A Distributed Framework for Cooperation of Many-Objective Evolutionary Algorithms

Gian Fritsche
Department of Computer Science
Federal University of Paraná
Curitiba, Paraná, Brazil
gmfritsche@inf.ufpr.br

Aurora Pozo
Department of Computer Science
Federal University of Paraná
Curitiba, Paraná, Brazil
aurora@inf.ufpr.br

ABSTRACT

The simultaneous optimization of multiple objectives arises in several problems in different disciplines. This optimization, mainly for many-objective problems brings challenges to the state-of-the-art Multi-Objective Evolutionary Algorithms. Given the various characteristics of the different problem instances and also the features of the algorithms, no single algorithm performs well in all problem instances. Although, if the algorithms characteristics could be combined, cooperatively, to face the problem together, the search ability can be improved. In this work, we evaluate this research question and propose a distributed framework for cooperation of Many-objective Evolutionary Algorithms. In the framework, different algorithms can be executed simultaneously, with small sub-populations and collectively solve the problem instance. The framework performs the cooperation by sharing solutions between the subpopulations. In this way, sharing the information learned from one algorithm to the other. The framework is evaluated using two state-of-the-art algorithms for cooperation, and compared to the algorithms executed alone. The results indicate that the cooperation improves the convergence and diversity of the algorithms in most problem instances. The obtained results motivate the investigation of future works on the proposed framework.

CCS CONCEPTS

•**Computing methodologies** → *Continuous space search; Cooperation and coordination;* •**Mathematics of computing** → *Continuous optimization;* •**Theory of computation** → *Evolutionary algorithms;* Distributed algorithms;

KEYWORDS

Distributed Multi-objective Algorithm, Many-Objective Optimization, NSGA-III, MOEA/D-STM, Multi-Population, Multi-Strategy

ACM Reference format:
Gian Fritsche and Aurora Pozo. 2017. A Distributed Framework for Cooperation of Many-Objective Evolutionary Algorithms. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3084213

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3084213

1 INTRODUCTION

Several real-world problems involve the simultaneous optimization of two or more objectives. They are usually conflicting, which difficulties the optimization of one without decreasing the quality of the other. The aim of the Multi-Objective Evolutionary Optimization is to find an approximation set of non-comparable (or non-dominated) solutions. Those solutions should be as close as possible to the non-dominated front of the problem (the Pareto front set), and well diversified, comprising solutions with different trade-offs between the objectives [3].

The Evolutionary Algorithms (EAs) were adapted to Multi-Objective Optimization (MOO), creating the Multi-Objective Evolutionary Algorithms (MOEAs). Besides their good results in the single-objective, another characteristic contributes to the use of EAs for MOO: the ability to output a set of solutions in a single run. This ability allows the algorithm to output, in a single execution, a Pareto set approximation, with several solutions representing different trade-offs between the objectives [3].

The adaptation of EAs to MOO was capable of achieving success when applied to multi-objective problems (with two or three). Although, problems with more than three objectives arise naturally in several disciplines, and the MOEAs face challenges when applied for many-objective optimization [6, 13]. One of the difficulties in handling many-objective problems is that a significant portion of the population is non-comparable, affecting the quality of the selection of solutions to guide the search. Also, the recombination operator may be inefficient, as it often generates solutions widely different from the parents due to the large-dimensional space. Another difficulty is the increasing of the computational cost to compute quality measures, such as diversity and convergence. It is also difficult for the decision maker, first because a significant number of solutions is necessary to represent the different trade-offs between the objectives. Further, the visual comparison of solutions in a high dimension may be difficult [6].

There are various algorithms for many-objective optimization in the literature [13]. Although, according to the No Free Lunch Theorem [19], if someone executes all algorithms for all problems, in average, they would all perform similarly. However, no single algorithm can outperform in all problems. That is, the algorithms perform differently depending on the problem characteristics and the number of objectives. The aim of this paper is to join the strengths of different algorithms in a cooperative way. For this goal, we propose a distributed framework for cooperation of Many-objective Evolutionary Algorithms (COMOEA). The COMOEA distributes the population between the algorithms creating small subpopulations, each one executing a different MOEA. The information learned using one algorithm can be used to improve the others, using an information exchange policy. Also, the cooperation of strategies or parameter settings increases the robustness of global search on different problem characteristics [1, 4]. The COMOEA framework is a multi-population and multi-strategy approach for many-objective optimization.

In this work, two state-of-the-art MOEAs were selected to assess the quality of the proposed framework: NSGA-III [6] and MOEA/D-STM [16]. We compare the cooperation between those MOEAs, using the proposed framework, to the MOEAs executed alone. The performed experiments use 3 to 15 objectives, in 6 different problem instances, representing different MOP characteristics. Two employed quality measures, IGD and Hypervolume, assessed the convergence and diversity of the generated results.

In most problem instances, the cooperation between the algorithms using the proposed approach was the best or equivalent than the MOEAs executed alone, for both IGD and Hypervolume. The obtained results indicate that the proposed framework is capable of combining the strengths of MOEAs, improving the quality of the output, and encourages further investigation on the topic. The following sections present initially some related works and background about many-objective optimization, hybrid approaches, and distributed evolutionary algorithms at Section 2. Section 3 describes the proposed framework, the general procedure, and details. Then, at Section 4, the COMOEA is instantiated using two MOEAs, NSGA-III and MOEA/D-STM. The section presents briefly the MOEAs used for cooperation and some implementation details. The experimental setup is shown next, in Section 5, such as the number of objectives, population sizes, benchmark problems and quality measures. Finally, Section 6 presents the average results of the evaluated algorithms, for each problem instance and quality indicator. Besides, it presents also a general comparison among the algorithms. At last, the major conclusions arrives in Section 7 along with some topics for further research.

2 LITERATURE REVIEW

This section presents some literature review and background about the topics related to this work. First, we present a review of the state-of-the-art many-objective optimization. Then, it is described some works on hybrid evolutionary algorithms for many-objective. Finally, we show a short review about distributed evolutionary algorithms.

2.1 Many-objective Optimization

There are different approaches for many-objective optimization in the literature [13]:

- The aggregation-based, such as MOEA/D, MOEA/D-DRA [20], MOEA/D-STM [16] and MOEA/DD [15]. They do not suffer from the selection pressure problem as they do not use Pareto dominance. However, they still suffer from the difficulty to search the space in high dimensionality ("curse of dimensionality"). Also, the construction of a well-spread set of points in a high dimension space may be an issue [13].
- The indicator based, such as SMS-EMOA, HypE, MOMBI-II, and IBEA [13]. They also do not use Pareto dominance, so, they do not suffer from the selection pressure problem. However, the computational cost is an issue when the indicator used is the Hypervolume, plus, the algorithm may prefer the knee points instead of a uniform distributed set. Although, other indicators may lead performance degradation due they are not strictly monotonic. Besides, they also suffer from the curse of dimensionality [13]. Li *et. al* [14] proposes a multi-indicator algorithm (SRA), since indicators may have different biases, one might complement the other. The SRA was implemented using two quality indicators, one for convergence and other for diversity. Besides, an archive strategy is capable of improving the performance of SRA [14].
- The preference-based approaches that can be split on: a priori, during the search and post optimization [13].
- The dimensionality reduction approach that tries to remove redundant objectives, which limits the application of the approach to problems with redundant objectives [13].
- Besides, there is also research on relaxed dominance, such as GrEA. A difficulty of these methods is the tuning of the relaxation for different problems [14].
- Also, there are the diversity based approaches, such as Shift-based Density Estimation (SDE) [14].
- Hybrid approaches, such as NSGA-III [6] that combines Pareto and aggregation [14]; and Two_Arch2 [17], where one archive is guided by a quality indicator while the other uses Pareto dominance and a distance measure.

Other research related to many-objective optimization may include customized mutation and crossover operators, and measures for diversity and convergence.

2.1.1 Many-objective Hybrid approaches. Since the proposed framework is a hybrid approach in this section, we emphasize some hybrid approaches from the literature.

- The Two_Arch2 [17], that uses two different selection principles to update two archives. One archive is intended to promote the convergence, using the additive epsilon indicator ($I_{\epsilon+}$) to select the solutions. The other archive is designed for diversity and stores the non-dominated solutions and prunes the archive with a proposed diversity maintenance scheme based on the L_p-norm-based distance.
- The PMEA [18], which keeps three different populations, guided by three distinct environment selection strategies: a decomposition-based (guided by the Penalty-based Boundary Intersection utility function - PBI), an indicator based ($I_{\epsilon+}$) and a Shift-based Density Estimation. First, the algorithm selects the parents randomly with equal probability for each population. Then the offspring set is generated by SBX and polynomial mutation. After that, each population is updated, in parallel, with the generated offspring, each one using its selection strategy.
- The HEA-DP proposes a dual population MOEA [21]. The Hybrid Evolutionary Algorithm with Dual Population combines decomposition and indicator based approaches. The environment selection for the first population is determined by an aggregation function (PBI), while for the other population it is defined by the additive epsilon indicator

($I_{\epsilon+}$). The parent selection is performed randomly selecting solutions from both populations.

The Two_Arch2 [17], PMEA [18], and HEA-DP [21] share similar characteristics for combining multiple strategies. All of them keeps multiple populations (or archives) updated by different quality measures. Also, they combine the information of the quality measures by selecting parents from the different populations. Another similarity is the selected quality measures used for environmental selection: the $I_{\epsilon+}$ for HEA-DP, PMEA, and Two_Arch; and the PBI utility function for PMEA and HEA-DP. Those works selected these metrics due to previous good results obtained in the literature.

2.2 Distributed Evolutionary Algorithms

The distributed evolutionary algorithms (dEA) for multi-objective optimization is a hot-spot in the dEA field [11]. The distribution allows to maintain the population diversity and avoiding local optima. This section presents some characteristics of previous works and an example of a multi-population distributed evolutionary algorithm.

- Many existing distributed MOEAs are based on NSGA-II, varying according to the model used [11].
- Also, some works have developed Distributed versions of MOEA/D [11].
- Besides, there is also works on distributed SPEA, VEPSO, and PSFGA [11].
- The coevolutionary MOEAs divides the decision vectors into subcomponents and evolves the different parts cooperatively [11].
- Although, most works apply the model homogeneous, applying the same algorithm for all subpopulations [11].

An example of multi-population distributed evolutionary algorithm is the AsAMPdDE, presented by Falco *et. al.* [10], that proposes an asynchronous adaptive multi–population model for Distributed Differential Evolution (AsAMP-dDE). The motivation is that the population partition explores the search space more evenly and preserves an overall higher diversity. The AsAMP-dDE is also adaptive, with the hope that the subpopulations with poor performance will eventually find a parameter set values capable of enhancing their performance. Every iteration, each subpopulation performs the information exchange using a migration policy. The algorithm was proposed for single objective problems and evaluated on the CEC2016 benchmark with good performance.

2.3 Discussion

In [13] it is suggested the research on hybrid algorithms, combining two or more approaches together. The approach proposed in this paper is one way to fulfill this suggestion, as it allows the execution of different approaches together. For instance, the performed experiments combine an aggregation approach MOEA/D-STM, with a hybrid (aggregation and Pareto) NSGA-III. In the case of using aggregation based approaches the decision maker preferences can be introduced a priori by placing the reference vectors on the preferred regions of the space. A two-layer weight vector generation method [15] can be used to generate a limited number of reference points, spread in the objective space and with intermediate points for many-objective problems.

Compared to other hybrid algorithms the COMOEA presents some differences and similarities. The Two_Arch2 [17], and PMEA [18], and HEA-DP [21] achieves good results by storing different archives (or populations) filtered by various selection approaches. Those works [17, 18, 21] demonstrates that the cooperation of different approaches can improve the MOEAs performance. They share with the proposed framework the characteristic of combining different strategies. One major difference of the COMOEA is that our objective is to combine entire MOEAs, while the works mentioned above only combine different environment selection. The COMOEA splits a single population, of size N, into k smaller populations of size N/k, each one executing a different MOEA. Each MOEA applying its parent selection, recombination strategy and environment selection approach. Besides the proposed framework is a general approach that could be implemented to combine any MOEAs. For instance, the proposed framework could be used to combine the Two_Arch2, PMEA and HEA-DP cooperatively.

According to the taxonomy proposed by Gong *et. al.* [11], the COMOEA uses the population distributed model. More specifically it can be seen as an island model, as it divides the global population into several subpopulations. Besides, the COMOEA is heterogeneous, as the algorithmic settings vary for each subpopulation. Compared to the AsAMP-dDE [10], the COMOEA also explores the multi-population characteristics. However, it is proposed to solve many-objective problems. Similarly to AsAMP-dDE, the COMOEA shares the information between the subpopulations. One difference is that, on COMOEA, each subpopulation executes a different MOEAs, sharing the information learned from various approaches.

3 THE PROPOSED FRAMEWORK

This section describes the proposed distributed framework for cooperation of Many-objective Evolutionary Algorithms. The objective is to use the different characteristics of the algorithms to explore the search space. An algorithm may improve in some areas of the space where others may stagnate. Besides, the cooperation may increase diversity, creating solutions based on different approaches.

We use the distributed population island model. The advantage of using the island model is that it improves the search ability of EAs, avoiding premature convergence [11]. The framework instantiates several islands, each island with a small population size. Each island executes a different MOEA on the same problem instance independently. Each MOEA applies its rules for the parent selection, recombination, and environmental selection. After a certain time, the MOEA shares some information with its neighbors. When an MOEA receives some information from a neighbor, it somehow introduces this information into its search.

The general framework proposed can be implemented in several ways, as each design choice leads to a different implementation. We identified four design choices about how to perform the distribution:

(1) Which information to exchange?
(2) When to perform the information exchange?
(3) How exchange the information?
(4) What to do with this information?

Also, we identified two design choices related to the MOEAs selected for cooperation:

(1) Which MOEAs select to cooperate?

(2) How to decompose the problem?

In this work we implemented the proposed framework as follows: The communication occurs when some individuals migrate from one island to the other. In COMOEA, each island (population) send all generated solutions to the others. Then each one filters which received solutions to accept and which solutions from its population to discard to keep the population size. The COMOEA uses a complete graph topology, as it shares the information between all populations. The framework was implemented synchronously, as the migration only occurs after all populations have finished generating the offspring.

Figure 1 depicts the implementation of the proposed framework used in this work. First, the framework initializes the problem instance, the maximum number of fitness evaluations and the pool of algorithms that are going to cooperate. The general structure required, for this approach, of an evolutionary algorithm is:

(1) *Initialize MOEA*: A method for initialization of the algorithm. Such as, create the initial population, initialize metrics and set the initial values for variables.
(2) *Generate Offspring*: Uses the algorithm population to select and reproduce generating an offspring set. In this step, the generated offspring set is evaluated on the problem instance and returned to the framework.
(3) *Update Population*: Receives a solution set and update the algorithm population based on the algorithm replacement rules.

After the initialization of the framework, the MOEAs from the pool are initialized, using the methodology of each MOEA. The stop condition is evaluated based on a maximum number of fitness evaluations (or the max number of generations times the population size). While is not met the stop criterion, the framework operates as follow: First, the MOEAs generate their offspring, each one using its population and methodology. Then, the produced solutions of all algorithms are joint on a single set. Finally, the MOEA updates its population using the offspring set of all MOEAs, based on the MOEA rules of replacement. The cooperation is when the offspring generated by all algorithms are joint. On the update population step, the MOEA may select solutions from any algorithm to compose its current population.

When the framework meets the stop criterion, it is joined all populations from all MOEAs, and the output is the non-dominated set of solutions. When cooperating between algorithms that use a set of reference vectors to guide the search the set of vectors may be split between the MOEAs to guide them towards different regions of the search space. It is proposed to intercalate the vectors among the MOEAs, combining the algorithms.

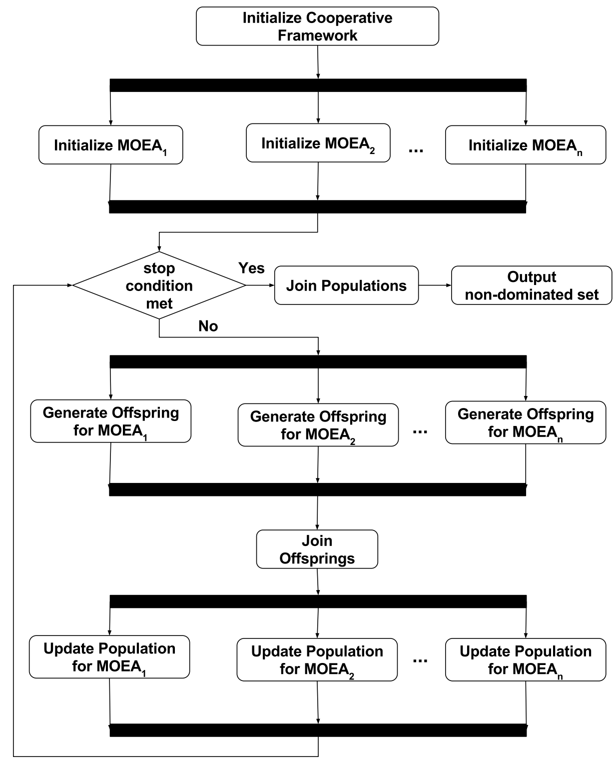

Figure 1: Implementation of the proposed framework

4 MAOEAS SELECTED FOR COOPERATION

To evaluate the proposed framework two state-of-the-art MOEAs were chosen, NSGA-III [6] and MOEA/D-STM [16]. The algorithms were adapted to be applied cooperatively, with methods to initialization, generating the offspring and update the population. The Sections 4.1 and 4.2 presents a description about the selected MOEAs. Both algorithms of the pool use reference vectors to guide the search. In this work, the objective space was decomposed using a reference set of points, split between the algorithms, alternating between one and the other. The COMOEA could also use algorithms that do not use reference vectors, cooperating with them or even with algorithms that use reference vectors. The proposed framework uses, as total population size, the population size of the algorithms executed alone, to perform a fair comparison. So, the population size of each subpopulation is the total population size divided by the number of MOEAs cooperating.

4.1 NSGA-III

The main characteristic of the NSGA-III [6] is the diversity maintenance by a supplied and adaptively updated set of well-spread reference points. The remaining of the algorithm is similar to the original NSGA-II [7]. First, it is applied recombination and mutation over the solutions; then the population (P of size N) and offspring (Q of size N) are merged, constituting a solution set R of size $2N$. After, the individuals are sorted using the non-dominated sort approach. Then, each non-dominated level is added to a set S until the size of S exceeds N. If there is no room for all solutions of the last non-dominated level (F_l) to be included in S, a diversity maintenance approach is applied.

At the diversity maintenance the NSGA-II and NSGA-III differs, one applies the Crowding Distance for diversity, while the other uses a reference point set. The first phase of the NSGA-III diversity approach is the adaptive normalization of the population. The normalization translates the solutions by subtracting the minimum value of each objective. Then the extreme point of each axis is identified. The extreme point is used to constitute a multi-dimensional hyperplane. Then it is computed the intersection of the hyperplane and the axis (a_i intercept a of i-th objective). Finally, it calculates the normalization as presented by Equation 1:

$$f_i^n(\mathbf{x}) = \frac{f_i(\mathbf{x}) - z_i^{min}}{a_i - z_i^{min}} \quad (1)$$

The normalization is performed every iteration and adaptively maintains the diversity. This normalization enables the application of NSGA-III to problems with different objective scales.

After the normalization, the algorithm associates each solution in S with the closest reference line (using perpendicular distance). Then, it counts the number of solutions from P_{t+1} (where $P_{t+1} = S_t \backslash F_l$) associated with each reference point and identifies the reference line (ω_j) with the minimum associated solutions. If the reference line has no solutions from P_{t+1} associated, and there are one or more members of F_l associated with the reference line, then the solution from F_l with the shortest perpendicular distance to the reference line is added to P_{t+1}. If the reference line has one or more solutions from P_{t+1} associated, then a solution from F_l, associated with ω_j, is randomly chosen. In any case that, there is no solution from F_l associated with ω_j the reference line does not receive a new solution. This process is repeated iteratively until the population size is full ($|P_{t+1}| = N$).

Initially, the reference points are generated using the Das and Dennis [5] approach, but any set of points could be used to represent the preferences direction of the user. Equation 2 gives the number of reference points, where m is the number of objectives and p is the number of divisions per objective. With a uniform spacing of $\delta = 1/p$, as illustrated in the Figure 2. The value of p should be $p \geq m$ to have intermediate points. However, with the increasing of the number of objectives, the number of reference points increases largely, affecting the computational cost. We used a two-layer method when $m \geq 8$ to overcome this difficulty, as presented by [6] and [15] and illustrated in the Figure 3. First, it creates a boundary layer; then another layer is produced and shrunk by a transformation to cover the inside regions of the objective space [15].

$$H = \binom{m+p-1}{p} \quad (2)$$

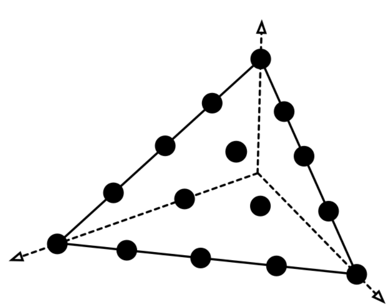

Figure 2: Reference points generation example, $p = 4$, $m = 3$ **and** $H = 15$**. Figure adapted from [15].**

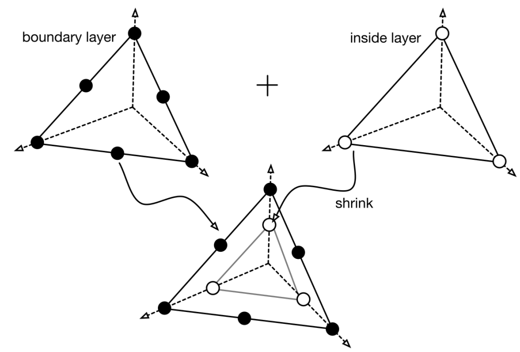

Figure 3: Two-layer reference points generation example, $p_{boundary} = 2$, $p_{inside} = 1$, $m = 3$ **and** $H = 9$**. Figure adapted from [15].**

4.2 MOEA/D-STM

The MOEA/D-STM [16] proposes a new environment selection for MOEA/D. The MOEA/D-STM uses the Stable Matching (STM) based selection model derived from the Stable Marriage Problem (SMP). The SMP matches two sets of agents A and B. It is considered an unstable marriage when two agents, one from A and one B, are married but both prefers to be married to other agents from B and A respectively. The objective is to match the agents from A and B without generating unstable marriages. On MOEA/D-STM the aim is to match the weight vectors with the solutions. The algorithm maintains a population P of size N associated with a set of weight vectors W, also of size N. It also maintains the ideal and nadir objective vectors $\boldsymbol{z}^*$ and $\boldsymbol{z}^{nad}$. The algorithm updates the nadir and ideal points with the minimal and maximal points for each objective. Moreover, each weight vector has an associated neighborhood. The neighborhood of one weight vector $\boldsymbol{w}^i$ are the T closest weight vectors to $\boldsymbol{w}^i$.

The MOEA/D-STM inherits from MOEA/D-DRA the Dynamic Resource Allocation. Every iteration a 10-tournament selection is applied to compose a set of weight vectors to be updated. A utility function evaluates the quality of the weight vector. The value of the utility function is updated every a certain number of iterations. The reproduction uses Differential Evolution and polynomial mutation. The selection is performed randomly from the neighborhood (with probability δ) or the whole population (with a probability of $1 - \delta$). Then the algorithm evaluates the generated solution by the fitness function. The ideal and nadir points are updated. The produced offspring is merged with the parent population and submitted to the STM environment selection. The Stable Matching (STM) selection works as follow: First, all subproblems and solutions are set to be free. Then, iteratively until there is no subproblems free do: select subproblem, that does not have an associated solution, randomly. Then finds the solution that minimizes the aggregation subproblem function, to which the subproblem has not proposed yet, and offer to pair. If the solution is free, it accepts to pair. If it is not free, the solution compares its current pair with the subproblem and selects the closest. Then it pairs with the closest one, and the other is free.

5 EXPERIMENTAL SETUP

This section presents the experimental setup used in this paper. The objective is: empirically evaluate if, the proposed approach for cooperation, improves the performance of the MOEAs regarding convergence and diversity. On the experimental analysis, we will compare three algorithms. 1) The cooperation between NSGA-III and MOEA/D-STM, using the proposed approach (COMOEA); 2) The NSGA-III and; 3) the MOEA/D-STM. For that comparison, six benchmark problems will be used, with different values of objectives. The methodology employed for the experiments is based on [6]. The number of iterations used for different test instances variates according to the problem and number of objectives and it is presented by Table 1. We selected the benchmark problems used in this experiments from DTLZ [8] and WFG [12] families of problems, as illustrated by Table 2. We set the number of decision variables according to the values proposed by the benchmark authors and varying according to the number of objectives [8, 12]. The experiments were performed using 3 to 15 objectives (m). The selected benchmark problems represent different characteristics of MOPs (Multi-Objective Problems), such as linear and concave shapes, multi-modality and separability. For instance, the DTLZ1 and DTLZ3 are hard-to-converge multi-modal problems, WFG6 is a non-separable reduced problem, and WFG7 is separable, unimodal [17].

Table 1: Number of iterations for different test instances.

	Number of objectives (m)				
Problem	3	5	8	10	15
DTLZ1	400	600	750	1000	1500
DTLZ2	250	350	500	750	1000
DTLZ3	1000	1000	1000	1500	2000
DTLZ4	600	1000	1250	2000	3000
WFG6	400	750	1500	2000	3000
WFG7	400	750	1500	2000	3000

Table 2: Characteristics and number of decision variables of the problems used.

Problem	Characteristics	Decision Variables Number
DTLZ1	linear, multi-modal	$m + k - 1, k = 5$
DTLZ2	concave	$m + k - 1, k = 10$
DTLZ3	concave, multi-modal	$m + k - 1, k = 10$
DTLZ4	concave, biased	$m + k - 1, k = 10$
WFG6	concave, non-separable	$k + l, k = 2 \times (m - 1), l = 20$
WFG7	concave, biased	$k + l, k = 2 \times (m - 1), l = 20$

The proposed framework was evaluated using two state-of-art Multi-Objective Evolutionary Algorithms: MOEA/D-STM [16] and NSGA-III [6] (described in Section 4). They share the characteristic of using a reference set of points to guide the search. However, they use different quality measures for parent and replacement selections. The reference set of points was split between the algorithms, alternating between one and the other. We configure the population size of each cooperative party according to the number of reference points associated with it. Table 3 presents the population size (and the number of reference points) associated with each algorithm. The reference points and the population size are based on [6].

Table 3: Population size (and corresponding reference points) used by the NSGA-III and MOEA/D-STM for cooperation, and NSGA-III and MOEA/D-STM alone.

m	COMOEA			NSGA-III	MOEA/D
	NSGA-III	MOEA/D	total		
3	48 (46)	45 (45)	93 (91)	92 (91)	91 (91)
5	108 (105)	105 (105)	213 (210)	210 (210)	210 (210)
8	80 (78)	78 (78)	158 (156)	156 (156)	156 (156)
10	140 (138)	137 (137)	277 (275)	276 (275)	275 (275)
15	68 (68)	67 (67)	135 (135)	136 (135)	135 (135)

In this table, MOEA/D means MOEA/D-STM.
m is the number of objectives.

To create the reference points set it we used the Das and Dennis's approach [5, 6]. The NSGA-III parameter configuration employed in this paper is based on [6]. The crossover operator is the SBX, with probability $p_c = 1.0$, and polynomial mutation with probability $p_m = 1/n$, where n is the number of decision variables. The crossover distribution index is set to $\eta_c = 30$, and the mutation distribution index is $\eta_m = 20$. For the MOEA/D-STM, the parameter configuration is the same as the MOEA/D-STM paper [16]. The reproduction is performed using differential evolution (DE) and polynomial mutation. The mutation probability is $p_m = 1/n$ and the distribution index is $\eta_m = 20$. The DE parameters CR and F are set to $CR = 1.0$ and $F = 0.5$. The neighborhood size T was set to 20 with a probability to select in the neighborhood of $\delta = 0.9$.

To evaluate the performance of the framework, versus the MOEAs, executed alone, we used two quality indicators, IGD and Hypervolume [15]:

- The Inverted Generational Distance (IGD) comprises both convergence and diversity in a single scalar, with a low computational cost. Given a Pareto approximation set A obtained by one execution of an MOEA, the IGD was computed as presented in Equation 3. Where Z is a discrete set of points representing the Pareto front, and $dist(a, b)$ is the Euclidean distance. It is possible to understand the IGD as the average distance, from every point of the true Pareto front to the closest point in the approximation set. As near to the Pareto front is the approximation set, smaller is the IGD value [6], [16].

$$IGD(A, Z) = \frac{\sum_{i=1}^{|Z|} \min_{j=1}^{|A|} dist(z_i, a_j)}{|Z|} \quad (3)$$

- The Hypervolume (HV) measures the multidimensional volume of the objective space dominated by a solution set, bounded by a reference point (or nadir point) z [15]. The Equation 4 illustrates the HV computation. Where VOL($\cdot$) is the Lebesgue measure; In this paper, we set the nadir point according to each problem instance, and any solution worse than the nadir point, in any objective, was not considered for the hypervolume computation [15]. The nadir point for DTLZ1 was set to $(1.0, \ldots, 1.0)$, for DTLZ2 to DTLZ4 to $(2.0, \ldots, 2.0)$, and for WFG6 and WFG7 to $(3.0, \ldots, 2.0 \times m + 1.0)$ [15].

$$HV(S) = \text{VOL}\left(\bigcup_{\mathbf{x}\in S}[f_1(\mathbf{x}), z_1] \times \ldots [f_m(\mathbf{x}), z_m]\right) \quad (4)$$

The hypervolume is the only metric strictly monotonic; *i.e.*, if a Pareto set A dominates another Pareto set B then the hypervolume value of A will be larger (better) than B. The main disadvantage of hypervolume is its computational cost and the bias towards the knee regions of the Pareto front [2].

6 EXPERIMENTAL RESULTS

This section presents the comparison among the configured COMOEA (using NSGA-III and MOEA/D-STM), versus NSGA-III and MOEA/D-STM alone. The Tables 4 and 5 presents the mean of the quality indicator (and standard deviation) results for Hypervolume and IGD. For each problem, the boldface represents the best result. The gray background represents if the result is different from the others. The difference is evaluated statistically with pairwise comparisons, after Kruskal-Wallis rank sum test with 95% significance. The pairwise comparison is performed using Tukey and Kramer (Nemenyi) test with Tukey-Dist approximation for independent samples.

The COMOEA achieved the best results in most problems for Hypervolume (Table 4), 26 best averages, 22 of them with statistical difference (for 30 evaluated problems). It also achieved the best result in most of the problem instances, 26 of 30, for IGD (Table 5), with a statistical difference in 21 problems. The MOEA/D-STM (without cooperation) achieved the best average in 4 problem instances for hypervolume and 3 instances for IGD, with statistical significance on the DTLZ1 with 15 objectives for the Hypervolume The NSGA-III achieved only 1 best averages for IGD, without a statistical difference to the other evaluated algorithms.

Due to the differences between the quality indicators they do not agree in all problem cases. Although, it is possible to observe that, in general, the COMOEA obtained the best results in most problem instances, with significance in many cases, for both quality indicators. The pairwise comparison supports this conclusion. The pairwise comparison compares all algorithms against each other with the Nemenyi test with 99% significance. The data used is the indicator mean for each problem. Critical difference plots, proposed by [9], present the results for each quality indicator. The plot connects the group name with its average ranking, and a bold line connects the groups that are statistically equivalent.

In a general comparison, the cooperation between the MOEAs obtained the best average ranking, with statistical significance, in all evaluated quality indicators, Hypervolume (Figure 4) and IGD (Figure 5). Followed by MOEA/D-STM, and then the NSGA-III. Although, there was statistically equivalence between the NSGA-III and MOEA/D-STM for IGD.

This section presented the experimental results. In most problems, the cooperation between the algorithms was capable of achieving best results or equivalent than the evaluated MOEAs applied alone. It was possible to observe a statistical significance in the

Table 4: The mean (and standard deviation) of Hypervolume. The best results are in boldface and a statistical difference in a gray background.

Obj.	problem	COMOEA	MOEADSTM	NSGAIII
3	DTLZ1	**9.7E-1(4.92E-4)**	9.67E-1(5.64E-4)	9.16E-1(3.43E-2)
	DTLZ2	**7.38E0(3.91E-3)**	7.37E0(3.06E-3)	6.68E0(2.75E-1)
	DTLZ3	**7.38E0(4.23E-3)**	7.35E0(1.2E-2)	6.2E0(7.17E-1)
	DTLZ4	7.33E0(2.18E-1)	**7.36E0(3.25E-2)**	6.82E0(8.05E-1)
	WFG6	**6.75E1(4.41E-1)**	6.71E1(1.28E-1)	5.35E1(4.05E0)
	WFG7	**7.21E1(5.59E-1)**	7.12E1(6.03E-1)	4.39E1(4.83E0)
5	DTLZ1	**9.98E-1(4.27E-5)**	9.54E-1(1.01E-1)	9.57E-1(5.26E-2)
	DTLZ2	**3.16E1(3.7E-3)**	3.15E1(8.48E-3)	2.96E1(1.05E0)
	DTLZ3	**3.16E1(5.49E-3)**	1.97E1(1.25E1)	2.49E1(1.12E1)
	DTLZ4	**3.16E1(9.68E-3)**	3.15E1(2.41E-2)	3.07E1(5.63E-1)
	WFG6	**7.81E3(9.17E1)**	7.18E3(1.57E2)	4.17E3(4.1E2)
	WFG7	**8.6E3(1.32E2)**	7.66E3(2.04E2)	4.36E3(6.74E2)
8	DTLZ1	9.95E-1(1.09E-3)	**9.95E-1(1.11E-3)**	6.26E-1(3.64E-1)
	DTLZ2	**2.41E2(2.58E0)**	2.37E2(2.47E0)	2.02E2(2.04E1)
	DTLZ3	**2.4E2(2.7E0)**	2.35E2(3.09E0)	0E0(0E0)
	DTLZ4	**2.53E2(1.16E0)**	2.42E2(1.42E0)	2.2E2(1.57E1)
	WFG6	**2.24E7(9.08E5)**	2.21E7(1.07E6)	5.65E6(1.17E6)
	WFG7	**2.32E7(8.84E5)**	2.11E7(9.02E5)	6.05E6(8.79E5)
10	DTLZ1	9.97E-1(6.07E-4)	**9.97E-1(2.43E-4)**	6.53E-1(3.1E-1)
	DTLZ2	**9.62E2(8.73E0)**	9.39E2(8.04E0)	8.75E2(5.25E1)
	DTLZ3	**9.57E2(7.48E0)**	9.27E2(9.49E0)	0E0(0E0)
	DTLZ4	**1.02E3(3.36E0)**	9.49E2(1.06E1)	8.22E2(4.16E1)
	WFG6	**9.43E9(6.88E8)**	8.78E9(7.41E8)	2.1E9(3.57E8)
	WFG7	**9.91E9(3.78E8)**	8.79E9(4.42E8)	2.72E9(4.77E8)
15	DTLZ1	9.78E-1(6.1E-3)	**9.85E-1(1.55E-3)**	6.15E-1(2.93E-1)
	DTLZ2	**2.54E4(8.39E2)**	2.47E4(5.66E2)	2.35E4(2.05E3)
	DTLZ3	**2.56E4(1.22E3)**	2.42E4(7.13E2)	0E0(0E0)
	DTLZ4	**3.13E4(4.11E2)**	2.57E4(7.74E2)	2.68E4(2.59E3)
	WFG6	**9.56E16(9.53E15)**	9.03E16(1.03E16)	1.96E16(5.5E15)
	WFG7	**8.11E16(6.07E15)**	6.52E16(8.32E15)	2.27E16(6.1E15)

Table 5: The mean (and standard deviation) of IGD. The best results are in boldface and a statistical difference in a gray background.

Obj.	problem	COMOEA	MOEADSTM	NSGAIII
3	DTLZ1	**7.17E-3(3.18E-4)**	8.15E-3(9.89E-5)	2.25E-2(1.02E-2)
	DTLZ2	8.98E-3(2.7E-4)	**8.97E-3(7.91E-5)**	1.95E-2(5.45E-3)
	DTLZ3	**9E-3(4.71E-4)**	9.17E-3(1.31E-4)	3.5E-2(1.64E-2)
	DTLZ4	**1.23E-2(1.39E-2)**	1.49E-2(1.05E-2)	3.67E-2(3.38E-2)
	WFG6	**1.09E-2(5.81E-4)**	1.2E-2(4.15E-4)	2.37E-2(5.49E-3)
	WFG7	**9.6E-3(6.65E-4)**	1.03E-2(2.92E-4)	3.83E-2(7.96E-3)
5	DTLZ1	**1.07E-2(2.59E-4)**	2.19E-2(6.49E-3)	2.12E-2(1.04E-2)
	DTLZ2	**1.57E-2(2.92E-4)**	2.21E-2(1.64E-4)	2.11E-2(3.18E-3)
	DTLZ3	**1.59E-2(4.45E-4)**	1.51E-1(2.59E-1)	5.92E-2(7.08E-2)
	DTLZ4	**1.88E-2(4.04E-4)**	2.38E-2(1.44E-3)	3.48E-2(9.51E-3)
	WFG6	**1.68E-2(9.97E-4)**	2.38E-2(1.18E-3)	3.07E-2(3.22E-3)
	WFG7	**2.1E-2(2.6E-3)**	3.31E-2(2E-3)	3.54E-2(7.03E-3)
8	DTLZ1	**2.88E-2(9.39E-4)**	3.26E-2(5.95E-4)	8.73E-2(5.16E-2)
	DTLZ2	**4.57E-2(1.56E-3)**	5.39E-2(9.97E-4)	5.76E-2(7.54E-3)
	DTLZ3	**4.66E-2(1.72E-3)**	5.6E-2(3.71E-3)	5.37E0(2.17E0)
	DTLZ4	**4.49E-2(5.16E-3)**	5.96E-2(4.43E-3)	8.34E-2(6.35E-3)
	WFG6	**4.1E-2(4.43E-3)**	4.44E-2(3.83E-3)	6.35E-2(5.43E-3)
	WFG7	6.13E-2(5.98E-3)	**5.87E-2(6.62E-3)**	7.8E-2(9.51E-3)
10	DTLZ1	**2.36E-2(5.59E-4)**	2.72E-2(3.02E-4)	6.84E-2(3.27E-2)
	DTLZ2	**3.95E-2(7.38E-4)**	4.58E-2(5.36E-4)	4.55E-2(4.64E-3)
	DTLZ3	**4.03E-2(1.3E-3)**	4.71E-2(1.13E-3)	4.66E0(2.26E0)
	DTLZ4	**3.71E-2(2.66E-3)**	4.87E-2(2.36E-3)	7.04E-2(2.22E-3)
	WFG6	**3.87E-2(4E-3)**	4.63E-2(2.79E-3)	4.97E-2(3.8E-3)
	WFG7	**6.57E-2(7.46E-3)**	6.79E-2(1.3E-2)	7.62E-2(9.92E-3)
15	DTLZ1	**5.17E-2(1.78E-3)**	5.41E-2(5.41E-4)	1.04E-1(5.24E-2)
	DTLZ2	**8.34E-2(3.79E-3)**	8.72E-2(1.92E-3)	9.82E-2(3E-3)
	DTLZ3	**8.32E-2(4.1E-3)**	8.92E-2(2.46E-3)	4.56E0(3.18E0)
	DTLZ4	**7.38E-2(2.39E-3)**	8.92E-2(3.31E-3)	1.07E-1(4.35E-3)
	WFG6	1.46E-1(7.2E-2)	1.27E-1(1.6E-2)	**1.19E-1(8.84E-3)**
	WFG7	1.24E-1(8.52E-3)	**1.2E-1(1.15E-2)**	2.09E-1(3.35E-2)

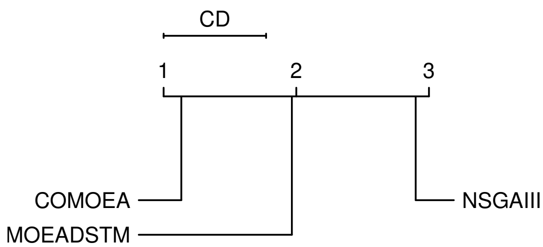

Figure 4: Critical difference plot for Hypervolume indicator. The algorithms that are not joined by a bold line can be regarded as different.

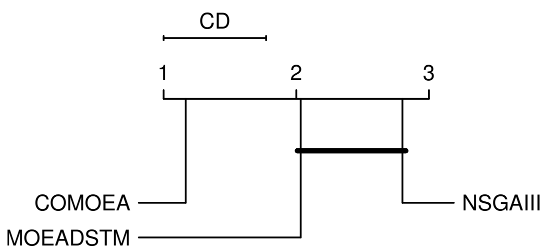

Figure 5: Critical difference plot for IGD indicator. The algorithms that are not joined by a bold line can be regarded as different.

quality improvement. The empirical results obtained in this experiments indicates that the cooperation between Multi-Objective Evolutionary Algorithms is capable of improving the quality of them for Multi-Objective Optimization.

7 CONCLUSION

In this work, we presented a distributed framework for cooperation of Many-objective Evolutionary Algorithms. The objective was to propose a way to combine the algorithms strengths, executing different sub-populations and migrating solutions from them. To do so, each MOEA generates an offspring, based on its rules and using its subpopulation. Then the offspring from all MOEAs are joined in the same solution set. This solution set is then sent to the MOEAs to update their population using each one its environmental selection rules. To evaluate the framework a pool of two state-of-the-art MOEAs were composed: MOEA/D-STM and NSGA-III. The cooperation between this two algorithms was compared with them executed alone and the results evaluated by IGD and hypervolume quality indicators. The collaboration between the algorithms was capable of improving the convergence and diversity of the algorithms in most problem instances for both evaluated quality indicators. The results obtained in this paper indicates that the cooperation between MOEAs may improve the convergence and diversity of the algorithms. Those results motivate the investigation of future works, such as: increasing the number of algorithms (MOEAs to cooperate); the study of offline selection approaches to compose the pool or online selection to chose after a certain time which MOEA to execute for each subpopulation.

REFERENCES

[1] Enrique Alba, Gabriel Luque, and Sergio Nesmachnow. 2013. Parallel metaheuristics: recent advances and new trends. *International Transactions in Operational Research* 20, 1 (2013), 1–48. DOI: http://dx.doi.org/10.1111/j.1475-3995.2012.00862.x

[2] Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. 2012. On the Properties of the R2 Indicator. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO '12)*. ACM, New York, NY, USA, 465–472. DOI: http://dx.doi.org/10.1145/2330163.2330230

[3] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. 2006. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[4] Teodor Gabriel Crainic and Michel Toulouse. 2003. *Parallel Strategies for Meta-Heuristics*. Springer US, Boston, MA, 475–513. DOI: http://dx.doi.org/10.1007/0-306-48056-5_17

[5] Indraneel Das and J. E. Dennis. 1998. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization* 8, 3 (March 1998), 631–657.

[6] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (Aug 2014), 577–601. DOI: http://dx.doi.org/10.1109/TEVC.2013.2281535

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (Apr 2002), 182–197. DOI: http://dx.doi.org/10.1109/4235.996017

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2002. Scalable multi-objective optimization test problems. In *IEEE Congress on Evolutionary Computation*, Vol. 1. 825–830.

[9] Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* 7 (Dec. 2006), 1–30. http://dl.acm.org/citation.cfm?id=1248547.1248548

[10] I. De Falco, U. Scafuri, E. Tarantino, and A. Della Cioppa. 2016. An asynchronous adaptive multi-population model for distributed differential evolution. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. 5010–5017. DOI: http://dx.doi.org/10.1109/CEC.2016.7748324

[11] Yue-Jiao Gong, Wei-Neng Chen, Zhi-Hui Zhan, Jun Zhang, Yun Li, Qingfu Zhang, and Jing-Jing Li. 2015. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing* 34 (2015), 286 – 300. DOI: http://dx.doi.org/10.1016/j.asoc.2015.04.061

[12] S. Huband, P. Hingston, L. Barone, and L. While. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10, 5 (Oct. 2006), 477–506. DOI: http://dx.doi.org/10.1109/TEVC.2005.861417

[13] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. 2015. Many-Objective Evolutionary Algorithms: A Survey. *ACM Comput. Surv.* 48, 1, Article 13 (Sept. 2015), 35 pages. DOI: http://dx.doi.org/10.1145/2792984

[14] B. Li, K. Tang, J. Li, and X. Yao. 2016. Stochastic Ranking Algorithm for Many-Objective Optimization Based on Multiple Indicators. *IEEE Transactions on Evolutionary Computation* 20, 6 (Dec 2016), 924–938. DOI: http://dx.doi.org/10.1109/TEVC.2016.2549267

[15] K. Li, K. Deb, Q. Zhang, and S. Kwong. 2015. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Transactions on Evolutionary Computation* 19, 5 (Oct 2015), 694–716. DOI: http://dx.doi.org/10.1109/TEVC.2014.2373386

[16] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang. 2014. Stable Matching-Based Selection in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation* 18, 6 (Dec 2014), 909–923. DOI: http://dx.doi.org/10.1109/TEVC.2013.2293776

[17] H. Wang, L. Jiao, and X. Yao. 2015. Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 19, 4 (Aug 2015), 524–541. DOI: http://dx.doi.org/10.1109/TEVC.2014.2350987

[18] Z. J. Wang, Z. H. Zhan, and J. Zhang. 2016. Parallel multi-strategy evolutionary algorithm using massage passing interface for many-objective optimization. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. 1–8. DOI: http://dx.doi.org/10.1109/SSCI.2016.7850228

[19] D. H. Wolpert and W. G. Macready. 1997. No Free Lunch Theorems for Optimization. *Trans. Evol. Comp* 1, 1 (April 1997), 67–82. DOI: http://dx.doi.org/10.1109/4235.585893

[20] Q. Zhang, W. Liu, and H. Li. 2009. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *2009 IEEE Congress on Evolutionary Computation*. 203–208. DOI: http://dx.doi.org/10.1109/CEC.2009.4982949

[21] Y. H. Zhang, Y. J. Gong, J. Zhang, and Y. b. Ling. 2016. A hybrid evolutionary algorithm with dual populations for many-objective optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. 1610–1617. DOI: http://dx.doi.org/10.1109/CEC.2016.7743981