Explicit and Implicit Parallelisms in Decomposition Based Evolutionary Many-Objective Optimization Algorithms

Lei Chen[*], Kalyanmoy Deb [†], and Hai-Lin Liu

COIN Report Number 2018004R [‡]

May 25, 2018

Abstract

Over the past two decades, evolutionary multi-objective optimization (EMO) algorithms have demonstrated their ability to find and maintain multiple trade-off solutions in two and three-objective problems. This is because their operators are able to establish an implicit parallel search within an evolving population around multiple optimal regions of the search space simultaneously. For many-objective optimization problems involving a large-dimensional objective space, the extent of parallelism present in early EMO methods was found to be too generic. Decomposition-based EMO algorithms which divide the overall computing task into a number of sub-tasks of focusing within a region of the search space have found to be successful in solving many-objective problems. In this paper, we term this external control of an algorithm's parallelism as 'explicit parallelism' set by the algorithm developer. Although such a decomposition concept compromises on the *implicit* parallelism aspect of an EMO algorithm, an externally defined coordination among different subtasks is able to bring back the requisite parallelism needed to solve them. In this paper, we consider three decomposition-based EMO algorithms – MOEA/D, M2M, and NSGA-III – to investigate the effect of user-controlled explicit parallelism mechanism on their search operators. For this purpose, we first consider a number of M2M variants with differing levels of explicit parallelism and identify the most-balanced algorithm between explicit and implicit parallelisms by applying them on a number of standard many-objective unscaled and scaled test problems (DTLZ and WFG problems). Results from our extensive study indicate that by relaxing the decomposition effect, thereby re-establishing an appropriate parallel search within M2M operators, the performance of the resulting M2M variants can be improved. Motivated by the M2M-variant study, we repeat the procedure on NSGA-III and MOEA/D operators and observe interesting but unique balancing acts between explicit and implicit parallelisms that each of the algorithms requires. We also investigate the effect of normalization of objectives in improving the performance of MOEA/D and M2M methods and report much improved performances than the original methods. The overall approach helps to develop more efficient algorithms than the original methods. The principles of this study can be used to improve the performance other EMO methods.

1 Introduction

After the success of population-based evolutionary algorithms (EAs) in solving two and three-objective optimization problems for finding multiple Pareto-optimal solutions simultaneously through an evolving population, researchers and applicationists became interested in solving four and more objective optimization problems. While the main crux of this paper is on solving evolutionary many-objective optimization problems involving up to 15-objective problems, first, we begin by discussing probable reasons for the success of EAs in solving two and three-objective optimization problems.

EAs use a population of solutions in each generation and its operators select good solutions, recombine them in the hope of mixing their good features to produce new and hopefully better offspring solutions. Occasional mutations of offspring solutions ensure that the evolving population is not stuck prematurely to sub-optimal regions and also help achieve locally better solutions. Starting with Holland's $O(N^3)$ (where N is the population size) schema processing argument in 1975 [21] to Goldberg's implicit parallelism argu-

[*]L. Chen and H.-L. Liu are with Guangdong University of Technology, Guangzhou, China, e-mail: chenaction@126.com, hlliu@gdut.edu.cn.

[†]K. Deb is with Department of Electrical and Computer Engineering and BEACON Center for the Study of Evolution in Action (NSF DBI-0939454), Michigan State University, 428 S. Shaw Lane, 2120 EB, East Lansing, MI 48824, USA, e-mail: kdeb@egr.msu.edu

[‡]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.

ment [18] in 1989 to more theoretical studies in the EA field have all provided reasons for the working of EAs as follows. When a population of N solutions are evolved within an EA framework, much more than $O(N)$ building blocks (the essential ingredients needed to form a near-optimal solution) get processed in every generation. More broadly, this means that more than N sub-regions of the search space gets evaluated and processed in a generation for the overall EA to sort out the promising regions to concentrate, despite the use of only N population members. A single solution based search algorithm misses this implicit parallelism aspect of population-based search algorithms, providing EAs their global perspectives and their ability to solve complex problems than their point-based counterparts.

When EAs are applied to two and three-objective optimization problems, there is no single optimal solution which is the target; rather the goal is to find multiple trade-off Pareto-optimal solutions in a single simulation run. Early EA researchers interested in solving multi-objective optimization problems [16, 18, 22, 41, 42] have realized that in addition to converging towards a single Pareto-optimal solution, a diversity-preserving or *niching* operator was needed to establish stable and sustained sub-populations along various regions of the Pareto-optimal front within an evolving population. For two or three-objective optimization problems, once a suitable niching is established, EA's other operators were able to impose the needed implicit parallelism to constitute an efficient search within each niche and also among the niches. The first generation algorithms were made better by introducing more efficient and, in some cases, parameter-less, second generation algorithms, such as NSGA-II [8], SPEA2 [50], PESA-II [5]. In most problems, a low-dimensional Pareto-optimal objective space comes from a low-dimensional variable space interactions, piece-wise or parametrically, despite having a large-dimensional variable space description of the original problem [11].

However, when many-objective problems involving four or more objectives are to be solved, variable interactions requiring to represent the Pareto-optimal solutions increase and the simple implicit niching based methods designed to solve two and three-objective optimization problems cease to find a diverse set of Pareto-optimal solutions on the entire front. The need to preserve solutions from a widely spread-out objective space for EA's implicit parallelism mechanism to find new and continuously evolving non-dominated solutions gets less emphasized by the so-called dominance-resistance phenomenon. This fact has alluded EA researchers for a long time and they took a number of different paths to reliably solve many-objective optimization problems. A lot of efforts have been made to modify the Pareto dominance relationship, such as, with α-dominance [24], ϵ-dominance [10, 20], subspace dominance [28], fuzzy dominance [33], L-optimality [15], grid dominance [46], and preference order ranking [13]. Another attempt to remedy the dominance resistance phenomenon in certain problems was to use a dimension reduction technique [4, 27, 40], in which redundant objectives were identified and eliminated, as and when found, thereby reducing the effective dimensionality of the Pareto-optimal set. Other ways to beat the high-dimensionality aspect was to use a scalarized metric to provide information about poorly progressed part of the Pareto-optimal front and then introduce an enhanced search there. Indicator based EMO algorithms, such as hypervolume indicator [1, 2, 30, 36, 39, 43, 45], R2 indicator [19] and $I_{\varepsilon+}$ indicator [44] used a metric-based selection for solving many-objective optimization problems.

In 2007, Zhang and Li [49] came out with a decomposition-based algorithm (MOEA/D) that made the implicit niching methods somewhat explicit, by dividing the overall task of finding the entire multi-dimensional Pareto-optimal front into a number of loosely interacting sub-tasks. This reduced the effect of entire search space wide implicit parallelism aspect of early evolutionary multi-objective optimization algorithms. Instead, MOEA/D relied more on implicit parallelism to take place within each sub-problem and leaving the automatic distribution of Pareto-optimal solutions to parallel but more independent evolution of individual sub-problems. The trick lied in choosing an appropriate sub-problem size and a suitable metric for evaluating solutions within a sub-problem. In decomposition-based evolutionary many-objective optimization (EMO) algorithms, a set of decomposition vectors is predefined, either for objective aggregation [25, 34, 49] or for diversity and convergence enhancement [3, 9, 29, 35, 47, 48]. 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 was introduced. One extreme with a zero parallelism is the classical "generating methods" [38], which attempts to find a single Pareto-optimal solution in a single optimization task.

The above discussion raises an important distinction between the well-known "implicit" parallelism associated with a population-based EA and an "explicit" parallelism mechanism which is introduced by the algorithm developer. The decomposition mechanism, the restriction on the choice of parents, and procedures for accepting a child solution to a subproblem are explicitly designed by the algorithm developer, but have a huge impact in interacting with the inherent implicit parallelism introduced by the EMO's standard selection, recombination and mutation operators. A balance between the explicit parallelism introduced by the developer and implicit parallelism introduced by the EMO's operators is crucial for the overall algorithm to work well. In this paper, we demonstrate the need for such a balance in solving large-dimensional problems and present a balancing procedure on three popularly used EMO algorithms. The current study uses three main decomposition-based EMO methods – MOEA/D, M2M and NSGA-III – for this purpose. M2M [37] is a new variant of MOEA/D for population decomposition, and it decomposes a many-objective optimization problem into a set of many-objective optimization sub-problems. M2M makes the sub-problems more independent from each other, thereby making the respective algorithm less implicitly parallel. NSGA-III [9] is the third generation non-dominated sorting genetic algorithm, but it uses a decomposition-based niching method to achieve the required convergence and diversity.

The remainder of paper is organized as follows. Section 2 defines a many-objective optimization problem and then briefly introduces the basic working principles of MOEA/D, M2M, and NSGA-III. Section 3 explains the concepts of the two modes of parallelisms associated with solving unscaled and scaled many-objective optimization problems. In Section 4, experiments are conducted with M2M algorithm and its variants. The balance between explicit and implicit parallelism is demonstrated through simulation results on both unscaled and scaled problems. MOEA/D and M2M algorithms did not have an explicit normalization procedure to handle scaled problems – common in practical problems. We apply two different normalization procedures within MOEA/D and M2M to comprehensively evaluate these two algorithms. Section 5 conducts the experimental studies and analysis on the performance of two normalization procedures in MOEA/D-M2M variants and MOEA/D. Further performance comparison on a series of NSGA-III variants and MOEA/D variants are conducted in Section 6 to find if these two methods are properly balanced for explicit and implicit parallelisms and interesting observations are made in Section 7. Finally, Section 8 concludes this paper.

2 Preliminaries

In this section, we first define an optimal solution in a multi-objective optimization problem and then briefly introduce three main algorithms used in this paper – MOEA/D-M2M [37], NSGA-III [9], and MOEA/D [49].

2.1 Problem Definition

A multi-objective optimization problem (MOP) can be generally defined as:

$$\begin{aligned} \text{minimize} \quad & F(\mathbf{x}) = \{f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\}, \\ \text{subject to} \quad & \mathbf{x} \in \mathbf{\Omega} \subset \mathbb{R}^n, \end{aligned} \tag{1}$$

meaning that each objective function f_i must be minimized. The vector $\mathbf{x}$ is a n-dimensional decision variable vector and the feasible search region is defined as $\mathbf{\Omega} \subset \mathbb{R}^n$. An objective vector $F \in \mathbb{R}^m$ is a m-dimensional vector, which is mapped from the decision space. When $m > 3$, it is called a many-objective optimization problem (MaOP) [32]. A feasible solution (or point) $\mathbf{u} = (u_1, ..., u_m)$ is said to *dominate* another feasible solution (or point) $\mathbf{v} = (v_1, ..., v_m)$, if $u_i \leq v_i$ for all $i = 1, \ldots, m$, and $\mathbf{u} \neq \mathbf{v}$. A point $\mathbf{x}^*$ is called Pareto-optimal, if there is no $\mathbf{x} \in \mathbf{\Omega}$ such that $F(\mathbf{x})$ dominates $F(\mathbf{x}^*)$. The set of all the Pareto-optimal points in the decision variable space is called the Pareto Set (PS). A Pareto Front (PF) in the objective space is defined as PF $= \{F(\mathbf{x}) \in \mathbb{R}^m | \mathbf{x} \in \text{PS}\})$ [38]. The unattainable vector $\mathbf{z}^* = (z_1^*, \ldots, z_m^*)$ is called the ideal point, formed with z_i^* describing the minimal value of $f_i(\mathbf{x})$ over the feasible decision space $\mathbf{\Omega}$, and another point $\mathbf{z}^{nad} = (z_1^{nad}, \ldots, z_m^{nad})$ is called the nadir point, formed with z_i^{nad} representing the maximal value of $f_i(\mathbf{x})$ over the PS.

2.2 MOEA/D Framework

MOEA/D decomposes an MOP or an MaOP into a number of single-objective optimization subproblems through aggregation functions with the aid of a set of predefined weights. Three aggregation methods, namely, weighted-sum, Tchebycheff and Penalty Boundary Intersection (PBI), are most commonly used with the MOEA/D framework. In our study here, we use the PBI method for aggregation, and thus we only briefly describe MOEA/D with PBI aggregation method.

For a set of given decomposition vector ($\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$), MOEA/D-PBI decomposes MOP or MaOP to N single-objective subproblems, as follows:

$$\begin{aligned} \text{minimize} \quad & g_j^{pbi}(\mathbf{x}|\mathbf{v}) = d_{j,1}(F(\mathbf{x})) + \theta d_{j,2}(F(\mathbf{x})), \\ \text{subject to} \quad & \mathbf{x} \in \mathbf{D}, \end{aligned} \tag{2}$$

where $j = 1, \ldots, N$, $\theta \geq 0$ is a pre-defined penalty parameter, and $\mathbf{v}_j = (v_{j,1}, \ldots, v_{j,m})$ is the decomposition vector for j-th subproblem. The first distance metric $d_{j,1}(F(\mathbf{x})) = (F(\mathbf{x}) - \mathbf{z}^*)^T \mathbf{v}_j / ||\mathbf{v}_j||$ is the projected distance of $F(\mathbf{x})$ to the decomposition vector $\mathbf{v}_j$, and $d_{j,2}(F(\mathbf{x})) = ||(F(\mathbf{x}) - \mathbf{z}^*) -$

$d_{j,1}(F(\mathbf{x}))\mathbf{v}_j||\mathbf{v}_j||)||$ is the perpendicular distance of $(F(\mathbf{x})-\mathbf{z}^*)$ vector to the decomposition vector $\mathbf{v}_j$. In MOEA/D, recombination is conducted between one parent solution from a decomposition vector $\mathbf{v}_j$ and another parent within a pre-defined neighborhood of $\mathbf{v}_j$, defined by a niching parameter T, with a probability of $p_r = 0.9$, or with another member from the entire population with a probability of 0.1. Once an offspring solution is created, a steady-state selection scheme is used to update the current population at most n_r times. With a probability of $p_u = 0.9$, the created offspring is checked with the first parent's neighborhood for an update and with a probability of 0.1 the offspring is checked with all subproblems. The updates are made based on their respective PBI metric and in the event of a success, the offspring replaces an existing population member from the respective subproblem. It is important to note that MOEA/D-PBI method involves a few user-defined parameters: θ, n_r, p_r, and p_u, in addition to other standard EA parameters. The original study suggested $\theta = 5$, $n_r = 2$, $p_r = 0.9$, and $p_u = 0.9$. No specific constraint handling procedure was suggested.

2.3 MOEA/D-M2M Framework

MOEA/D-M2M is a population decomposition framework, where the M2M decomposition can decompose a multi-objective or many-objective optimization problem into a number of multi- and many-objective optimization subproblems. In this section, we will briefly introduce the basic decomposition principle.

For simplicity, it is assumed that all the translated objective functions ($\tilde{f}_j(\mathbf{x}) = f_j(\mathbf{x}) - z_j^*$) are non-negative. MOEA/D-M2M [37] population decomposition utilizes K unit vectors $\mathbf{v}_1, \ldots, \mathbf{v}_K$ in $\mathbf{R}_+^m$ for decomposition, and $\mathbf{R}_+^m$ is divided into K subregions in the objective space $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_K$, where $\mathbf{\Omega}_k$ ($k = 1, \ldots, K$) is defined as follows:

$$\mathbf{\Omega}_k = \{\mathbf{u} \in \mathbf{R}_+^m | \angle(\mathbf{u}, \mathbf{v}_k) \leq \angle(\mathbf{u}, \mathbf{v}_j) \text{ for any } j = 1, \ldots, K\}, \quad (3)$$

where $\angle(\mathbf{u}, \mathbf{v}_j)$ is the acute angle between individual objective vector $\mathbf{u}$ and the decomposition vector $\mathbf{v}_j$. In other words, $\mathbf{u}$ belongs to $\mathbf{\Omega}_k$ if and only if $\mathbf{v}_k$ has the smallest angle to $\mathbf{u}$ among all the K decomposition vectors. In this way, an MOP/MaOP can be transformed into K constrained multi- and many-objective optimization subproblems. Subproblem k is:

$$\begin{aligned} \text{minimize} \quad & \tilde{F}(\mathbf{x}) = (\tilde{f}_1(\mathbf{x}), \ldots, \tilde{f}_m(\mathbf{x})), \\ \text{subject to} \quad & \tilde{F}(\mathbf{x}) \in \mathbf{\Omega}_k. \end{aligned} \quad (4)$$

During the evolutionary process, recombination and the generational selection are conducted for each subproblem independently. If an offspring is created outside the subregion describing the subproblem, then it is migrated to the respective subproblem. After all N/K offsprings are created for a subproblem, they are used to update the parent subpopulation using the PBI metric. The additional parameters involved in this algorithm are K and θ. No specific constraint handling procedure was suggested.

2.4 NSGA-III Framework

At every generation, NSGA-III uses a normalization procedure based on the objective values of the current population to transform all normalized objective values to take non-negative values. A detailed description is provided in Section 5.2. Thereafter, NSGA-III sorts all population members (parents and offspring) according to increasing level (or front) of domination. Each front members are then accepted in bulk, starting from the non-dominated front, until the population cannot accept any more. A reduced set of final front members are selected based on a niching procedure to maintain the population size at every generation. The niching procedure is described here. Like in MOEA/D, NSGA-III uses a set of reference directions, created initially by joining origin to a uniformly distributed set of points on a m-dimensional unit simplex. Normalized objective vectors are then associated with a single reference direction based on the d_2 metric described in Section 2.2. Then, for each reference direction, all associated objective vectors are identified and a systematic niching procedure is used to establish an uniform selection of associated members from each reference direction. It is important to note that both recombination and niching operations are performed on the entire population, making this method less restrictive and probably most parallel among the three methods of this study. Also, NSGA-III does not require any additional parameters. Constraint handling is included in its domination principle.

3 Implicit and Explicit Parallelism

In an EA, the implicit parallelism comes from the effect of a selecto-recombination search operator applied to an evolving population. Depending on the "designed" extent of parallelism allowed to the operators, a highly fit population member can be recombined with another such population member from a completely different part of the search space and produce a child having mixed characteristics of both parents. In problems where such a child is evaluated as fitter than at least one of its parents, such implicitly parallel evolutionary algorithms will flourish and perform well. The designed aspect of introducing parallelism to an algorithm externally is termed here as explicit parallelism here. In other studies, this act is termed as 'parameter control' [14, 31], but we view here this act as an explicit way of introducing a parallelism in the search process. The developer of an algorithm can restrict the extent of implicit parallelism to be established in indirectly in a population by directly controlling externally who should mate with whom and who can be compared against each other in the its selection process.

Here, we explain the two modes of parallelism related to decomposition-based EMO algorithms. 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 according to the ensuing algorithm.

- **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 used in creating new solutions and in competing with each other. 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 an overall effective search. 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 an overall 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 restrictions or relaxations on recombination, selection and migration procedures of EMO algorithms. Figure 1 presents a sketch explaining the levels of explicit parallelism introduced in a few existing multi- and many-objective optimization algorithms. Classical generating methods [38], in

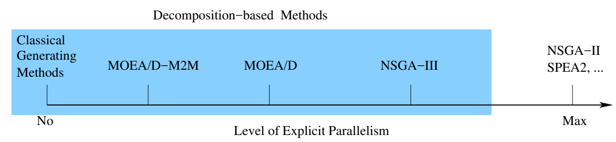

Figure 1: Levels of explicit parallelism in different decomposition-based multi- and many-objective optimization algorithms.

which each Pareto-optimal solution is attempted to be found separately by scalarizing the problem into a single objective, are least parallel in terms of finding a set of well-distributed Pareto-optimal solutions. In decomposition-based EMO algorithms, the population is divided into different subpopulations and they can either evolve independently or collaboratively within an algorithm. The interactions can be controlled by an explicit parallelism mechanism. For example, in MOEA/D-M2M, the population is decomposed into a number of subpopulations describing each subproblem. If the explicit parallelism is made too restrictive, each subpopulation is expected to evolve independently with implicit parallelism effect restricted only to each subproblem. We can easily relax this explicit parallelism mechanism of MOEA/D-M2M by allowing subpopulations to interact more with each other. MOEA/D-M2M framework allows us to control the effect of explicit parallelism by relaxing the original restrictions of its operators. In this sense, the MOEA/D-M2M method introduces a higher level of explicit parallelism than the classical generating methods. In MOEA/D, the selection procedure is confined within a neighborhood of a reference vector, but recombination is not restricted within the same neighborhood, thereby making the process more explicitly parallel. NSGA-III, on the other hand, is not restrictive in its recombination operator, but restricts its niching operation within associated members of every reference line. Although, it introduces a higher level explicit parallelism than MOEA/D or M2M due to its global dominance check, evolutionary multi-objective optimization algorithms, such as, NSGA-II, SPEA2 and others, possess the largest level of explicit parallelism without any restriction on selection or recombination operations. As mentioned before, such a flexible explicit parallelism could not be extended to solve many-objective optimization problems, at least as yet. Table 1 shows the restriction of different operators within MOEA/D-M2M, NSGA-III, and MOEA/D.

It is important to note that a higher level of explicit parallelism does not mean that the algorithm is more efficient. Every problem requires a good balance between explicit and implicit parallelism for an algo-

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

Operation	NSGA-III	MOEA/D-M2M	MOEA/D
Selection	Yes	Subregion	Neighbor
Recombination	Yes	Subregion	Neighbor
Migration	Yes	Yes	No

rithm to work its best. For some problems, NSGA-III's explicit parallelism may be too flexible for it to establish the necessary focused search needed for them to be solved efficiently and MOEA/D-M2M's explicit parallelism may be just adequate for the problem. On the other hand, some complex problems may make MOEA/D-M2M's explicit parallelism to be too restrictive, and need NSGA-III's much flexible level of explicit parallelism. Explicit parallelism is controlled by the user, and depending on its set level, the algorithm's operators facilitate a certain level of implicit parallelism within an evolving population. If the search power generated by such a derived implicit parallelism mechanism is adequate for a problem class to be solved efficiently, the algorithm with its set explicit parallelism will be successful to solve the specific problem class. It is then important for an algorithm designer to know how easy or flexible it is to set an algorithm's level of explicit parallelism, so that some procedures can be developed to set the right level of explicit parallelism either initially or temporarily. While we do not address the issue of sophisticated ways of setting the right level of explicit parallelism for a problem class, we demonstrate here how to achieve the right level of explicit parallelism initially and manually into three popular EMO algorithms – MOEA/D, MOEA/D-M2M, and MSGA-III – in solving two commonly-used series of problem classes – DTLZ and WFG.

4 Variants of MOEA/D-M2M and Results

First, we discuss MOEA/D-M2M variants and the results obtained with them. 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 subpopulations. Here, we vary the probabilities of selection and recombination within each subpopulation of the original MOEA/D-M2M to have different levels of explicit parallelism, and then compare the performance of MOEA/D-M2M variants with the original NSGA-III and MOEA/D methods. We create a total of 14 MOEA/D-M2M variants including the original one (Table 2). For simplicity, we create (v1,...,v14) with corresponding probabilities (p_1, p_2) to represent each of the variants of MOEA/D-M2M, where p_1 and p_2 are the probabilities of selection within the subpopulation and the probability of recombination within the subpopulation, respectively. For example, MOEA/D-M2M-v1 represents the original MOEA/D-M2M with $p_1 = 1$ and $p_2 = 1$, respectively. In this M2M-variant, both selection and recombination operators are restricted to take place within each subpopulation, thereby representing the original MOEA/D-M2M. In the parlance of this paper, this corresponds to no explicit parallelism among subpopulations. On the other hand, MOEA/D-M2M-v4 uses $p_1 = 1$ and $p_2 = 0$ and allows recombination to take place among the entire population, while selection is still restricted among members of each subpopulation. MOEA/D-M2M-v5 (with $p_1 = 0.67$ and $p_2 = 0.67$) restricts selection and recombination within each subpopulation 67% of the time and applies the operators among the entire population the rest 33% of the time. MOEA/D-M2M-v11 (with $p_1 = 0$ and $p_2 = 0$) is the most explicitly parallel M2M variant, in which any two population members can be compared in the selection operator and any two population members are allowed to participate in the recombination operator.

In the remainder of this section, we present results from an extensive experimental study to demonstrate the effect of different levels of explicit parallelism on the performance of M2M method. Mutation operator is applied as usual. In our experimental study, two sets of widely used benchmark problems from DTLZ [12] and WFG [23] families with three to 15 objectives are tested. These two types of benchmark problems are popular among the EMO community, because they cover a number of problem characteristics that bring challenges for evolutionary algorithms to obtain a set of well representative solutions along the Pareto Front. Each of the M2M variants, the original MOEA/D and the original NSGA-III methods are run 15 times independently on each test problem. The IGD-metric [51] is used to measure the quality of obtained solutions for each test problem. Statistical evaluation of the multiple runs is accounted by means of Wilcoxon rank-sum test in each case.

4.1 Parameter Settings

In the experimental studies, the population size N, number of sub-regions (K), and number of divisions H for generating initial decomposition vectors for all algorithms are kept the same, and are shown in Table 3. Problems DTLZ1 to DTLZ4 and WFG5 to WFG8 with the number of objectives $m = 3$, 5, 8, 10 and 15 are tested here. The number of decision variables is set as $n = m + 4$ for DTLZ1, $n = m + 9$ for DTLZ2 to DTLZ4, and $n = m + 19$ for WFG5 to WFG8. The number of generations for each problem

Table 2: Variants of MOEA/D-M2M and their respective parameter (p_1, p_2) values.

Variant	v1	v2	v3	v4	v5	v6	v7
(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	v8	v9	v10	v11	v12	v13	v14
(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)

are also presented in Table 2. The SBX [7] operator with $p_c = 1$ and $\eta_c = 30$, and polynomial mutation [6] with $p_m = 1/n$ and $\eta_m = 20$ are used for each algorithm. We use other MOEA/D and M2M parameters

Table 3: Population size, number of divisions in each objective axis for generating decomposition vectors, and number of subregions used in this study. Note that ♯Div. values for 8 10, and 15 objectives are shown for two layers of reference directions [9].

♯Obj.	♯Div.	K	PopSize
3	12	10	91
5	6	14	210
8	3,2	12	156
10	3,2	17	275
15	2,1	12	135

Table 4: Number of generations for DTLZ and WFG 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
15	1500	1000	2000	3000	3000

as they were suggested in the respective original studies.

4.2 IGD Metric

The IGD-metric [51] is used to measure the performance of all 16 algorithms, and another 20 variants of NSGA-III and MOEA/D in the next section. To calculate the IGD-metric, first, we need to obtain $\mathbf{P}^*$ – a set of reference points which are uniformly distributed along the PF in the objective space. The IGD-metric between the $\mathbf{P}^*$ and $\mathbf{P}$ can then be defined as follows:

$$IGD(\mathbf{P}^*, \mathbf{P}) = \frac{\sum_{\mathbf{v} \in \mathbf{P}^*} d(\mathbf{v}, \mathbf{P})}{|\mathbf{P}^*|},$$

where $d(\mathbf{v}, \mathbf{P})$ is the minimum Euclidean distance from the point $\mathbf{v}$ to $\mathbf{P}$. Intuitively, the smaller the value of IGD-metric, the better is the algorithm.

4.3 Simulation Results on DTLZ Problems

Median IGD values on DTLZ problems are presented in Table 5 for the top three M2M variants (v2, v3 and v4) and three other original algorithms – NSGA-III, MOEA/D and M2M itself. The MOEA/D and M2M variants are applied without any objective normalization procedure, as they were originally proposed. The best performance is highlighted in bold with a gray background. We also perform the Wilcoxon rank-signed test for all 15 runs starting with different initial populations and the performance without a significant difference with 95% confidence with the best-performing algorithm is italicized with a gray background as well. It is interesting to note that while MOEA/D performs well on all low-dimensional DTLZ problems, certain M2M variants perform exceedingly well on low and high-dimensional DTLZ problems. Importantly, the three M2M variants perform much better than the original M2M algorithm. M2M-v2 with $p_1 = 1$ and $p_2 = 0.67$ allows recombination to take place among all population members irrespective of the subpopulations they belong. Such an explicit parallelism allows M2M-v2 to perform much better than the more restrictive original M2M algorithm.

Wilcoxon signed-rank test results are presented in Table 6. Each algorithm is compared with the best-performing algorithm (shown with a bold zero in the table) in terms of their median performance. If an algorithm's performance is not significantly different with p-value less than 0.05, a value of zero is assigned, otherwise a value of one is assigned. Since the performance of the algorithms vary significantly with an increase in number of objectives, we consider $m = 3$ and 5-objective problems as relatively low-dimensional problems and $m = 8$, 10, and 15-objective problems as high-dimensional. A sum of the above values on eight low-dimensional problems instances reveal that MOEA/D performs the best, followed by M2M-v2, v3 and v4. Importantly, the original M2M (v1) and NSGA-III fail to perform well on these problems. On 12 high-dimensional DTLZ problems, MOEA/D performs well, but the best perfor-

Table 5: Median IGD values of NSGA-III, MOEA/D, original M2M, and the top three MOEA/D-M2M variants on DTLZ problems.

Method	DTLZ1-3	DTLZ1-5	DTLZ2-3	DTLZ2-5	DTLZ3-3	DTLZ3-5	DTLZ4-3	DTLZ4-5
NSGA-III	0.002447	0.003372	0.001878	0.006894	0.004459	0.009924	0.000836	0.001176
MOEA/D	*0.001525*	*0.000486*	**0.000540**	**0.000919**	*0.004297*	*0.001819*	0.000133	**0.000082**
M2M	0.007047	0.001015	0.001078	0.001863	0.030759	0.017584	0.000131	0.000348
M2M-v2	*0.001133*	**0.000392**	0.000640	0.001549	**0.003053**	*0.002317*	**0.000111**	0.000254
M2M-v3	**0.001089**	**0.000392**	0.000612	0.001755	*0.003778*	**0.001367**	*0.000119*	0.000251
M2M-v12	*0.001620*	0.000678	0.000682	0.001529	*0.004059*	0.002477	*0.000118*	0.000206

Method	DTLZ1-8	DTLZ1-10	DTLZ1-15	DTLZ2-8	DTLZ2-10	DTLZ2-15
NSGA-III	0.006439	0.004012	0.005126	0.002169	0.017623	0.019434
MOEA/D	0.003949	0.005122	0.015709	**0.002035**	**0.002879**	*0.005962*
M2M	0.011439	0.011507	0.048595	0.007692	0.006900	0.010018
M2M-v2	0.003673	0.003591	0.013176	0.005761	0.005235	*0.006207*
M2M-v3	*0.002521*	0.002672	0.011815	0.005691	0.005164	*0.006151*
M2M-v12	**0.002349**	**0.001988**	**0.003665**	0.005186	0.004720	**0.005590**

Method	DTLZ3-8	DTLZ3-10	DTLZ3-15	DTLZ4-8	DTLZ4-10	DTLZ4-15
NSGA-III	0.034200	0.019540	0.029303	0.004183	*0.004802*	*0.006760*
MOEA/D	**0.008675**	**0.003935**	0.018911	**0.001148**	**0.001175**	0.108072
M2M	0.031047	0.013649	0.037458	0.004516	*0.004388*	0.013238
M2M-v2	*0.008868*	*0.005157*	**0.005950**	0.002532	*0.002246*	*0.003711*
M2M-v3	*0.010207*	0.005458	0.008016	0.002366	*0.002144*	**0.002765**
M2M-v12	0.013780	0.007031	*0.006414*	0.002190	*0.001932*	*0.003281*

mance comes from M2M-v12 variant with $p_1 = 0$ and $p_2 = 0.33$. M2M-v2 performs as well as MOEA/D and the original M2M (v1) performs the worst.

Overall, we may conclude from the above results that while MOEA/D performs the best in well-scaled DTLZ problems from three to 15 objectives, certain M2M variants compare well with MOEA/D, but importantly these M2M variants are much better than the original M2M algorithm. For example, MOEA/D-M2M-v2 employs a restricted selection within each subpopulation, whereas the recombination is explicitly 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 offspring solutions and 33% probability event makes a good balance between exploiting parallelism and negotiating generality needed to solve DTLZ problems. It is also interesting to note that the M2M 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.

4.4 Simulation Results on WFG Problems

WFG problems produce a non-uniform scaling of objective values, thereby requiring an appropriate normalization of objectives for an EMO algorithm to work well. Table 7 shows the median IGD values of top three M2M variants and other original methods – NSGA-III, MOEA/D and M2M – on WFG problems. As before, MOEA/D and M2M algorithms are applied without any objective normalization, as they were originally proposed. M2M variants are also applied without any normalization here. We observe a completely different outcome compared to those observed in DTLZ problems. Now, NSGA-III works the best and it works so well compared to other algorithms that all other methods are significantly worse (with $p < 0.05$) in the Wilcoxon rank-sum test.

To show the relative performance of non-NSGA-III methods, we exclude NSGA-III from the comparison and perform a Wilcoxon rank-sum test with the rest of the methods. Table 8 shows the rank values (zero or one with bold zero indicating the best-performing non-NSGA-III method). It is interesting to note that M2M-v2 to v3 perform better than MOEA/D in these problems for both low and high-dimensional versions of WFG problems. Specifically for high-dimensional WFG problems, these three M2M versions perform exceedingly well.

Overall, for solving scaled WFG problems, NSGA-III outperforms MOEA/D, M2M and M2M-variants on both low and high-dimensional problems. However, M2M-v2, v3 and v4 are much better than original M2M and other M2M-variants and also better than the MOEA/D algorithm. While NSGA-III's super-performance is due to its internal normalization procedure which handles WFG problems better, certain M2M-variants with a controlled explicit parallelism built in perform better than MOEA/D and original M2M despite the absence of any normalization proce-

Table 6: Wilcoxon rank-sum test values for the variants of MOEA/D-M2M, NSGA-III, and MOEA/D on DTLZ test problems.

Instance	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	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**
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
DTLZ1-15	1	1	1	1	1	1	1	1	1	1	0	**0**	0	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**
DTLZ2-15	1	0	0	1	1	1	1	1	1	1	0	0	**0**	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**
DTLZ3-15	1	**0**	1	1	1	1	1	1	1	1	1	0	0	1	1	1
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**
DTLZ4-15	1	0	**0**	0	1	0	1	0	1	1	0	0	1	1	0	1
Sum	11	5	7	9	11	10	11	10	11	11	6	**4**	7	11	10	5

dure. This indicates that MOEA/D and M2M variants must be associated with a suitable normalization procedure to solve scaled problems, which are usually the case in real-world applications – a matter which we consider next.

5 Extended M2M and MOEA/D Algorithms with Normalization

Here, we modify M2M variants and MOEA/D algorithm with two normalization procedures – (i) a simple and naive normalization and (ii) NSGA-III's normalization.

5.1 Simple Normalization Procedure

First, we introduce the simple normalization procedure to MOEA/D-M2M and MOEA/D to negotiate the scalability issue. The objectives $f_i(\mathbf{x})$ ($i = 1, 2, ..., m$) of a solution $\mathbf{x}$ is normalized as follows:

$$\tilde{f}_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - z_i^{\min}}{z_i^{\max} - z_i^{\min}}, \qquad (5)$$

where $z_i^{\min}$ and $z_i^{\max}$ are the minimum and maximum value of f_i found in the evolving population since the start of a run. These values are updated in every generation. We refer to the modified MOEA/D with simple normalization as MOEA/D-S and M2M variants as M2M-S variants.

5.1.1 Results on DTLZ Problems

We apply all 14 M2M-S variants on low and high-dimensional DTLZ problems and observe that M2M-S-v11, v12 and v13 (Table 2) are the best-performing methods. These variants allow selection operator to choose parents from the entire evolving population and recombination operator with 100%, 67% and 33% probabilities of choosing a partner for mating from the entire population, respectively. These variants are more explicitly parallel than the original M2M. A mating restriction scheme was applied on NSGA-II to obtain a better performance in another study [17, 26].

Table 9 presents the median IGD-metric values of 15 independent runs. Not surprisingly, the simple normalization procedure does not help M2M variants much in solving the unscaled DTLZ problems, except in the 15-objective DTLZ4 problem. The performance of MOEA/D-S is also not better, in general, from the original MOEA/D, except in large-objective DTLZ4 problems. The table also presents results from an NSGA-III variant which we shall discuss in Section 6.1.

Table 7: Median IGD values of NSGA-III, MOEA/D, original M2M and the top three MOEA/D-M2M variants on WFG problems. NSGA-III outperforms all other algorithms on all chosen WFG problems.

Method	WFG5-3	WFG5-5	WFG6-3	WFG6-5	WFG7-3	WFG7-5	WFG8-3	WFG8-5
NSGA-III	**0.030312**	**0.032270**	**0.032546**	**0.034340**	**0.007695**	**0.009096**	**0.055666**	**0.088318**
MOEA/D	0.068102	0.206356	0.160424	0.368342	0.076203	0.279480	0.079065	0.242569
M2M	0.068986	0.178531	0.077098	0.180466	0.081060	0.179978	0.086545	0.194794
M2M-v2	0.068391	0.178995	0.073241	0.180259	0.070699	0.178796	0.081202	0.194540
M2M-v3	0.068190	0.178774	0.072668	0.180095	0.068557	0.178712	0.080951	0.194575
M2M-v4	0.068196	0.178923	0.070584	0.180031	0.068001	0.178740	0.080918	0.194206

Method	WFG5-8	WFG5-10	WFG5-15	WFG6-8	WFG6-10	WFG6-15
NSGA-III	**0.031780**	**0.033264**	**0.032821**	**0.030175**	**0.033839**	**0.040254**
MOEA/D	0.557644	0.681880	0.910326	0.842535	0.940924	1.096261
M2M	0.391055	0.546182	0.686288	0.462567	0.652632	0.836909
M2M-v2	0.385829	0.541831	0.679672	0.467113	0.661605	0.820415
M2M-v3	0.382307	0.534454	0.678947	0.469270	0.672973	0.831685
M2M-v4	0.381844	0.538845	0.671721	0.457262	0.649093	0.884749
NSGA-III	**0.008356**	**0.011790**	**0.011752**	**0.178498**	**0.234745**	**0.284391**
MOEA/D	0.809235	0.931088	1.092941	0.794623	0.955314	1.133327
M2M	0.449549	0.641922	0.770839	0.407763	0.666609	0.800532
M2M-v2	0.459846	0.643176	0.804302	0.437204	0.666723	0.827281
M2M-v3	0.450851	0.638046	0.808478	0.455618	0.649783	0.824768
M2M-v4	0.444614	0.643029	0.789138	0.435786	0.664692	0.819770

5.1.2 Results on WFG Problems

As expected, the performance of M2M-S variants on all scaled WFG problems is better compared to their non-normalized versions, as can be seen from Table 10. For WFG problems, M2M-S variants v2, v3 and v13 worked the best. Similar outcome can be observed for MOEA/D-S algorithm as well. While the original MOEA/D did not perform well on these problems at all (Table 7), on three and five-objective WFG8 problems, MOEA/D performs the best, even compared to NSGA-III. On eight and 10-objective WFG8, M2M-S-V13 performs the best.

5.2 NSGA-III Normalization Procedure on M2M and MOEA/D

NSGA-III uses a more sophisticated normalization procedure, which was found to work well on many scaled problems in the original study [9, 29]. Following normalization procedure was adopted in NSGA-III and is used to update M2M variants and MOEA/D algorithms. At each generation, first, the objective values are translated to origin by using the minimum objective-wise values of the current population. Then, m extreme points are picked from the current non-dominated front, where m is the number of objectives. These m points are used to form a linear hyperplane and the intercept of the hyperplane with each objective axis is found, as described in the original study [9]. To ensure that the linear hyperplane poses a non-dominated relationship of points lying on it, several tests are done. In the event of duplicate extreme points, the hyperplane formation fails and we then consider the nadir point (computed from the current non-dominated front) as intercept values. In the event of negative intercept value resulting from the hyperplane formation, we replace the intercept with the corresponding nadir point value. Finally, if the intercept is found to be smaller than a small value (1e-6 is used here), we replace the intercept with the maximum value across all population members for the particular objective axis. The extreme points, found above, are updated with the new extreme points at every generation. The extreme point and origin are then used to normalize each objective function. We apply the NSGA-III's normalization procedure to MOEA/D (call it MOEA/D-N) and M2M variants (call them M2M-N).

5.2.1 Results on DTLZ Problems

M2M-N variants v11, v12 and v13 performed the best with NSGA-III's normalization on DTLZ problems. It is clear from Table 11 that these three methods worked better on many problems compared to their non-normalized M2M variants. Compared to the results presented in Table 11 using the simple normalization, NSGA-III's normalization procedure resulted in better performance, in general. However, original MOEA/D (without any normalization) performed better than NSGA-III's normalization on DTLZ problems, due to a uniform scaling naturally present in DTLZ problems.

Table 8: Wilcoxon rank-sum test for the variants of MOEA/D-M2M and MOEA/D on WFG test problems. NSGA-III is excluded in computing the ranking in this table to get better idea of relative ranking of other algorithms.

Instance	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	MOEA/D
WFG5-3	1	0	0	0	1	0	**0**	1	0	1	0	0	0	1	0
WFG5-5	**0**	1	1	1	1	1	1	1	1	1	1	1	1	1	1
WFG6-3	1	1	1	**0**	1	0	0	1	1	0	0	0	1	1	0
WFG6-5	**0**	0	0	0	1	1	1	1	1	1	1	1	1	1	1
WFG7-3	1	1	0	0	1	0	0	1	0	**0**	0	0	1	1	1
WFG7-5	0	0	**0**	0	1	1	1	1	1	1	1	1	1	1	1
WFG8-3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**
WFG8-5	1	0	0	**0**	1	1	1	1	1	1	1	1	1	1	1
Sum	5	4	3	**2**	8	5	5	8	6	6	5	5	7	8	5
WFG5-8	0	0	0	**0**	1	1	1	1	1	1	1	1	1	1	1
WFG5-10	0	0	**0**	0	1	1	1	1	1	1	1	1	1	1	1
WFG5-15	1	0	0	**0**	1	1	1	1	1	1	1	1	1	1	1
WFG6-8	0	0	0	**0**	1	1	1	1	1	1	1	1	1	1	1
WFG6-10	0	0	0	**0**	0	1	0	1	1	0	1	1	1	1	1
WFG6-15	0	**0**	0	0	1	1	1	1	1	1	1	1	1	1	1
WFG7-8	0	0	0	**0**	1	1	1	1	1	1	1	1	1	1	1
WFG7-10	0	0	**0**	0	1	1	1	1	1	1	1	1	1	1	1
WFG7-15	**0**	0	0	0	0	1	0	1	1	1	1	1	1	1	1
WFG8-8	0	0	0	**0**	1	1	1	1	1	1	1	1	1	1	1
WFG8-10	0	0	**0**	0	1	1	1	1	1	1	1	1	1	1	1
WFG4-15	**0**	0	0	0	0	1	1	1	1	1	1	1	1	1	1
Sum	1	**0**	**0**	**0**	9	12	10	12	12	11	12	12	12	12	12

5.2.2 Results on WFG Problems

Table 12 presents the median IGD metric values of each 15 independent run for each algorithm on WFG problems. Like in Table 10, NSGA-III's normalization helped M2M variables (v2, v3, and v13) to perform better than their non-normalization counterparts. Similar performance is also observed with MOEA/D-N. Many of these methods now exhibit comparable performance to the original NSGA-III, which was not possible without any normalization, as presented in Table 7 before. Results of NSGA-III and MOEA/D variants – NSGA-III-0.7, MOEA/D-0.1, and MOEA-D/N-0.1, presented in Table 12 – are discussed next.

6 NSGA-III and MOEA/D Variants and Results

Motivated by the effect of explicit parallelism on M2M variants with and without normalization, we attempt to modify the original NSGA-III and MOEA/D procedure so that a similar explicit parallelism can be established in both of them. First, we present NSGA-III variants.

6.1 NSGA-III Variants

To introduce different degrees of explicit parallelism to NSGA-III, we modify the original NSGA-III by restricting two parents for recombination within the same associated reference line (or, establishing a mating restriction) with a certain probability, α. The original NSGA-III algorithm randomly selected two parents from the entire population for recombination, thereby allowing a complete parallelism to be played. If such a procedure causes an overly flexible search, this can be controlled explicitly by restricting the recombination to take place between two close-by solutions. This situation is opposite to the restriction faced in the original M2M algorithm. Our original NSGA-III is declared to have a zero probability of mating exclusively with same niched solutions and is open to mate with any other population member. In NSGA-III-α variant, a parent mates with any other population member with a probability $(1-\alpha)$ and mates with a member from its own subproblem (a similar subproblem concept in M2M introduced in NSGA-III) with the remaining probability α. We test the performance of 11 NSGA-III variants including the original one ($\alpha = 0$) on DTLZ and WFG problems.

Table 13 and Table 14 show the Wilcoxon rank-signed test results of 15 sets of the obtained solutions for each NSGA-III variant on DTLZ and WFG problems, respectively. The first row of Table 13 and Table 14 indicates the probability, α. For DTLZ problems, we observe from Table 13 that NSGA-III variant with the probability of $\alpha = 0.7$ of recombining with a similar solution produces the best performance. Interestingly, similar phenomenon can is also be observed in Table 14 for all WFG problems.

Tables 4 to 6 present the median IGD metric values for NSGA-III-0.7 for DTLZ and WFG problems.

Table 9: Median IGD values of NSGA-III, MOEA/D, MOEA/D-S, original M2M and the top three M2M variants with simple normalization on DTLZ problems.

Method	DTLZ1-3	DTLZ1-5	DTLZ2-3	DTLZ2-5	DTLZ3-3	DTLZ3-5	DTLZ4-3	DTLZ4-5
NSGA-III	0.002447	0.003372	0.001878	0.006894	0.004459	0.009924	*0.000836*	0.001176
NSGA-III-0.7	0.009922	0.001783	0.001593	0.004666	0.005160	0.008945	*0.001000*	0.001030
MOEA/D	**0.001525**	**0.000486**	**0.000540**	**0.000919**	**0.002934**	**0.001819**	*0.000133*	**0.000082**
MOEA/D-S	0.003064	0.492988	*0.000766*	*0.001029*	1.323132	1.124727	**0.000052**	*0.000085*
M2M-O	0.007047	0.001015	0.001078	0.001863	0.030759	0.017584	*0.000131*	0.000348
M2M-S-v11	0.030108	0.464759	0.006718	0.006218	0.430047	1.090534	*0.006565*	0.004929
M2M-S-v12	0.026674	0.469530	0.006463	0.005820	0.205835	1.102464	*0.007041*	0.005075
M2M-S-v13	0.030235	0.465849	0.006795	0.005189	0.010619	1.099218	*0.007140*	0.004773

Method	DTLZ1-8	DTLZ1-10	DTLZ1-15	DTLZ2-8	DTLZ2-10	DTLZ2-15
NSGA-III	0.006439	0.004012	**0.005126**	0.017261	0.017623	0.019434
NSGA-III-0.7	**0.002841**	**0.002185**	*0.005364*	0.007811	0.007811	0.014239
MOEA/D	0.003949	0.005122	0.015709	**0.002035**	**0.002879**	**0.005962**
MOEA/D-S	0.529817	0.539642	0.582147	1.228621	1.264533	1.323132
M2M-O	0.011439	0.011507	0.048595	0.007692	0.006900	0.010018
M2M-S-v11	0.526998	0.534093	0.582132	0.007810	0.008652	0.430047
M2M-S-v12	0.527336	0.533610	0.582128	0.007713	0.008035	0.205835
M2M-S-v13	0.521615	0.531582	0.582124	0.007597	0.007261	0.010619
Method	DTLZ3-8	DTLZ3-10	DTLZ3-15	DTLZ4-8	DTLZ4-10	DTLZ4-15
NSGA-III	0.034200	0.019540	*0.029303*	0.004183	0.004802	0.006760
NSGA-III-0.7	0.018117	0.008192	**0.026173**	0.002360	*0.002111*	0.007780
MOEA/D	**0.008868**	**0.005157**	*0.962856*	*0.001148*	0.002246	0.306230
MOEA/D-S	1.230680	1.264697	1.324349	**0.001097**	**0.002109**	0.306230
M2M-O	0.031047	0.013649	*0.037458*	0.004516	0.004388	0.013238
M2M-S-v11	1.226945	1.260186	1.323221	0.004594	0.005107	*0.004776*
M2M-S-v12	1.228679	1.262320	*1.320097*	0.004569	0.004665	**0.004291**
M2M-S-v13	1.228593	1.263071	1.323239	0.004984	0.004808	0.007054

While the original NSGA-III could not match well with MOEA/D and certain M2M variants on unscaled DTLZ problems, with 70% restriction of parent selection within the same subproblem (or niche) allows it to have the best performance in 8 and 10-objective DTLZ1 problems and better performance than original NSGA-III in many other problems. For WFG problems, NSGA-III-0.7's performance is best on most problems and is better or equivalent than the original NSGA-III on all problems. The developers of NSGA-III did not consider the possibility of a more restrictive parent selection (or mating restriction) procedure and is found to be a useful property in this study.

The above results indicate that the original NSGA-III introduced too much flexibility for any two parents to be recombined to produce effective offspring solutions. For many-objective optimization problems having a large dimensional objective space, it turns out to be too generic and may have consumed unnecessarily more solution evaluations to get close to the Pareto-optimal front. Motivated by the idea of explicit parallelism introduced by making the recombination more flexible and more parallel in M2M algorithms, we are able to develop an opposite and a more restrictive version of NSGA-III's recombination operation to reduce its overly flexible search power. Of all the parameters tried, a 70% restriction of recombination within the same niche and the rest 30% recombination to the entire population turns out to be a good compromise to other exploitation-exploration providing operators of NSGA-III.

6.2 MOEA/D Variants

So far, we are able to improve performance of original M2M and NSGA-III algorithms by explicitly controlling the parallelism introduced by their selection and recombination operators. Now, it is the turn of MOEA/D algorithm. The original MOEA/D uses its neighborhood update and recombination operation among solutions coming from a few neighboring reference lines 90% of the time and used the entire population the remaining 10% of the time. Here, we make this parameter to vary from 0% to 100% and create 11 MOEA/D-variants. Tables 15 and 16 show the Wilcoxon rank-sum outcomes for DTLZ and WFG problems, respectively. MOEA/D is applied without any normalization here. The best performing algorithm is marked with a zero and in bold. All other variants which makes an insignificant performance difference with $p < 0.05$ are marked with a zero, but all significantly worse performing algorithms are marked

Table 10: Median IGD values of NSGA-III, MOEA/D, MOEA/D-S, original M2M and the top three M2M variants with simple normalization on WFG problems.

Method	WFG5-3	WFG5-5	WFG6-3	WFG6-5	WFG7-3	WFG7-5	WFG8-3	WFG8-5
NSGA-III	**0.030312**	*0.032270*	*0.032821*	*0.034340*	*0.007695*	0.009096	0.055666	0.088318
NSGA-III-0.7	*0.030362*	**0.031869**	**0.032591**	**0.032464**	**0.007145**	**0.007812**	0.054998	0.082672
MOEA/D	0.068115	0.209306	0.072426	0.281740	0.076652	0.279958	0.078439	0.240437
MOEA/D-S	0.035975	0.032449	0.045477	0.036901	0.036095	0.009524	**0.054071**	**0.060853**
M2M-O	0.068861	0.178420	0.075951	0.180331	0.079863	0.179443	0.087754	0.194563
M2M-S-v2	0.036710	0.034293	0.050889	0.041038	0.031133	0.011467	0.074262	0.081592
M2M-S-v3	0.037076	0.034076	0.048549	0.038063	0.025557	0.010985	0.077658	0.085262
M2M-S-v13	0.036100	0.032752	0.048157	0.036146	0.028044	0.009439	0.072120	0.073859

Method	WFG5-8	WFG5-10	WFG5-15	WFG6-8	WFG6-10	WFG6-15
NSGA-III	0.031780	0.033264	*0.034340*	**0.030175**	0.033839	0.040254
NSGA-III-0.7	**0.031206**	**0.031330**	**0.032820**	*0.032464*	**0.028814**	**0.032825**
MOEA/D	0.560162	0.690478	0.907988	0.835942	0.935138	1.230484
MOEA/D-S	0.033340	0.033064	1.313734	1.220516	1.257855	1.319654
M2M-O	0.384840	0.543788	0.692934	0.479359	0.656128	0.825372
M2M-S-v2	0.048775	0.050177	0.052043	0.059423	0.060223	0.073087
M2M-S-v3	0.049857	0.047923	0.043620	0.059372	0.062252	0.050323
M2M-S-v13	0.037610	0.035842	*0.033084*	0.041063	0.038993	0.840959

Method	WFG7-8	WFG7-10	WFG7-15	WFG8-8	WFG8-10	WFG8-15
NSGA-III	**0.008356**	0.011790	0.011752	0.178498	0.234745	0.284391
NSGA-III-0.7	*0.008391*	**0.006762**	**0.010363**	0.171897	0.224020	**0.268413**
MOEA/D	0.808708	0.931775	1.104779	0.827976	0.982200	1.098366
MOEA/D-S	1.228617	1.264532	1.323131	1.228614	1.264531	1.323130
M2M-O	0.455113	0.647000	0.785423	0.440436	0.661559	0.811220
M2M-S-v2	0.026548	0.027855	0.031651	0.128932	0.156859	0.531418
M2M-S-v3	0.024652	0.028116	0.030074	0.130773	0.156787	0.526650
M2M-S-v13	0.014537	0.011528	1.144563	**0.108742**	**0.128405**	1.313875

with a one. For DTLZ problems, MOEA/D variant with 80% probability performed the best. This is similar to 90% probability used in the original MOEA/D in restricting neighborhood selection and recombination operators. However, Table 16 indicates that for scaled WFG problems MOEA/D-0.1 (without any normalization) with only 10% restriction of neighborhood selection and recombination within the same niche performs the best. For scaled problems, more parallelism needs to be introduced within the evolving population for the rest of the algorithm to work well. In fact, no restriction (MOEA/D-0) is also good, but the original MOEA/D-0.9 performs poorly.

The median IGD metric for MOEA/D-0.1 with and without NSGA-III's normalization for WFG problems are also presented in Table 12. Interestingly, MOEA/D-N-0.1 works significantly better for low-objective problems and is the fourth-best algorithm out of 36 different EMO algorithms considered in this extensive study.

7 Discussion on Explicit Parallelism on EMO Algorithms

In Figure 1, we started with a sketch of our initial idea of the levels of explicit parallelism present in popular evolutionary many-objective optimization algorithms. We argued that NSGA-III may constitute an excessive parallelism, while M2M provide a restrictive parallelism. Our extensive simulation studies on 36 algorithms of two sets of widely-used test problems – unscaled DTLZ and scaled WFG problems – and having three to 15 objectives have not only revealed that our intuition about the extent of parallelism in these algorithms was right, but the statistical Wilcoxon rank-sum tests have allowed us a way to rank these algorithms. Tables 9 to 12 are used to count the number of times an algorithm performs similar (with $p < 0.0001$) compared to the best performing algorithm for each problem. Figures 2 and 3 show the relative ranking of different algorithms on DTLZ and WFG problems, respectively. It is interesting to observe that the original MOEA/D performs much better on DTLZ problems due to uniform scaling of the objective values and any normalization (a simple or a more sophisticated) procedure deteriorates its performance. These normalization procedures depend on the population diversity at each generation, which, in principle, can vary from start to the end of a run. Since DTLZ problems do not require any additional normalization of objectives to make meaningful computation of Euclidean distances (d_1 and d_2) needed for PBI or other objective distance computations, any artificial normalization based on

Table 11: Median IGD values of NSGA-III, MOEA/D, MOEA/D-N, original M2M and the top three M2M variants with NSGA-III's normalization on DTLZ problems.

Method	DTLZ1-3	DTLZ1-5	DTLZ2-3	DTLZ2-5	DTLZ3-3	DTLZ3-5	DTLZ4-3	DTLZ4-5
NSGA-III	0.002447	0.003372	0.001878	0.006894	*0.004459*	0.009924	0.000836	0.001176
NSGA-III-0.7	0.009922	0.001783	0.001593	0.004666	*0.005160*	0.008945	0.001000	0.001030
MOEA/D	**0.001525**	**0.000486**	**0.000540**	**0.000919**	**0.004297**	**0.001819**	*0.000133*	**0.000082**
MOEA/D-N	0.003388	0.492988	0.002495	0.003462	0.010592	1.122972	0.000810	0.000440
M2M-O	0.007047	0.001015	0.001078	0.001863	0.030759	0.017584	**0.000131**	0.000348
M2M-N-v11	*0.002105*	0.000806	0.003754	0.005161	0.006161	0.004673	0.000819	0.000747
M2M-N-v12	*0.002174*	0.000759	0.002668	0.006258	0.007917	0.009633	0.000597	0.000776
M2M-N-v13	0.002959	0.002271	0.002610	0.005559	0.011175	0.007543	0.000773	0.001511

Method	DTLZ1-8	DTLZ1-10	DTLZ1-15	DTLZ2-8	DTLZ2-10	DTLZ2-15
NSGA-III	0.006439	0.004012	**0.005126**	0.017261	0.017623	0.019434
NSGA-III-0.7	**0.002841**	**0.002185**	*0.005364*	0.007811	0.007811	0.014239
MOEA/D	0.003949	0.005122	0.015709	**0.002035**	**0.002879**	**0.005962**
MOEA/D-N	0.529796	0.539650	0.582138	1.228621	1.264533	1.323132
M2M-O	0.011439	0.011507	0.048595	0.007692	0.006900	0.010018
M2M-N-v11	0.005339	0.005137	0.486882	0.014358	0.014386	0.033220
M2M-N-v12	0.004577	0.004883	0.528055	0.012854	0.014087	0.019192
M2M-N-v13	0.003754	0.003516	0.481213	0.012436	0.015678	0.014386

Method	DTLZ3-8	DTLZ3-10	DTLZ3-15	DTLZ4-8	DTLZ4-10	DTLZ4-15
NSGA-III	0.034200	0.019540	*0.029303*	*0.004183*	0.004802	0.006760
NSGA-III-0.7	0.018117	0.008192	**0.026173**	*0.002360*	0.002111	0.007780
MOEA/D	**0.008868**	**0.005157**	*0.962856*	*0.001148*	**0.001267**	0.306230
MOEA/D-N	1.228803	1.264568	1.323254	**0.001738**	0.002141	*0.003716*
M2M-O	0.031047	0.013649	*0.037458*	*0.004516*	0.004388	0.013238
M2M-N-v11	0.019850	0.009748	1.300411	*0.003526*	0.003229	0.005813
M2M-N-v12	0.016086	0.012543	1.211846	*0.003940*	0.002908	**0.003264**
M2M-N-v13	0.018066	0.012430	1.261727	*0.005615*	0.003293	*0.003653*

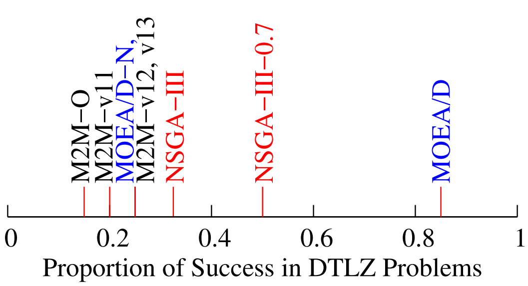

Figure 2: Proportion of successful performance over 20, three to 15-objective DTLZ problems.

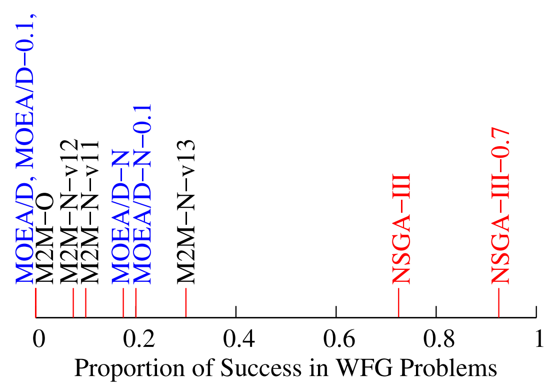

Figure 3: Proportion of successful performance over 20, three to 15-objective WFG problems.

statistics of changing populations makes a noisy evaluation of these distance measures. However, for M2M with simple normalization procedure and NSGA-III with its usual normalization procedure, their variants perform much better.

For real-world problems, a non-uniform scaling of objectives is likely. In this respect, WFG problems represent real-world scenarios better than DTLZ problems. Figure 3, showing performances on 20 scaled WFG problems, indicates that one of the NSGA-III variants works much better than the original NSGA-III and all other algorithms. For M2M and MOEA/D algorithms, their variants make a better balance of explicit and implicit parallelism aspects. While other algorithms are not able to match the performance of NSGA-III or its variant, MOEA/D with its NSGA-III-based normalized version and its relaxed variant perform much better than the original MOEA/D. Similarly, one of the M2M variants with its NSGA-III-based normalization procedure performed much better than original M2M. Interestingly, the explicit control of parallelism (relaxation in M2M and MOEA/D, and

Table 12: Median IGD values of NSGA-III, MOEA/D, MOEA/D-N, original M2M and the top three M2M variants NSGA-III's normalization on WFG problems.

Method	WFG5-3	WFG5-5	WFG6-3	WFG6-5	WFG7-3	WFG7-5	WFG8-3	WFG8-5
NSGA-III	**0.030312**	*0.032270*	**0.032546**	*0.034340*	*0.007695*	0.009096	*0.055666*	0.088318
NSGA-III-0.7	*0.030362*	**0.031869**	*0.032591*	**0.032464**	**0.007145**	**0.007812**	**0.054998**	*0.082672*
MOEA/D	0.068115	0.209306	0.072426	0.281740	0.076652	0.279958	0.078439	0.240437
MOEA/D-N	0.041079	0.032088	0.048323	*0.033825*	0.030538	*0.008214*	0.061934	**0.082357**
MOEA/D-0.1	0.067690	0.186369	0.163470	0.349031	0.071192	0.231133	0.080456	0.220772
MOEA/D-N-0.1	0.035041	*0.032288*	0.041970	*0.032015*	0.029158	0.009062	0.064772	0.087351
M2M-O	0.068986	0.178531	0.077098	0.180466	0.081060	0.179978	0.086545	0.194794
M2M-N-v2	0.036036	0.036460	0.045960	*0.032636*	0.025624	0.009160	0.079412	0.083883
M2M-N-v3	0.035217	0.035359	0.041647	*0.034928*	0.026428	0.008296	0.079294	0.085786
M2M-N-v13	0.034054	0.035623	0.047766	*0.033848*	0.028705	*0.008140*	0.083459	0.084037

Method	WFG5-8	WFG5-10	WFG5-15	WFG6-8	WFG6-10	WFG6-15
NSGA-III	0.031780	0.033264	*0.032821*	**0.030175**	0.033839	0.040254
NSGA-III-0.7	**0.031206**	**0.031330**	**0.032820**	*0.032464*	**0.028814**	**0.032825**
MOEA/D	0.560162	0.690478	0.907988	0.835942	0.935138	1.230484
MOEA/D-N	0.032499	0.033141	1.313735	1.220100	1.257918	1.319730
MOEA/D-0.1	0.541315	0.660125	0.896406	0.821414	0.886578	1.073753
MOEA/D-N-0.1	0.031811	0.032144	1.313735	0.039094	1.258331	1.319627
M2M-O	0.391055	0.546182	0.686288	0.462567	0.652632	0.836909
M2M-N-v2	0.036460	0.037930	0.047584	0.040046	0.186754	0.946065
M2M-N-v3	0.035738	0.033427	0.129783	0.037419	0.060202	1.313267
M2M-N-v13	0.035623	0.033270	0.128241	0.038369	0.035558	1.313238

Method	WFG7-8	WFG7-10	WFG7-15	WFG8-8	WFG8-10	WFG8-15
NSGA-III	**0.008356**	0.011790	0.011752	0.178498	0.234745	0.284391
NSGA-III-0.7	*0.008391*	**0.006762**	**0.010363**	0.171897	**0.224020**	**0.268413**
MOEA/D	0.808708	0.931775	1.104779	0.827976	0.982200	1.098366
MOEA/D-N	1.228616	1.264532	1.323131	1.228614	1.264531	1.323130
MOEA/D-0.1	0.687692	0.754708	1.078039	0.727818	0.826891	1.122822
MOEA/D-N-0.1	0.015561	1.264533	1.323131	1.228614	1.264531	1.323130
M2M-O	0.449549	0.641922	0.770839	0.407763	0.666609	0.800532
M2M-N-v2	0.019850	0.009748	1.072646	**0.154454**	0.983667	1.284147
M2M-N-v3	1.228620	1.259566	1.214997	*0.155159*	0.715033	1.221974
M2M-N-v13	1.228644	1.263786	1.263649	*0.154924*	0.672806	0.882245

restriction in NSGA-III) enabled much better performances to be achieved on WFG problems.

8 Conclusions

In this paper, we have argued that a proper balance between the extent of explicit parallelism introduced by the developer and the inherent implicit parallelism introduced by evolutionary operators is the key to the success of an EMO algorithm. Although it is difficult to exactly quantify these extents in terms of their associated parameters (such as probabilities used in this paper) for a particular EMO algorithm, finding the proper balance between the two parallelisms can be made through careful experimentations illustrated in this paper. The operators used in evolutionary many-objective optimization algorithms may have made the implicit parallelism effect too flexible for them to work well on high-dimensional problems. In this case, a more restrictive parallelism controlled externally may control the generality of implicit parallelism. On the other hand, when the evolutionary operators are designed to have a more focused approach with little implicit parallelism effect on the population, a more relaxed explicit parallelism mechanism should make a better balance. In this study, we have been able to enhance the performance of three popular EMO algorithms – MOEA/D, M2M and NSGA-III – by using a more relaxed explicit parallelism effect to the first two algorithms and by using a restricted explicit parallelism effect compared to their original versions.

Specifically, 14 variants of MOEA/D-M2M, 11 variants of NSGA-III, and 11 variants of MOEA/D with different degrees of parallelism in their selection and recombination operators have been proposed and evaluated with their original methods on two classes of test problems – three to 15-objective, unscaled DTLZ and scaled WFG problems. Following conclusions can be drawn from this extensive simulation study.

- Most variants of M2M with relaxed selection and recombination operators have performed better than the original M2M. This means that for

Table 13: Wilcoxon rank-sum test values for the variants of NSGA-III on DTLZ test problems.

Instance	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
DTLZ1-3	0	0	**0**	0	0	0	0	1	1	1	1
DTLZ1-5	1	1	0	**0**	0	1	0	0	0	0	0
DTLZ1-8	1	1	1	1	1	1	0	**0**	0	1	1
DTLZ1-10	1	1	1	1	0	**0**	0	0	0	0	0
DTLZ1-15	0	1	1	0	0	0	1	0	0	**0**	0
DTLZ2-3	0	0	0	0	0	0	0	**0**	0	0	0
DTLZ2-5	1	0	0	0	0	**0**	1	0	0	0	0
DTLZ2-8	1	1	1	1	1	1	1	0	0	0	**0**
DTLZ2-10	1	1	1	1	1	1	1	0	0	0	**0**
DTLZ2-15	1	1	1	1	0	0	0	**0**	0	0	0
DTLZ3-3	0	0	0	0	0	**0**	0	0	0	0	0
DTLZ3-5	0	0	0	0	**0**	0	0	0	0	0	1
DTLZ3-8	1	1	1	1	1	0	0	0	1	1	**0**
DTLZ3-10	0	0	0	0	0	0	0	0	0	**0**	0
DTLZ3-15	0	0	0	0	0	0	0	0	**0**	0	0
DTLZ4-3	0	0	**0**	0	0	0	0	0	1	1	1
DTLZ4-5	0	0	0	0	**0**	0	0	0	0	0	0
DTLZ4-8	1	1	1	1	0	0	0	0	0	0	**0**
DTLZ4-10	1	1	1	1	1	1	1	1	0	**0**	0
DTLZ4-15	0	**0**	0	0	0	0	0	0	0	1	1
Sum	10	10	9	8	5	5	5	**2**	3	5	5

Table 14: Wilcoxon rank-sum test values for the variants of NSGA-III on WFG test problems.

Instance	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
WFG5-3	0	0	0	1	0	**0**	0	0	0	0	0
WFG5-5	1	1	1	1	1	0	**0**	0	0	0	1
WFG5-8	1	1	1	1	1	1	0	0	**0**	0	1
WFG5-10	1	1	1	1	1	1	1	**0**	1	1	1
WFG5-15	0	1	0	0	0	**0**	0	1	0	1	1
WFG6-3	0	0	0	1	0	1	0	0	**0**	0	1
WFG6-5	1	1	1	0	1	0	0	**0**	1	0	1
WFG6-8	1	1	0	0	0	1	0	0	0	**0**	1
WFG6-10	1	1	0	0	0	**0**	0	0	0	0	1
WFG6-15	0	1	0	0	0	0	0	**0**	0	0	0
WFG7-3	0	0	0	0	**0**	0	0	0	0	1	0
WFG7-5	1	1	1	1	1	0	0	**0**	0	0	1
WFG7-8	0	0	1	0	**0**	0	0	0	0	0	0
WFG7-10	1	1	1	1	1	0	0	**0**	0	0	1
WFG7-15	0	1	0	0	0	0	0	**0**	0	1	0
WFG8-3	1	1	1	1	1	1	0	0	**0**	1	1
WFG8-5	1	1	1	1	1	1	1	0	0	**0**	1
WFG8-8	1	1	1	0	0	0	0	0	0	**0**	1
WFG8-10	1	1	1	1	0	1	0	0	0	**0**	1
WFG8-15	1	0	1	1	0	0	0	0	**0**	0	1
Sum	13	15	12	11	8	7	2	**1**	2	5	15

Table 15: Wilcoxon rank-sum test values for the variants of MOEA/D on DTLZ test problems.

Instance	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
DTLZ1-3	0	0	0	0	**0**	0	0	0	0	0	1
DTLZ1-5	1	1	1	1	0	0	1	**0**	0	0	1
DTLZ1-8	1	1	1	1	**0**	1	1	1	1	1	1
DTLZ1-10	1	1	1	1	**0**	0	1	1	1	1	1
DTLZ1-15	1	0	0	**0**	1	0	1	1	1	1	1
DTLZ2-3	1	1	0	**0**	1	0	0	0	1	0	0
DTLZ2-5	1	1	1	1	0	1	1	1	0	0	**0**
DTLZ2-8	1	1	1	1	1	1	1	1	0	0	**0**
DTLZ2-10	1	1	1	1	1	1	0	0	**0**	0	1
DTLZ2-15	1	0	0	0	0	0	**0**	0	0	0	1
DTLZ3-3	1	1	0	1	**0**	0	0	0	0	0	0
DTLZ3-5	1	1	1	1	0	1	0	1	**0**	1	1
DTLZ3-8	1	1	1	1	**0**	1	1	0	0	1	1
DTLZ3-10	1	1	1	1	0	1	**0**	1	0	1	1
DTLZ3-15	1	1	1	0	**0**	1	0	1	0	1	1
DTLZ4-3	0	0	0	0	1	0	0	0	0	**0**	0
DTLZ4-5	1	1	1	1	1	1	1	1	0	**0**	0
DTLZ4-8	0	0	0	1	0	1	**0**	0	0	0	1
DTLZ4-10	0	0	0	0	0	1	**0**	0	0	0	1
DTLZ4-15	0	0	**0**	0	0	0	1	0	0	0	0
Sum	15	13	11	12	6	11	9	9	**4**	7	13

Table 16: Wilcoxon rank-sum test values for the variants of MOEA/D on WFG test problems.

Instance	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
WFG5-3	1	0	0	0	0	0	0	**0**	1	1	1
WFG5-5	**0**	0	0	0	1	1	1	1	1	1	1
WFG5-8	1	**0**	1	1	1	1	1	1	1	1	1
WFG5-10	0	0	0	**0**	1	0	0	1	1	1	1
WFG5-15	**0**	0	1	0	1	0	1	1	1	1	1
WFG6-3	0	0	0	0	0	0	0	0	**0**	0	0
WFG6-5	**0**	0	0	0	1	1	1	1	1	1	1
WFG6-8	**0**	0	1	1	1	1	1	1	1	1	1
WFG6-10	0	0	**0**	1	1	1	1	1	1	1	1
WFG6-15	0	**0**	0	0	1	0	0	0	1	1	1
WFG7-3	**0**	0	0	0	0	0	0	1	1	1	1
WFG7-5	**0**	0	0	1	1	1	1	1	1	1	1
WFG7-8	**0**	0	0	1	1	1	1	1	1	1	1
WFG7-10	**0**	0	1	1	1	1	1	1	1	1	1
WFG7-15	**0**	0	0	1	0	0	0	0	1	0	0
WFG8-3	1	1	1	0	1	0	0	**0**	0	0	0
WFG8-5	0	**0**	0	0	0	1	1	1	1	1	1
WFG8-8	0	0	0	**0**	0	0	1	0	1	1	1
WFG8-10	0	0	**0**	0	0	1	1	1	1	1	1
WFG8-15	**0**	0	1	0	0	0	0	0	0	0	0
Sum	3	**1**	6	7	12	10	12	13	17	16	16

M2M, a more relaxed explicit parallelism makes a better balance with the implicit parallelism introduced by M2M's other operators.

- For M2M, relaxing recombination to take place among the any two population members (rather than from the same subpopulation) for 33% to 67% of the time has been found to be most beneficial. For WFG problems, relaxing the update procedure for a child to replace a parent from the entire population, rather than a parent from the same subproblem, has also been found as a better strategy.

- MOEA/D's original suggestion of neighborhood update and recombination operators to be restricted within a solution's own neighborhood around 90% of the time has turned out to produce an optimal balance between explicit and implicit parallelisms on DTLZ problems, where MOEA/D performs well.

- NSGA-III's recombination operator has produced the best outcome when a parent is restricted to mate 70% of the time with another individual from its own neighboring reference line, rather than picking its partner from the entire population. The original NSGA-III did not implement this restricted parallel variant and our study here has shown consistently better results than the original and more flexible NSGA-III.

- For scaled WFG problems, our systematic parametric study is able to improve the performance of all three original methods. While in the cases of MOEA/D and M2M, the operators were needed to be made more flexible for a better performance, in the case of NSGA-III, it was the opposite – its recombination operator was needed to be more restrictive for it to work better.

There is another investigation which has revealed interesting conclusions. DTLZ class of problems involve objectives which have similar range of values: each objective lies in [0,1] for Pareto-optimal solutions. These problems do not require any additional normalization of objectives during the optimization process. In fact, if any normalization based on statistics of evolving populations is performed at every generation, they will change the relative location of objective vectors with respect to specified reference directions from one generation to another. Since the PBI metric (used in both MOEA/D and M2M) needs two distance metric values (d_1 and d_2) and that their weighted combination determines the entire population update procedure, the stability of the normalization process becomes a very important matter for these two algorithms. We have observed here that while MOEA/D method performs the best over all other 35 algorithms used in this study on DTLZ class of problems, it has performed the worst among all 36 algorithms on WFG problems. Also MOEA/D with either a simple or a more sophisticated normalization procedure has performed worse than MOEA/D without any normalization on DTLZ problems. Thus, the clear advantage of MOEA/D in DTLZ problems comes from the natural unscaled nature of DTLZ problems. Since real-world problems are likely to have scaled objectives, the results on WFG problems are more relevant. It has been observed that the original MOEA/D and M2M methods and their variants have enhanced their performances with NSGA-III's normalization procedure on WFG problems.

Despite making a controlled study on explicit parallelism using probabilities of selection and recombination, results on WFG problems have revealed that NSGA-III and its variant perform much better than MOEA/D and M2M variants even with a sophisticated normalization procedure. In our future studies, we plan to make further studies involving other MOEA/D parameters, such as θ and r_s. A similar study on constrained problems will be another useful future study. Nevertheless, this extensive study clearly brings out the need for making a balance in introducing explicit and implicit parallelisms in an algorithm in restricting or relaxing the search through a decomposition approach, which is a current trend in EMO research. The approach of this paper has helped create better algorithms than the original methods. The proposed approach is generic and can be applied to other EMO algorithms. More such studies will provide a deeper understanding of the trade-off between supplied and inherent parallelisms, which is crucially important in developing computationally efficient evolutionary many-objective optimization algorithms.

References

[1] J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation Journal*, 19(1):45–76, 2011.

[2] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[3] R. Cheng, Y. Jing, M. Olhofer, and B. Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791, 2016.

[4] Y.-M. Cheung, F. Gu, and H.-L. Liu. Objective extraction for many-objective optimization problems: Algorithm and test problems. *IEEE Transactions on Evolutionary Computation*, 20(5):755–722, 2016.

[5] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 283–290. San Mateo, CA: Morgan Kaufmann Publishers, 2001.

[6] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.

[7] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[9] 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 Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

[10] K. Deb, M. Mohan, and S. Mishra. Towards a quick computation of well-spread Pareto-optimal solutions. In *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632)*, pages 222–236, 2003.

[11] K. Deb, A. Sinha, and S. Kukkonen. Multi-objective test problems, linkages and evolutionary methodologies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)*, pages 1141–1148. New York: The Association of Computing Machinery (ACM), 2006.

[12] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multi-objective Optimization*, pages 105–145. London: Springer-Verlag, 2005.

[13] F. di Pierro, S.-T. Khu, and D. A. Savic. An investigation on preference order ranking scheme for multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 11(1):17–45, 2007.

[14] A. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3:124–141, 1999.

[15] M. Farina and P. Amato. A fuzzy definition of optimality for many criteria optimization problems. *IEEE Trans. on Systems, Man and Cybernetics Part A: Systems and Humans*, 34(3):315–326, 2004.

[16] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. San Mateo, CA: Morgan Kaufmann, 1993.

[17] C. M. Fonseca and P. J. Fleming. Multiobjective genetic algorithms made easy: Selection sharing and mating restriction. In *Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 45–52, 1995.

[18] D. E. Goldberg. *Genetic Algorithms for Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[19] R. H. Gomez and C. Coello Coello. MOMBI: A new metaheuristic for many-objective optimization based on the r2 indicator. In *Proceedings of IEEE Congress on Evol. Computation*, pages 2488–2495, 2013.

[20] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. C. Coello, and J. Molina. Pareto adaptive ϵ-dominance. *Evolutionary Computational Journal*, 15:493–517, 2007.

[21] J. H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: MIT Press, 1975.

[22] J. Horn, N. Nafploitis, and D. E. Goldberg. A niched Pareto genetic algorithm for multi-objective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87, 1994.

[23] S. Huband, L. Barone, L. While, and P. Hingston. A scalable multi-objective test problem toolkit. In *Proceedings of the Evolutionary Multi-Criterion Optimization (EMO-2005)*. Berlin, Germany: Springer, 2005.

[24] K. Ikeda, H. Kita, and S. Kobayashi. Failure of pareto-based moeas: Does non-dominated really mean near to optimal? In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 957–962, 2001.

[25] H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and reviews*, 28(3):392–403, 1998.

[26] H. Ishibuchi and Y. Shibata. A similarity-based mating scheme for evolutionary multiobjective optimization. In *Genetic and Evolutionary Computation Conference (GECCO-2003)*, pages 1065–1076. Springer, 2003.

[27] A. L. Jaimes, C. A. C. Coello, and D. Chakraborty. Objective reduction using a feature selection technique. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2008)*, pages 673–680, 2008.

[28] A. L. Jaimes, C. A. Coello Coello, H. Aguirre, and K. Tanaka. Adaptive objective space partitioning using conflict information for many-objective optimization. In *Proceedings of Evolutionary Multi-Criterion Optimization*, pages 151–165, 2011.

[29] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2014.

[30] S. Jiang, J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan. A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Transactions on Evolutionary Cybernetics*, 45(10):2202–2213, 2014.

[31] G. Karafotias, M. Hoogendoorn, and A. E. Eiben. Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 19:167–187, 2015.

[32] V. Khare, X. Yao, and K. Deb. Performance scaling of multi-objective evolutionary algorithms. In *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632)*, pages 376–390, 2003.

[33] M. Köppen and K. Yoshida. Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In *Proceedings of Fourth International Conference on Multi-Objective Optimization, EMO 2007 (LNCS 4403)*, pages 727–741. Heidelberg, Germany: Springer-Verlag, 2007.

[34] H. Li and Q. Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans on Evolutionary Computation*, 12(2):284–302, 2009.

[35] K. Li, K. Deb, Q. Zhang, and S. Kwong. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716, 2015.

[36] M. Li, S. Yang, and X. Liu. A comparative study on evolutionary algorithms for many-objective optimization. In *Proceedings of Evolutionary Multi-Criterion Optimization*, pages 261–275, 2013.

[37] H.-L. Liu, F. Gu, and Q. Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):773–791, 2013.

[38] K. Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer, Boston, 1999.

[39] L. M. S. Russo and A. P. Francisco. Quick hypervolume. *IEEE Transactions on Evolutionary Computation*, 18(4):481–502, 2014.

[40] D. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang. Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *IEEE Transactions on Evolutionary Computation*, 77(1):77–99, 2013.

[41] N. Srinivas and K. Deb. Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation Journal*, 2(3):221–248, 1994.

[42] Kay Chen Tan, Tong Heng Lee, and Eik Fun Khor. Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 5(6):565–588, 2001.

[43] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Proceedings of Evolutionary Multi-Criterion Optimization*, pages 742–756, 2007.

[44] H. Wang, L. Jiao, and X. Yao. Two_arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Cybernetics*, 19(4):524–541, 2015.

[45] L. While, P. Hingston, L. Barone, and S. Huband. A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1):29–38, 2006.

[46] S. Yang, M. Li, X. Liu, and J. Zheng. A grid-based evolutionary algorithm for many-objective

optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736, 2013.

[47] Y. Yuan, H. Xu, B. Wang, and X. Yao. Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, 20(2):180–198, 2016.

[48] Y. Yuan, H. Xu, B. Wang, and X. Yao. A new dominance relation based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(1):16–37, 2016.

[49] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, 2007.

[50] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou et al., editor, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2001.

[51] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.