0.03384**	**0.00836**	0.01179	0.17850	0.23475
MOEA/D	0.03686	**0.03271**	0.03793	0.03403	0.01206	**0.00864**	**0.10324**	**0.12373**
M2M	0.03802	0.03612	0.04290	0.03654	0.01964	0.01372	0.11026	0.12909
M2M-v2	0.03751	0.03581	0.04116	0.03761	0.01381	0.01160	0.10962	0.12837

in Tables 9 and 10 for low and high-dimensional problems. Wilcoxon Rank-Sum test results of 15 sets of obtained solutions for each algorithm are presented as before. For WFG5-8 problems with 3 and 5 objectives, we observe from Table 9

Table 9. Wilcoxon Rank-Sum test score for the variants of MOEA/D-M2M, NSGA-III, and MOEA/D on 3 and 5-objective WFG problems.

Instance	v-1	v-2	v-3	v-4	v-5	v-6	v-7	v-8	v-9	v-10	v-11	v-12	v-13	v-14	NSGA-III	MOEA/D
WFG5-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**	1
WFG5-5	1	1	1	1	1	1	1	1	1	1	1	1	0	0	**0**	0
WFG6-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**	1
WFG6-5	1	1	1	1	1	1	1	1	1	1	1	0	1	1	**0**	0
WFG7-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**	1
WFG7-5	1	1	1	1	1	1	1	1	1	1	0	0	0	1	**0**	0
WFG8-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	**0**
WFG8-5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
sum	8	8	8	8	8	8	8	8	8	8	7	6	6	7	1	3

Table 10. Wilcoxon Rank-Sum test score for the variants of MOEA/D-M2M, NSGA-III, and MOEA/D on 8 and 10-objective WFG problems.

Instance	v-1	v-2	v-3	v-4	v-5	v-6	v-7	v-8	v-9	v-10	v-11	v-12	v-13	v-14	NSGA-III	MOEA/D
WFG5-8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**	1
WFG5-10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	**0**
WFG6-8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**	1
WFG6-10	0	1	0	1	1	1	1	1	1	1	1	1	1	1	**0**	1
WFG7-8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**	1
WFG7-10	1	0	0	0	1	1	1	1	1	1	1	1	1	1	**0**	1
WFG8-8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
WFG8-10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
sum	7	7	6	7	8	8	8	8	8	8	8	8	8	8	2	5

that the variants 12 and 13 have achieved the best performance among all M2M variants, and its performance is better than the original MOEA/D-M2M (variant 1). For high-dimensional problem (Table 10), variant 2 performs better than the original M2M algorithm. It is also interesting to note that in WFG problems, NSGA-III outperforms all other algorithms, including MOEA/D. This is due to the lack of a suitable normalization procedure in MOEA/D and its M2M variants, which we discuss further in the next section.

4 Conclusions and Future Studies

In this paper, we have performed an extensive investigation of introducing an explicit parallelism with a decomposition-based EMO algorithm. Variants of MOEA/D-M2M with different degrees of parallelism in its selection and recombination operators are proposed and evaluated. Two other representative decomposition-0based EMO algorithms – NSGA-III and MOEA/D – are also used as comparison algorithms. Both NSGA-III and MOEA/D have been reported to be very effective in multi-objective/many-objective optimization problems and they exhibit a fixed explicit parallelism pattern in their own ways. Following conclusions can be drawn from our extensive simulation studies on the chosen DTLZ and WFG problems:

- Most variants of M2M approach have performed better than the original M2M with restricted selection and recombination operations within each subpopulation. This means that a more relaxed parallelism is found beneficial.
- A more relaxed recombination introducing an extended parallelism in creating new solutions is found to be more important than relaxing parallelism by the selection operator in comparing parent solutions.

The results also bring out an important aspect in solving uniformly and non-uniformly scaled many-objective optimization problems. Objectives in DTLZ problems are uniformly scaled and the performance of MOEA/D has been found to perform the best; while objectives in WFG problems are non-uniformly scaled and the performance of NSGA-II has been found to be the best. Since MOEA/D algorithm does not have any explicit normalization operator, it does not perform well on scaled problems. In the future, we shall launch a more detailed study investigating the performance of MOEA/D with an NSGA-III like normalization operator. The normalization of objectives should also improve the performance of M2M variants. Nevertheless, this extensive study brings out the need for a balance in introducing parallelism in an algorithm and restricting the search through a decomposition approach, which is a current trend in EMO research. Further such studies will provide a more clear understanding of the trade-off between parallelism and generality, which is crucially important in developing computationally efficient evolutionary multi- and many-objective optimization algorithms.

Acknowledgments

The visit of the first author to Michigan State University was supported by China Scholarship Council. This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454.

References

1. K. Deb, Multi-objective optimization using evolutionary algorithms. Chichester, UK: Wiley,2001.
2. J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. on Genetic Algorithms*, Pittsburgh, PA, USA, 1985, pp. 93-100.
3. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, 2002.
4. E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm", in *Proc. EUROGEN*, 2001, pp. 95-100.
5. D. W. Corne, N. R. Jerram, J. D. Knowles and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization", in *Proc. GECCO*, 2001, pp. 283-290.
6. K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal? " in *Proc. IEEE CEC*, Seoul, Korea, 2001, pp. 957-962.

7. M. Li , S. Yang , and X. Liu , "A comparative study on evolutionary algorithms for many-objective optimization," in *Proc. ENO*, 2013, pp. 261-275.
8. T. Wagner, N. Beume and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. EMO*, 2006, pp. 742-756.
9. Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, pp. 712-731, Dec. 2007.
10. H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284-302, Apr. 2009.
11. H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. IEEE Trans. Syst., Man, Cybern. C*, vol. 28, no. 3, pp. 392-403, Aug. 1998.
12. K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577-601, Aug. 2015.
13. K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694-716, Oct. 2015.
14. Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16-37, Feb. 2016.
15. R. Cheng, Y. Jing, M. Olhofer and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773-791, Oct. 2016.
16. K. Miettinen, "Nonlinear Multiobjective Optimization", Boston: Kluwer. 1999.
17. H-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, Jun. 2013.
18. I. Das and J. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631-657, Jul. 1998.
19. M. W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb, and M. Ó. Cinnéideet, "High dimensional search-based software engineering: finding tradeoffs among 15 objectives for automating software refactoring using NSGA-IIIin *Proc. ACM Annu. Conf. Genet. and Evol. Computat.*, 2014, pp. 1263-1270.
20. K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115-48, Nov. 1995.
21. S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477-506, 2006.
22. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multi-Objective Optimization.* Kanpur, India: Kanpur Genetic Algorithms Lab. (KanGAL), Indian Inst. Technol., 2001. KanGAL Report 2001001.
23. E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257-271, Nov. 1999.
24. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117-132, Apr. 2003.