

Objective Space Partitioning Using Conflict Information for Solving Many-Objective Problems

Antonio López Jaimes^{a,1,*}, Carlos A. Coello Coello^a, Hernán Aguirre^b, Kiyoshi Tanaka^b

^a*CINVESTAV-IPN, Computer Science Department, Mexico City, MEXICO*

^b*Shinshu University, Faculty of Engineering, Nagano, JAPAN*

Abstract

We present an algorithm that partitions the objective space based on an analysis of the conflict information obtained from the current Pareto front approximation. By partitioning the objectives in terms of the conflict among them, we aim to separate the multiobjective optimization into several subproblems in such a way that each of them contains the information to preserve as much as possible the structure of the original problem. We implement this framework by performing ranking and parent selection independently in each subspace. Our experimental results show that the proposed conflict-based partition strategy outperforms a similar algorithm in a test problem with independent groups of objectives. In addition, the new strategy achieves a better convergence and distribution than that produced by a strategy that creates subspaces at random. In problems in which the degree of conflict among the objectives is significantly different, the conflict-based strategy presents a better performance.

Keywords: Multiobjective optimization, Many-objective optimization, Space Partitioning, Objective Conflict, Objective Correlation.

1. Introduction

Since the first implementation of a Multiobjective Evolutionary Algorithm (MOEA) in the mid 1980s [37], a wide variety of new MOEAs have been proposed, gradually improving in both their effectiveness and efficiency to solve Multiobjective Optimization Problems (MOPs) [9]. However, until recently, most of these algorithms had been evaluated and applied to problems with only two or three objectives, in spite of the fact that many real-world problems have more than three objectives (e.g., see [18, 25, 39]).

*Corresponding author, telephone: +81-50-3362-5271, fax: +81-42-759-8461

Email addresses: tonio.jaimes@gmail.com (Antonio López Jaimes), cocoello@cs.cinvestav.mx (Carlos A. Coello Coello), ahernan@shinshu-u.ac.jp (Hernán Aguirre), ktanaka@shinshu-u.ac.jp (Kiyoshi Tanaka)

Recent experimental [23, 42, 32, 34] and analytical [40, 27] studies have shown that MOEAs based on Pareto optimality [31] scale poorly when the number of objectives is increased. Although some scalability issues are known to mainly affect Pareto-based MOEAs (see e.g., [27, 16, 26]), optimization problems with a high number of objectives (also known as *many-objective problems*) introduce some difficulties common to any other multiobjective optimizer (e.g., visualization of the Pareto front). Three of the most serious difficulties are the following: *i*) Deterioration of the search ability because the proportion of nondominated solutions in a population increases rapidly with the number of objectives [16]. As a consequence, in many-objective problems, the selection of solutions is carried out almost at random or guided by diversity criteria only. *ii*) The number of points required to achieve a representative sample of a Pareto front increases exponentially with the number of objectives. *iii*) The visualization of the Pareto front is more complicated since with more than three objectives it is not possible to plot the Pareto front as usual. This is a serious problem since visualization plays a key role for a proper decision making.

Currently, there are three main approaches to solve many-objective problems, namely:

1. Adopt or propose an optimality relation that yields a solution ordering finer than that yielded by Pareto optimality. Among these alternative relations we can find k -optimality [16], preference order ranking [14], and a method that controls the dominance area [35]. Besides providing a richer ordering of the solutions, these relations obtain an optimal set which is a subset of the Pareto optimal set. Therefore, these techniques can be used as a partial remedy for the first and second issues of the previous enumeration.
2. Reduce the number of objectives of the problem during the search process [6, 30] or, a posteriori, in the decision making process [36, 4, 29]. The main goal of this kind of reduction techniques is to identify the redundant objectives (or redundant to some degree) in order to discard them. A redundant objective is one that can be removed without changing the dominance relation¹ induced by the original objective set.
3. Incorporation of preference information interactively throughout the course of the optimization process [11, 41, 17]. By incorporating preferences, the search can be focused on the decision maker's region of interest, avoiding this way, the evaluation of a huge number of solutions.

A general scheme for partitioning the objective space into several subspaces in order to deal with many-objective problems was introduced in [3, 2]. In that study we investigated the following three strategies to partition the objective space in equally sized subspaces: random (objectives for each subspace are assigned at random), fixed (objectives for each subspace are assigned sequentially),

¹The dominance relation induced by a given set F of objectives is defined by $\preceq_F = \{(\mathbf{x}, \mathbf{y}) | \forall f_i \in F : f_i(\mathbf{x}) \leq f_i(\mathbf{y})\}$.

and shift (at each generation, objectives in the partition are shifted one position to the right). Here, we propose a new partition strategy that creates objective subspaces based on the analysis of the conflict information obtained from the Pareto front approximation found by the underlying MOEA. Additionally, we introduce a new version of the general partitioning scheme in order to improve the distribution along the Pareto front. By grouping objectives in terms of the conflict among them, we aim to separate the MOP into several subproblems in such a way that the union of these independent subproblems contains the information to preserve as much as possible the structure of the original problem.

In order to evaluate the effectiveness of the new conflict-based partition strategy, we compare its performance against different strategies. First, against a similar partition strategy proposed by Purshouse and Fleming [33] which is based on the Kendall correlation. We also compared the new partitioning approach with the random strategy and the original Non-dominated Sorting Genetic Algorithm II (NSGA-II). The experimental results show that the conflict-based strategy outperform the Kendall-based partitioning method. Additionally, the conflict-based and random partition strategies outperform NSGA-II in all the test problems considered in this study. Regarding these two partition strategies, the conflict-based partition strategy achieves a better distribution of solutions than that achieved by the random strategy. In problems in which the degree of conflict among pairs of objectives is different, the conflict-based strategy presents a better performance.

The remainder of this paper is structured in the following manner. The next section presents some basic concepts and the notation adopted throughout the paper. Section 3 briefly describes the relevant research related to our work. Section 4 introduces the new partition strategy based on conflict information. Section 5 presents an experimental analysis to evaluate the effectiveness of the new partition strategy. Finally, we present our conclusions in Section 6.

2. Basic Concepts and Notation

Definition 1 (Multiobjective optimization problem). A *Multiobjective Optimization Problem (MOP)* is defined as:

$$\begin{aligned} & \text{Minimize } \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})]^T \\ & \text{subject to } \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{1}$$

The vector $\mathbf{x} \in \mathbb{R}^n$ is formed by n *decision variables* representing the quantities for which values are to be chosen in the optimization problem. The *feasible set* $\mathcal{X} \subseteq \mathbb{R}^n$ is implicitly determined by a set of equality and inequality constraints. The vector function $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^M$ is composed of $M \geq 2$ scalar *objective functions* $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ($i = 1, \dots, M$). In multiobjective optimization, the sets \mathbb{R}^n and \mathbb{R}^M are known as *decision variable space* and *objective function space*, respectively. The image of \mathcal{X} under the function \mathbf{f} is a subset of the objective function space denoted by $\mathcal{Z} = \mathbf{f}(\mathcal{X})$ and referred to as the *feasible set in the objective function space*.

Definition 2 (Objective set Φ). The objective set of a MOP is determined by the set $\Phi = \{f_1, f_2, \dots, f_M\}$ containing the M objective functions to be optimized.

Definition 3 (Subspace ψ). A subspace ψ of Φ is a lower-dimensional space that includes some of the objective functions in Φ , i.e. $\psi \subset \Phi$.

Definition 4 (Non-overlapping subspaces). Two subspaces $\psi_1 \subset \Phi$ and $\psi_2 \subset \Phi$ are said to be non-overlapping if they have no common objectives, i.e. $\psi_1 \cap \psi_2 = \emptyset$.

Definition 5 (Space partition Ψ). A space Φ is said to be partitioned into N_S subspaces, denoted as Ψ , if $\Psi = \{\psi_1, \dots, \psi_{N_S} \mid \cup_{i=1}^{N_S} \psi_i = \Phi \wedge \cap_{i=1}^{N_S} \psi_i = \emptyset\}$.

In multiobjective optimization, the *Pareto dominance relation* originally proposed by Edgeworth in 1881 [15], and generalized by Vilfredo Pareto in 1896 [31] is usually adopted.

Definition 6 (Pareto dominance relation). A solution \mathbf{x}^1 is said to Pareto dominate another solution \mathbf{x}^2 in the objective space formed by Φ , denoted by $\mathbf{x}^1 \prec \mathbf{x}^2$, if and only if: $\forall f_i \in \Phi : f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \wedge \exists f_i \in \Phi : f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$.

Thus, to solve a MOP we have to find those solutions $\mathbf{x} \in \mathcal{X}$ which are not Pareto dominated by any other solution with respect to objectives in Φ .

Definition 7 (Pareto optimality). A solution $\mathbf{x}^* \in \mathcal{X}$ is Pareto optimal if there does not exist another solution $\mathbf{x} \in \mathcal{X}$ such that $\mathbf{x} \prec \mathbf{x}^*$.

Definition 8 (Pareto optimal set). The Pareto optimal set, P_{opt} , is defined as: $P_{\text{opt}} = \{\mathbf{x} \in \mathcal{X} \mid \nexists \mathbf{y} \in \mathcal{X} : \mathbf{y} \prec \mathbf{x}\}$.

Definition 9 (Pareto front). For the Pareto optimal set P_{opt} , the Pareto front, PF_{opt} , is defined as: $PF_{\text{opt}} = \{\mathbf{z} = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})) \mid \mathbf{x} \in P_{\text{opt}}\}$.

In practice, the goal of the optimization process is finding the “best” approximation set of the Pareto optimal front. An approximation set is a finite subset of \mathcal{Z} composed of mutually nondominated vectors and denoted by PF_{approx} . Currently, it is well accepted that the best approximation set is determined by the closeness to the Pareto optimal front, and the spread over the entire Pareto optimal front [13, 45, 9].

Definition 10 (Pearson Correlation coefficient). This correlation coefficient measures the linear relationship between two observed data sets. Let X and Y be two data sets with ℓ elements each, and let $s_X > 0$ and $s_Y > 0$ denote their respective sample standard deviations. Then, the sample correlation coefficient, r_{XY} , of the data pairs (x_i, y_i) , $i = 1, \dots, \ell$ is defined by

$$r_{XY} = \frac{\sum_{i=1}^{\ell} (x_i - \bar{X})(y_i - \bar{Y})}{(\ell - 1)s_X s_Y}. \quad (2)$$

If $s_X = 0$ or $s_Y = 0$, then $r_{XY} = 0$. By definition, $-1 \leq r_{XY} \leq 1$.

When $r_{XY} > 0$ it is said that the sample data pairs are positively correlated, and when, $r_{XY} < 0$ it is said that they are negatively correlated. If $r_{XY} = 0$ the data pairs are not correlated. In other words, if $r_{XY} > 0$ ($r_{XY} < 0$, resp.) the data points tend to fall along a line of positive slope (negative slope, resp.).

Definition 11 (Correlation matrix). *The main diagonal elements of this $q \times q$ matrix are unity and the off diagonal elements r_{ij} are the correlation between data sets X_i and X_j for $i, j \in \{1, \dots, q\}$.*

2.1. Conflict Among Objectives

In the current literature it is possible to find several definitions of conflict among objectives (see e.g., [20, 1, 34, 4]). However, we used the definition proposed by Carlsson and Fullér [8] since it is intuitive and, as we explain in Section 4, it can be estimated using a low time complexity algorithm. Let $S_{\mathcal{X}}$ be a subset of \mathcal{X} , then, according to Carlsson and Fullér, two objectives can be related in the following ways (assuming minimization):

1. f_i is in conflict with f_j on $S_{\mathcal{X}}$ if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ implies $f_j(\mathbf{x}^1) \geq f_j(\mathbf{x}^2)$ for all $\mathbf{x}^1, \mathbf{x}^2 \in S_{\mathcal{X}}$.
2. f_i supports f_j on $S_{\mathcal{X}}$ if $f_i(\mathbf{x}^1) \geq f_i(\mathbf{x}^2)$ implies $f_j(\mathbf{x}^1) \geq f_j(\mathbf{x}^2)$ for all $\mathbf{x}^1, \mathbf{x}^2 \in S_{\mathcal{X}}$.
3. f_i and f_j are independent on $S_{\mathcal{X}}$, otherwise.

When $S_{\mathcal{X}} = \mathcal{X}$, it is said that f_i is in conflict with (or supports) f_j globally. However, in many MOPs the relation among the objectives changes for different subsets of \mathcal{X} . Figure 1 shows an example in which two functions are in conflict in some subsets of \mathcal{X} , while in others, they support each other.

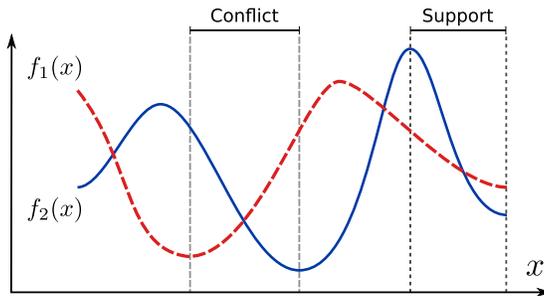


Figure 1: Two objective functions can be in conflict in some subsets of the feasible space, and can be supportive in other subsets.

In the relation of the case 2, those objectives are also called *non-conflicting*, *nonessential* or *redundant* objectives (see e.g., [19, 4, 36]) because, as pointed out by Gal and Hanne [19], when a nonessential objective is removed from the original set of objectives, the resulting Pareto front does not change. Based on this notion of nonessential objectives, Brockhoff and Zitzler [5] proposed a conflict definition that defines degrees of conflict according to the extent to which the Pareto front changes when some objectives are removed.

3. Related Work

In this section we describe two kinds of methods that incorporate concepts related to the objective partitioning algorithm proposed in this paper. The first kind of methods are closely related to our approach since they divide the original problem into subproblems. However, as detailed below, the main difference is the problem domain for which these methods were designed and therefore, the criteria adopted to divide the problem. Objective reduction methods need to be mentioned in this section as well since, as one of the elements of our method, they define a mechanism to measure conflict among objectives.

3.1. Objective Partitioning Methods

The Divide and Conquer MOEA proposed by Purshouse and Fleming [33] was designed for MOPs composed of subproblems (with their own group of objectives and variables) that can be solved independently from each other. That is, objectives of different subproblems are neither related by conflict or by a supporting relation. This algorithm divides the population into subpopulations according to an objective partition created using the Kendall rank correlation among the objectives. Specifically, a subpopulation is assigned to each objective subset and each subpopulation evolves totally independent of the others. A new objective partition is created at each generation according to the following procedure. First, a correlation test is applied on each pair of objectives to determine if they are independent or not. Using this information, a dependent subset is formed by objectives that are significantly correlated (either by conflict or by a supporting relation) with at least one of the other elements of the subset.

Unlike Purshouse and Fleming’s approach, the goal of the partitioning technique proposed in this paper is to deal with problems where all the objectives might be correlated, some of them by conflict and others by a supporting relation. Besides, as we indicate in Section 4.2, our approach takes into account the degree of conflict in order to partition the objective set. This way, our method is useful both in separable problems and in cases where all the objectives are dependent.

Another partitioning approach is the multi-level multiobjective genetic algorithm proposed by Gunawan et al. [21]. In this method the problem is divided into subproblems which in turn can be further decomposed in more levels. In this scheme the objectives are classified in exclusive and functionally separable. An objective of the first class does not need to be combined with others to form a subproblem. In the second class, the objectives are a combination of others (e.g., a linear aggregation), and thus, they can be separated to form subproblems. Unlike our approach, in this method the division of the problem is performed by the user before the search. This way, the user needs to know the nature of the problem in order to identify which objectives are exclusive or functionally separable.

3.2. Conflict Estimation Methods

Our approach is also related to techniques that reduce the number of objectives based on redundancy, especially those applied throughout the course of the search. Nevertheless, unlike objective reduction techniques, our approach integrates all the objectives (including those with low conflict) in order to cover the entire Pareto front.

Deb and Saxena [36] proposed a method for reducing the number of objectives based on principal component analysis. The main assumption is that if two objectives are negatively correlated (taking the generated Pareto front as the data set), then these objectives are in conflict with each other. This method was implemented using an iterative scheme in which a principal component analysis is applied on the PF_{approx} achieved by NSGA-II [10] in order to gradually discard the less conflicting objectives.

Brockhoff and Zitzler [4] proposed two greedy algorithms to reduce the number of objectives. Both algorithms use the ϵ -dominance relation to measure the change of PF_{approx} using the reduced and the original objective set. Objectives whose removal does not change PF_{approx} are considered non-conflicting objectives. In [6] these algorithms were incorporated into a MOEA to reduce the number of objectives during the search. Since the goal in this work was to improve the efficiency of hypervolume-based MOEAs, the non-conflicting objectives were discarded or aggregated to form a unique objective.

Similar to the previous approach, López Jaimes et al. [29] proposed two schemes to reduce the number of objectives. These algorithms are based on a feature selection technique which uses correlation between nondominated vectors to estimate the conflict between each pair of objectives. The complexity of both algorithms is $O(NM^2)$, where N is the size of the nondominated set and M is the number of objectives. Later, in [30] these schemes were used to reduce the objectives during the search.

4. Description of the Conflict-Based Partitioning Framework

In this section we describe the main idea of the partitioning framework which was introduced by Aguirre and Tanaka [2, 3]. A modification to this scheme is also explained. Then, the new partition strategy based on conflict information is introduced.

4.1. General Idea of the Partitioning Framework

The basic idea of the partitioning framework is to divide the objective space into several subspaces so that a different portion of the population focuses the search on a different subspace. By partitioning the objective space into subspaces, we aim to emphasize the search within smaller regions of objective space. In other words, this framework divides the original optimization problem into several small subproblems. Recently, Schütze et al. [38] have provided evidence that partitioning the space can improve the performance of a MOEA. In a multiobjective problem a descent cone is defined as the set of all directions in

which dominating solutions can be found. Thus, a large descent cone increases the probability of improving a previous solution. Regarding the partitioning method, the descent cone generated in each of their subproblems is larger than in the original problem and therefore the speed of convergence might be improved.

In our approach, instead of dividing the population into independent subpopulations, a fraction of the pool of parents for the next generation is selected based on a different subspace. This way, the pool of parents will be composed of individuals having a good performance in each subspace. Then, the crossover and mutation operators are applied as usual.

In our approach, we partition the objective set $\Phi = \{f_1, f_2, \dots, f_M\}$ into N_S non-overlapping and pairwise disjoint subspaces $\Psi = \{\psi_1, \psi_2, \dots, \psi_{N_S}\}$ (see Definition 5). Thus, the nondominated sorting and truncation procedures of NSGA-II are modified in the following way. For each subspace, the mixed population, \mathcal{R} , composed of parents, \mathcal{P} , and offspring, \mathcal{Q} , is ranked using nondominated sorting, as Figure 2 shows. That is, only the objectives of the given subspace are considered to rank the population. Then, from each sorted population, the best $|\mathcal{P}|/N_S$ solutions are selected to form a new parent population of size $|\mathcal{P}|$. After this, the new population is generated by means of recombination and mutation using binary tournaments.

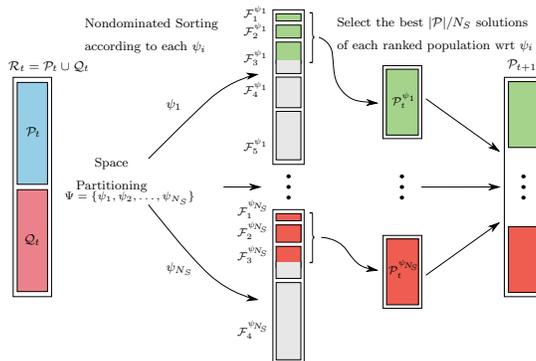


Figure 2: Nondominated sorting and truncation based on different subspaces of a partition $\Psi = \{\psi_1, \psi_2, \dots, \psi_{N_S}\}$.

The number of all possible ways to partition Φ into N_S subspaces is given by the Stirling number of the second kind (see e.g., [7]), which grows rapidly with the size of Φ . If the subspaces of the partition have the same size, the number of all possible partitions of Φ is given by $\frac{M!}{(k!)^{N_S} N_S!}$, where $M = |\Phi|$ and k is the size of each subspace. Therefore, it is not feasible searching in all possible subspaces. Instead, we can determine suitable subspaces using a partition strategy. In [2] we investigated three strategies to partition Φ : random, fixed, and shift partition strategy. In these strategies all the subspaces have the same number of objectives. In the *random strategy* a partition is uniformly chosen at random among all possible partitions of size N_S with equally sized subsets. The *fixed strategy* deterministically assigns objectives $f_i \in \Phi$ to sub-

spaces $\psi_s \in \Psi$ and keeps the same assignment throughout the generations. The *shift strategy*, at the first generation, assigns objectives in such a way that objectives assigned to a given ψ_s are ordered by their index i . Then, in subsequent generations, the objective with highest index in the s -th subspace is shifted to the $((s + 1) \bmod N_S)$ -th subspace, $\forall \psi_s \in \Psi$.

4.2. A New Partition Strategy

In this paper we investigate a new partition strategy using the conflict information among objectives. More specifically, the first partition would contain the least conflicting objectives, the second one the next least conflicting objectives, and so forth. In previous studies [29, 30], the conflict information among objectives has been used to remove objectives after, and during the search. However, instead of removing the least conflicting objectives, here we propose to integrate those objectives to form other subspaces. This way, compromise solutions from objectives with small conflict, but no zero, can be also found.

By grouping objectives in terms of the conflict among them, we are trying to separate the MOP into subproblems in such a way that the union of these independent subproblems contains the information to preserve most of the structure of the original problem. In other words, the goal is minimizing the difference between the original Pareto front and the one obtained by independently searching in different subspaces. However, in most real-world optimization problems, none or very few objectives are completely redundant. Therefore, in such cases, preserving completely the original structure of the problem is only possible when objectives in different subspaces are not in conflict at all.

In the following we present the details for measuring conflict among objectives and how to deal with two important difficulties previously outlined, namely: *i*) local conflict and *ii*) how to approximate the entire Pareto front when conflicting objectives are located in different subspaces.

In this paper we suggest using the correlation (see Def. 10) among the solutions in PF_{approx} to estimate the conflict among objectives in the sense defined by Carlsson and Fullér (Section 2.1). In this approach, each solution in PF_{approx} is an observation. A negative correlation between a pair of objectives means that one objective increases while the other decreases and vice versa. Thus, a negative correlation estimates the conflict between a pair of objectives. On the other hand, if the correlation is positive, then both objectives increase or decrease at the same time. That is, the objectives support each other. Furthermore, since the correlation coefficient values are in the range $[-1, 1]$, it is possible to define a measure of the degree of conflict between objectives. Therefore, in our approach we interpret that the more negative the correlation between two objectives, the more the conflict between them.

Although this approach only takes into account linear correlation among objectives, in [29] it was shown that this method produces similar results than those obtained by the methods proposed both in [4] and [36] (described in Section 3). In addition, using the correlation between objectives to estimate conflict has a low time complexity (as shown in Section 3) which makes this method suitable to be applied several times during the search of a MOEA.

Next, we will introduce a new version of the basic partitioning framework. In order to implement the new partition strategy we should consider that the conflict relation among the objectives might change during the search (local conflict). Besides, the conflict relation among the solutions in PF_{opt} might differ from that observed in the current PF_{approx} found during the search. Thus, to deal with this situation we suggest a new partitioning framework in which the search is divided into several cycles. In turn, each of these cycles is divided into two phases, namely, an *approximation phase* followed by a *partitioning phase*. In the approximation phase all the objectives are used as usual to select the new parent population. The goal of this phase is twofold: *i*) to update the current PF_{approx} in order to deal with local conflict and potential poor representations of PF_{opt} previously generated; and *ii*) to generate solutions representing the tradeoffs between conflicting objectives assigned to different subspaces. In turn, at the beginning of the partitioning phase, the current PF_{approx} is used to compute the correlation matrix for creating a new partition of the objective space. In each cycle, the approximation phase is carried out during G_Φ generations, whereas the partitioning phase is carried out during G_Ψ generations using the partition created at the beginning of the cycle. This idea is graphically explained in Figure 3.

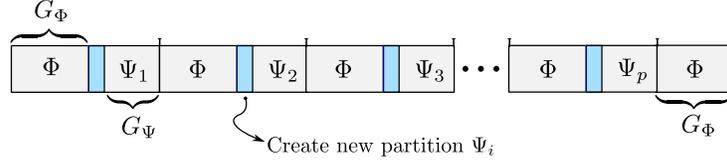


Figure 3: Alternation between entire objective space, Φ , and the partitioned space, Ψ_i .

The pseudocode of the entire proposed algorithm is given in Algorithm 1. The selection, crossover and mutation are carried out as usual to generate the offspring population, Q_t , from the current population, P_t (Alg. 1, line 9). The essential part of Algorithm 1 is the special way in which the nondominated sorting and truncation procedures are performed (Alg. 1, line 13). This procedure is described in Algorithm 2. The basic idea of this procedure was explained at the beginning of this section. In order to switch between the approximation and partitioning phases, we only need to change the kind of partition, Ψ , used in procedure SORT&TRUNCATION. For the approximation phase a special partition containing a single subspace is used (Alg. 1, line 4), while for the partitioning phase, a partition with N_S subspaces is employed. The method to create a partition is detailed in the next section.

4.3. Partitioning Using Conflict Information

The correlation matrix is computed using equation (2) on the current parent population. Since we are interested in measuring the negative correlation between objectives, the correlation matrix was modified so that each entry, r_{f_i, f_j} , contains the value $1 - r_{f_i, f_j}$. Thus each value of this new “conflict matrix” is

Algorithm 1 Pseudocode of our proposed partitioning MOEA.

Input:

Evolutionary operators values,
 G_{\max} , maximum number of generations,
 N_S , num. of subspaces,
 G_{Φ} , num. of generations using all the objectives as usual in each cycle,
 G_{Ψ} , num. of generations using the current partition, Ψ , in each cycle.

Output:

Pareto front approximation.

```

1:  $\mathcal{P}_1 \leftarrow \text{RANDOMPOPULATION}()$ 
2:  $\text{EVALUATE}(\mathcal{P}_1)$ 
3:  $\text{CROWDING}(\mathcal{P}_1)$  // Compute crowding distance for each solution.
4:  $\Psi \leftarrow \{f_1, \dots, f_M\}$  // All the objectives in a single subspace.
5:  $\text{phase} \leftarrow \text{APPROXIMATION}$  // Flag that indicates the current phase.
6:  $G_{\text{change}} \leftarrow G_{\Phi}$ 
7:  $g \leftarrow 0$ 
8: for  $t \leftarrow 1$  until  $G_{\max}$  do
9:    $Q_t \leftarrow \text{NEWPOP}(\mathcal{P}_t)$  // selection, crossover and mutation.
10:   $\text{EVALUATE}(Q_t)$ 
11:   $\mathcal{R}_t \leftarrow \mathcal{P}_t \cup Q_t$ 
12:  // Sort and truncate population using objective partition  $\Psi$ .
13:   $\mathcal{P}_{t+1} \leftarrow \text{SORT\&TRUNCATION}(\mathcal{R}_t, |\mathcal{P}_t|, \Psi)$ 
14:  if  $g = G_{\text{change}}$  then
15:     $g \leftarrow 0$ 
16:    if  $\text{phase} = \text{APPROXIMATION}$  then
17:      // Generate a new partition using the current population.
18:       $\Psi \leftarrow \text{CONFLICTPARTITION}(\mathcal{P}_{t+1}, \Phi, N_S)$ 
19:       $G_{\text{change}} \leftarrow G_{\Psi}$ 
20:       $\text{phase} \leftarrow \text{PARTITIONING}$ 
21:    else
22:       $\Psi \leftarrow \{f_1, \dots, f_M\}$ 
23:       $G_{\text{change}} \leftarrow G_{\Phi}$ 
24:       $\text{phase} \leftarrow \text{APPROXIMATION}$ 
25:    end if
26:  end if
27:   $g \leftarrow g + 1$ 
28: end for

```

in the range $[0, 2]$. A result of zero indicates that objectives f_i and f_j completely support each other, and a value of 2 indicates that they are completely in conflict.

Then, the subspaces are created from the least conflicting subspace to the most conflicting subspace. The procedure to create subspaces of size k is the following:

1. Create q -sized neighborhoods around each objective f_i , where $q = k - 1$. The conflict between objectives takes the role of the distance. That is, the more the conflict between two objectives, the more distant they are in the “conflict” space. Figure 4(a) shows two of these neighborhoods of a hypothetical situation to form subspaces of size $k = 3$.
2. Select the most compact neighborhood, i.e., the neighborhood with the smallest distance to its q -th neighbor (farthest neighbor). Figure 4(b)

Algorithm 2 Procedure of nondominated sorting and truncation.

Input:

\mathcal{R} : Initial population.
 P_{size} : Size of the final population.
 Ψ : Partition to be used.

Output:

A final population \mathcal{P}^* with size P_{size} .

```

procedure SORT&TRUNCATION( $\mathcal{R}, P_{\text{size}}, \Psi$ )
   $\mathcal{P}^* \leftarrow \emptyset$  // Initialize the truncated population.
  for  $i \leftarrow 1$  until  $|\Psi|$  do
    // Nondominated sort  $\mathcal{R}$  considering only the objectives in  $\psi_i$ .
     $\mathcal{F}^{\psi_i} \leftarrow \text{NONDOMINATEDSORT}(\mathcal{R}, \psi_i)$ 

    // Compute crowding dist. considering only the objectives in  $\psi_i$ .
    CROWDING( $\mathcal{F}^{\psi_i}, \psi_i$ )

    // Keep the best  $P_{\text{size}}/|\Psi|$  solutions for each subset  $\psi_i$ .
     $\mathcal{P}^{\psi_i} \leftarrow \text{TRUNCATION}(\mathcal{F}^{\psi_i}, P_{\text{size}}/|\Psi|)$  //  $|\mathcal{P}^{\psi_i}| = P_{\text{size}}/|\Psi|$ 
     $\mathcal{P}^* \leftarrow \mathcal{P}^* \cup \mathcal{P}^{\psi_i}$ 
  end for
  return  $\mathcal{P}^*$ 
end procedure

```

shows the farthest neighbor for each of the two neighborhoods. In the example, the neighborhood on the left is the most compact one.

3. Finally, the objectives in the most compact neighborhood, including objective f_i , form a new subspace, and these objectives are removed from the conflict matrix.

This process is repeated until all objectives are assigned to a subspace. Therefore, the first subspace created contains the least conflicting objectives, and the last subspace is formed by the most conflicting objectives. Algorithm 3 shows the pseudocode of this process. In that algorithm, the neighborhood of objective f_i is denoted by the ordered list L_{f_i} , in which the element $L_{f_i}[j]$ represents the conflict value between the j -th neighbor and objective f_i .

In the current implementation all subspaces have the same dimension M/N_S in case $r = (M \bmod N_S)$ is zero. Otherwise, r of the N_S subspaces have dimension $M/N_S + 1$ and the rest M/N_S .

This algorithm can be classified as a greedy approach since in order to add a new subspace to the partition, it always selects the most compact neighborhood from the remaining objective set. Nonetheless, some other approaches to create the objective subspaces are also possible. An example is creating a partition that maximizes the sum of the average conflict among objectives in each subspace. This way, we could avoid allocating the same proportion of individuals to subspaces with a very low contribution to the search. Unfortunately, to find this partition a large number of candidate partitions needs to be explored (see a bound for this number at the end of Section 4.1). Therefore, in this study, for efficiency reasons, we have chosen a greedy approach to create partitions.

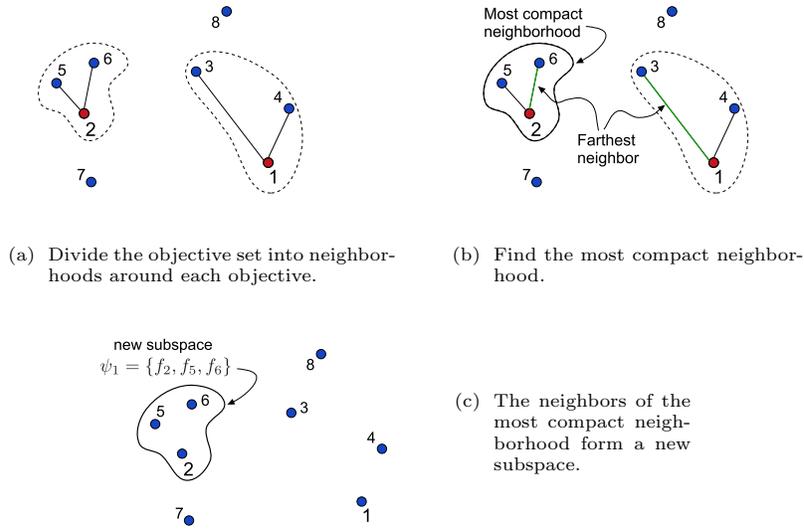


Figure 4: Basic strategy to create subspaces using conflict information.

Algorithm 3 Partitioning Using Conflict Information.

Input:

\mathcal{P} , population to compute the conflict matrix;
 Φ , set of objectives to be partitioned;
 N_S , number of subsets of the partition.

Output:

A partition, Ψ , of the set of objectives Φ .

procedure CONFLICTPARTITION(\mathcal{P}, Φ, N_S)

cMatrix \leftarrow COMPUTECONFLICTMATRIX(\mathcal{P})

ADJUSTMATRIX(cMatrix)

$k \leftarrow (|\Phi|/N_S) - 1$ // Num. of neighbors around each objective.

$\Phi' \leftarrow \Phi = \{f_1, \dots, f_M\}$ // Set of remaining objectives.

for $s \leftarrow 1$ until $N_S - 1$ **do**

// Create the neighbor list, L_{f_i} , for each objective f_i , where $L_{f_i}[j]$

// is the conflict value between f_i and the j -th objective in the list.

for each objective f_i in Φ' **do**

$L_{f_i} \leftarrow$ Ascending ordered list of the k -nearest neighbors
of f_i wrt the conflict using cMatrix.

end for

// Create a new subspace with the objectives of the most compact neighborhood.

$\psi_s \leftarrow L_{f_i} \cup \{f_i\} : L_{f_i}[k] \leq L_{f_j}[k], \forall f_j \in \Phi'$

$\Psi \leftarrow \Psi \cup \psi_s$ // Add the recently created subspace.

$\Phi' \leftarrow \Phi' - \psi_s$ // Update the set of remaining objectives.

end for

$\Psi \leftarrow \Psi \cup \Phi'$ // All the remaining objectives form the last subspace.

return Ψ

end procedure

Finally, we have to explain some necessary modifications to the correlation matrix in order to use it to select the most conflicting objectives properly. These modifications are carried out by the procedure `ADJUSTMATRIX` in Algorithm 3. If we used the original correlation matrix, it would be possible that some highly conflicting objectives might be placed in a subspace with a low conflict. For instance, if objective f_2 is in conflict with f_3 but not with f_1 , then f_2 would be very close to f_1 and, thus f_2 would be placed in a low conflicting subspace even if it is one of the most conflicting objectives. To overcome this problem we carried out the following process to the correlation matrix:

- Find the maximum conflict value $c_{i,max}$ of each row i in the matrix (i.e., the maximum negative correlation value for each objective).
- Add the value $c_{i,max}$ to the column i . This means that we are assuming that if objective f_i is in conflict with some objectives, then it is in conflict with all the objectives.

5. Experimental Results

5.1. Algorithms and Parameter Settings Employed

As mentioned before, we used NSGA-II’s framework to study the conflict-based partitioning strategy. In the first part of the experimental results we compare the conflict-based partitioning method and the Kendall-based strategy proposed by Purshouse and Fleming [33]. We implemented this strategy into NSGA-II following the description in [33]. Here we computed the Kendall coefficient taking care of tied data and we also carried out two-tailed tests at 99% level of confidence for determining independence between each pair of objectives. Additionally, we wanted to investigate the advantages or disadvantages of the conflict-based strategy with respect to the random strategy. Therefore, we compare the original NSGA-II, and the NSGA-II using the conflict and random partition strategies.

In the experiments we used the parameter values presented in Table 1. The number of generations, G_c , for each cycle (approximation + partition) is determined by the number of generations and cycles, that is $G_c = G_{max}/(\text{Num. cycles})$. Thus, for setting G_Φ and G_Ψ , we only have to set one of them. Here we used $G_\Phi = G_c - G_\Psi$. In the present study we adopted 70% of G_c for G_Ψ in order to emphasize the search on the subspaces and using only a few generations to approximate the entire Pareto front. For all the cases we carried out 30 runs for each MOEA. The results presented were averaged over the total number of executions.

5.2. Test Problems Employed

In order to show how the conflict-based strategy works, we used a test problem in which the conflicting objectives can be defined a priori by the

Common Values		Per Experiment Values			
Crossover rate	0.9		vs Kendall	vs Random	
Mutation rate	$1/n$	Population size	240	200	
Crossover index	15	Cycles	20	10	
Mutation index	20	Objectives	6	4-9	10-15
Generations	200	Subspaces	2 and 3	2	3
G_Ψ	70% of G_c				

Table 1: Parameter values employed in the computational experiments, where n is the number of variables and G_c the generations per cycle.

user. Namely, the problem DTLZ5(I, M) which is a variant proposed by Saxena and Deb [12] based on the original DTLZ5. In this problem, from a total of M objectives, only I of them are required to completely generate the Pareto front. The objectives of this problem can be classified in two subsets, the redundant subset $F_R = \{f_1, \dots, f_{M-I}\}$, and the necessary subset $F_N = \{f_{M-(I-1)}, f_{M-(I-2)}, \dots, f_M\}$, composed of the last $I - 1$ objectives. The Pareto front can be generated using only one objective from F_R and all the elements in F_N . Another feature of this problem is that all the objective are correlated in some way. There is conflict among every objective in F_N , but no conflict among elements in F_R . Nonetheless, there is conflict from elements in F_R to objectives in F_N . We also have used the problem WFG3(I, M) proposed by Huband et al. [22] which, as in the previous problem, the user can define the number of essential objectives. Since the obtained results are similar to those achieved using DTLZ5(I, M) and due to space constraints we do not show their results in this paper.

In order to compare the conflict-based strategy against the Kendall-based strategy we employed a problem, proposed by Purshouse and Fleming [33], in which an instance of the bi-objective ZDT1 problem (variables and objectives) is concatenated m times to create a MOP with $2m$ objectives. As result, only objectives in each pair (f_{2i-1}, f_{2i}) , for $i = 1, \dots, m$, are correlated, while there is no correlation between objectives in different pairs. This variant is denoted by c-ZDT1(m). For the computation results we adopted $30m$ variables for each instance of this problem.

Additionally, we employed three test problems in which the conflicting relation among the objectives is not known a priori. One of them is the problem DTLZ2_{BZ} proposed by Brockhoff and Zitzler [6] based on the original DTLZ2. When any of the objectives is removed from the original DTLZ2, the resulting Pareto front is reduced to a single nondominated solution. The DTLZ2_{BZ} variant avoids this problem, but it preserves the property that $\sum_{i=1}^M (z_i)^2 = 1$, for all $\mathbf{z} \in PF_{\text{opt}}$. The second problem is the multiobjective 0/1 Knapsack problem as formulated in [44]. For this problem we also varied the number of objectives from 4 to 15, and for all the instances, 300 items were adopted. Finally, we also adopted the problem WFG1 [22] which is a hard problem presenting a strong bias towards solutions away from the Pareto optimal set.

In problems DTLZ5(I, M), DTLZ2_{BZ}, WFG1 and WFG3(I, M) we em-

ployed a similar configuration in order to maintain test problem’s complexity for every number of objectives. Specifically, we fixed the number of distance-related variables² to 20. The number of position-related variables³ was set to $M - 1$ in DTLZ5(I, M) and DTLZ2_{BZ}, whereas for WFG1 and WFG3(I, M) we used $2(M - 1)$, which is the number of variables recommended in [22].

5.3. Quality Indicators Employed

Since in many-objective problems it is not possible to use 3D plots to help in the interpretation of results, we have to rely on the results obtained by the quality indicators. For this reason, we used several indicators, and in some cases, we resort to parallel coordinates plots to interpret the results.

In order to evaluate the convergence achieved by the MOEAs we used the *generational distance* (GD). Since DTLZ2_{BZ} and DTLZ5(I, M) have the property $\sum_{i=1}^M (z_i)^2 = 1$ for all $\mathbf{z} \in PF_{\text{opt}}$ the generational distance was computed using $GD = (\sum_{i=1}^M (z_i)^2 / |PF_{\text{approx}}|) - 1$. In the case of the Knapsack problem, the usual definition of generational distance was adopted, using as reference Pareto front, the resulting nondominated individuals of the union of PF_{approx} obtained by the three algorithms in all the runs for a given test problem. As for WFG1, we compared the output of the algorithms against a sample of 100000 well distributed points of the optimal Pareto front for any number of objectives.

Additionally, to directly compare the convergence of the MOEAs in all the test problems, we utilized the additive ϵ -indicator [45]. This indicator is defined as $I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{\forall \mathbf{z}^2 \in B \exists \mathbf{z}^1 \in A : \mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2\}$ for two nondominated sets A and B , where $\mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2$ iff $\forall i : z_i^1 \leq \epsilon + z_i^2$, for a given ϵ . In general, $I_{\epsilon+}(A, B) \neq I_{\epsilon+}(B, A)$ so we have to compute both values. The smaller $I_{\epsilon+}(A, B)$ and larger $I_{\epsilon+}(B, A)$, the better A over B .

In order to evaluate diversity, we used the *inverted generational distance* (IGD). Similarly to GD, for this indicator we used the nondominated solutions of all the PF_{approx} generated for a given test problem as reference Pareto front. In the case of WFG1 we used the sample of the PF_{opt} described above.

Besides IGD, another often used indicator to assess both convergence and diversity is the *hypervolume* indicator (HV). For DTLZ2_{BZ}, DTLZ5(I, M) the reference point was $\mathbf{z}^{\text{ref}} = 1.5^M$. The results presented correspond to the normalized hypervolume using the enclosed hypervolume between the ideal point $\mathbf{z}^* = 0^M$ and the reference point. For the knapsack problem, the reference point was formed using the worst value in each objective of all the PF_{approx} generated for all the algorithms. In this case, the hypervolume was normalized using the hypervolume yielded by NSGA-II. Due to the high computational complexity of the hypervolume with respect to number of objectives, we only computed this indicator for 4 to 10 objectives.

²Distance-related variables control the progress towards the Pareto optimal front.

³Position-related variables generate solutions in the same local Pareto front.

5.4. Comparison Against the Kendall-based Strategy

Since the Kendall-based strategy is intended for MOPs in which there are groups of independent objectives, in problems where all the objectives are correlated (either by conflict or non-conflict), they are grouped together. As a consequence, from the adopted test problems in this study, the Kendall-based strategy can be successfully applied only on $c\text{-ZDT1}(m)$. In the other problems, even in $\text{DTLZ5}(I, M)$ or $\text{WFG3}(I, M)$, this strategy has no effect in the search of NSGA-II. Therefore, in this section we compare both approaches only on $c\text{-ZDT1}(m)$.

In this problem the ideal partition groups the objectives into pairs. Therefore, the performance of the conflict-based strategy depends on the number of subspaces provided by the user. In order to make a fair comparison, two configurations were tested for the conflict-based strategy: one with subspaces of size 2 (the best case), and another one in which a pair of correlated objectives will always be separated (the worst case). For this purpose we used an instance of $c\text{-ZDT1}(m)$ with $m = 3$ (i.e., 6 objectives) to create configurations with 3 subspaces and 2 subspaces.

Like in the study presented in [33], here the objective partition is updated at every generation. On the other hand, since the alternation between the partitioning and approximation phases is a key element in the conflict-based strategy, for this method the partition is updated every 10 generations (i.e., 20 cycles of the partitioning/approximation phase). In preliminary experiments we also tested the Kendall-based strategy revising the partitions every 10 generations. However, updating at every generation provided better results.

In Table 2 we present the performance results using hypervolume, generational distance and inverted generational distance. As it can be seen, the conflict-based method outperformed the Kendall-based strategy with respect to the three indicators. As expected, the configuration with 3 subspaces achieves the best results. However, the performance using 2 subspaces has the second best performance. This is interesting since despite the fact that with 2 subspaces at least a pair of correlated objectives is divided, this configuration also outperforms the Kendall strategy.

This result can be explained by monitoring the percentage of times over the 30 runs that the ideal objective partition, $\{\{f_1, f_2\}, \{f_3, f_4\}, \{f_5, f_6\}\}$, is identified. Figure 5 shows this percentage at each generation for each partitioning method. Since the conflict-based strategy with 2 subspaces always separates a pair of correlated objectives, it can not generate the ideal partition. However, for comparison purposes, we plot the percentage of partitions where, excluding a single separated pair, the other objectives are correctly grouped in the same subset. The figure shows that the percentage of ideal partitions generated by the Kendall-based strategy is considerably low compared with the conflict-based method with 3 subspaces. This is an advantage of the conflict-based method with 3 subspaces because during a larger number of generations the search is focused in the correct objective subsets. Regarding the configuration with 2 subspaces, although there is always a broken pair of correlated objectives, the

others are correctly grouped nearly 100% of the times. In addition, as shown in Figure 6, the separated pair of objectives is not the same at each generation, and therefore, a missing part of the Pareto front can be covered in following generations. Also note that the figure shows a case (the sixth partition) in which all the objectives were incorrectly grouped.

		Kendall	Conflict 2s	Conflict 3s
HV	Min	5.6673	6.5374	6.5362
	Max	6.6144	6.6841	6.6928
	Mean	6.2436	6.6122	6.6251
	Std	0.2592	0.0398	0.0343
GD	Min	0.1771	0.1220	0.1213
	Max	1.3417	0.2487	0.2267
	Mean	0.5433	0.1805	0.1596
	Std	0.3030	0.0360	0.0242
IGD ($\times 10^{-2}$)	Min	1.0770	0.2590	0.0670
	Max	13.7710	4.5000	3.4390
	Mean	4.6692	1.1051	1.0875
	Std	2.9982	0.7245	0.7396

Table 2: Performance comparison between the Kendall-based and the Conflict-based strategies (2 and 3 subspaces).

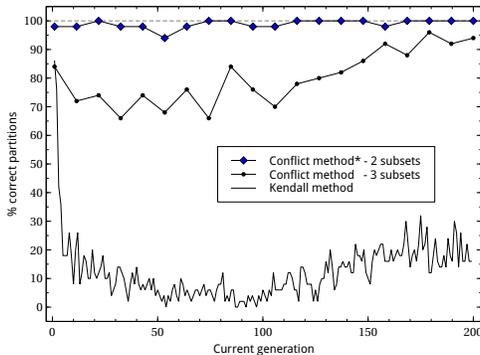


Figure 5: Percentage of correct partitions using the Kendall and Conflict strategies with 2 and 3 subspaces. For the configuration with 2 subspaces the separation of one correlated pair of objectives is not taken into account.

In order to investigate why the Kendall method fail so often to generate the ideal partition we can analyze the typical Kendall correlation coefficients among the objectives in $c\text{-ZDT1}(m)$. Here, we used $m = 2$ so that the related objectives are the pairs (f_1, f_2) and (f_3, f_4) . In Figure 7 we show a typical approximation of $c\text{-ZDT1}(4)$ in different subspaces. The figure also shows the computed Kendall correlation coefficient τ and the p -value used to determine

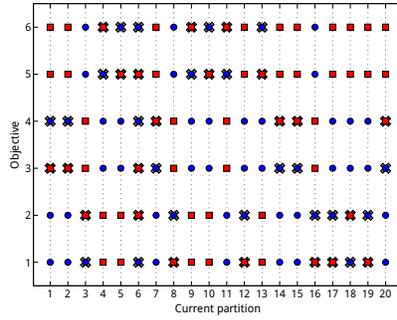


Figure 6: Partitions generated by the conflict strategy using 2 subspaces on the problem c-ZDT1(6). The cross symbol indicates objectives assigned to a wrong subspace (i.e., separated from its pair).

independence⁴.

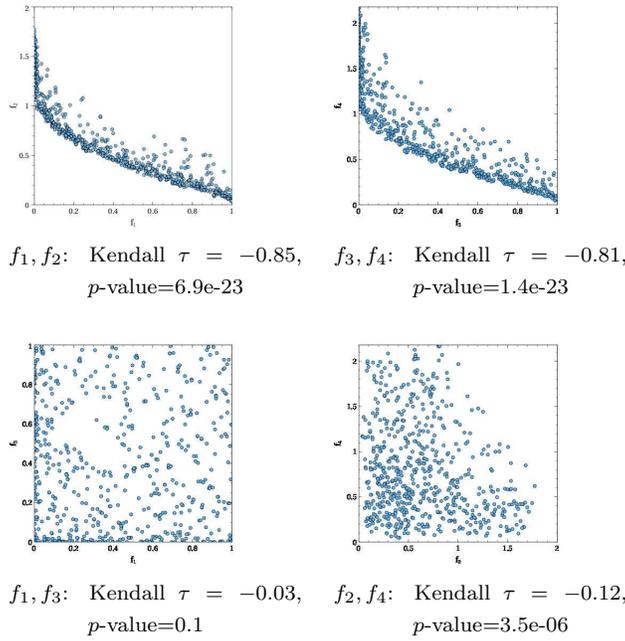


Figure 7: Approximation of c-ZDT1(4) projected on different subspaces.

Using a significance level $\alpha = 0.01$, as expected, the objectives in each pair (f_1, f_2) and (f_3, f_4) are dependent since $p\text{-value} < \alpha$. However, for (f_2, f_4) also

⁴If $p\text{-value} < \text{significance level}$, the objectives are considered dependent.

p -value $< \alpha$. Thus, f_4 would be included in the set $\{f_1, f_2\}$, and since f_3 is related to f_4 , it is also included. As result, all the objectives are grouped in the same subset. Here, the problem is that the extent of correlation between the objectives pairs is not considered. For example, the magnitude of the correlation between (f_1, f_2) (-0.85) is considerably higher than that of (f_2, f_4) (-0.12). However, since both p -values are below the significance level, the objectives in each pair are classified as dependent.

5.5. Problems With A Priori Known Conflict

In this section we present the experimental results using the problem DTLZ5(I, M). In these experiments, we used $I = 4$ conflicting objectives from a total of $M = 4, \dots, 15$ objectives. For 4 to 9 objectives, 2 subspaces were used, whereas for 10 to 15 objectives, we employed 3 subspaces.

First, we want to show that the conflict-based strategy was able to correctly identify the conflicting objectives in most of the partitions generated during the search process. Figure 8 shows the subspaces generated by the conflict-based and random partition strategies during the search process. In this example, there is a total of $M = 8$ objectives. The conflicting objectives are objectives 6 to 8 and any of the other objectives. The objectives in the most conflicting subspace are denoted by crosses, and the other subspace is denoted by circles.

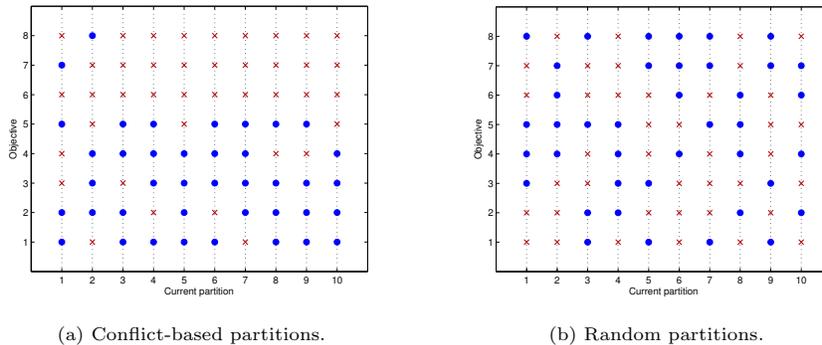


Figure 8: Subspaces generated using the conflict and random partition strategies on problem DTLZ5($I = 4, M = 8$). Objectives 6 to 8 and any of the other objectives are the conflicting objectives.

In Figure 8(a) we can see that in the first partitions generated, some of the objectives were assigned to the wrong subspace. The reason of this behavior is that in the first generations the current population does not yet represent an accurate sample of the real shape of the Pareto front. However, as the search progresses, the input PF_{approx} used to estimate the conflict approaches the true Pareto front. Therefore, in the last stages of the search, the conflict-based strategy was able to create the correct partition. On the other hand, by using the random strategy, the chances that the correct partition is created are very

low. In the example shown in Figure 8(b), only the fourth partition generated contains the correct subspaces. Consequently, in most of the generations of the search, the selected parents emphasize objective subspaces that do not maximize the contribution to form the true Pareto front.

Figure 9 shows the results for the generational distance. The most evident fact in that plot is that the convergence of NSGA-II degrades dramatically when the number of objectives is more than 6. In fact, the convergence in terms of GD tends to diverge. A possible reason of this behavior is the generation of *dominance resistant solutions*⁵ (DRSs) in $DTLZ5(I, M)$. These solutions are far from the true Pareto front, however, since they are nondominated solutions, they are candidates to form the new parent population. Since DRSs are boundary solutions, most of them will have the best crowding value. Therefore, these solutions will always be included in the new parent population. As mentioned in the introduction, the proportion of nondominated solutions in a population increases exponentially with respect to the number of objectives. As a result, this problem gets more difficult when the number of objectives grows. In contrast, it seems that the GD values using any of the partition strategies, are not affected by the number of objectives. In particular, we can see that the convergence obtained by using the conflict-based partition strategy is better than the one achieved by the random strategy.

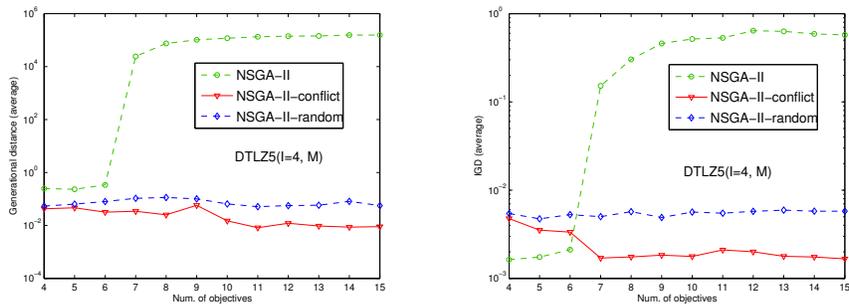


Figure 9: Generational distance (left) and inverted generational distance (right) results in problem $DTLZ5(I = 4, M)$. From 4 to 9 objectives a partition with 2 subspaces was generated, while for 10 to 15 objectives one with 3 subspaces.

Nonetheless, this difference is only marginal. By inspecting the parallel coordinate plot presented in Figure 10 we realized that NSGA-II with the random strategy converges to the extremes of the Pareto front. That is, most of the solutions are close to 0 or 1 in one objective, but very few solutions are generated in between. Within the context of the objective space, that would mean that solutions in the middle region of the Pareto front are not yielded. In contrast, as

⁵Dominance resistant solutions are those with a poor value in at least one of the objectives, but with near optimal values in the others.

shown in Figure 10, the conflict-based strategy covers a wide range of trade-offs between the objectives. Solutions in the middle region represent a range of trade-offs in which one objective gradually improves while other gradually gets worse.

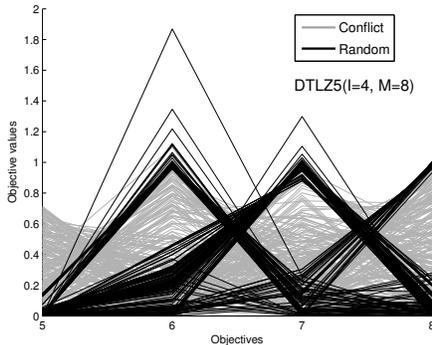


Figure 10: DTLZ5($I = 4, M = 8$): Parallel coordinate plot of the Pareto front approximations obtained with the random and the conflict partition strategies.

In order to measure this situation, we compute the inverted generational distance. Figure 9 shows that the conflict-based partition strategy achieves better values in terms of the inverted generational distance. This indicates a better distribution using the conflict-based partition strategy.

Next, we present the convergence assessment using the additive ϵ -indicator. To some degree, this indicator also takes into account the distribution of the Pareto front approximations compared. For example, let A be a nondominated set which is well-distributed along the entire Pareto front, and B a subset of A concentrated on a small region of the Pareto front. A already weakly dominates every solution in B , however some positive ϵ value must be added to A in such a way that B weakly dominates every solution in A .

The results of the ϵ -indicator are presented in Figure 11. We can interpret these results as follows. $I_{\epsilon+}(A, B)$ is the subplot located in row A , and column B of the matrix. The boxes in each subplot depict the results for each number of objectives considered. As we can see, the results of the ϵ -indicator indicate that NSGA-II is clearly outperformed by NSGA-II using either of the partition strategies. With respect to the comparison of both partition strategies we can observe that the average ϵ values using the conflict-based strategy are better than those achieved by the random strategy, especially for 6 or more objectives. That is, $I_{\epsilon+}(\text{Conflict}, \text{Random}) < I_{\epsilon+}(\text{Random}, \text{Conflict})$ for any number of objectives.

Finally, we present the results of the hypervolume indicator. Since the hypervolume considers both convergence and distribution to assess two nondominated sets, as we can see in Figure 12, the conflict-based partition strategy outperforms the random strategy.

Like in the previously analyzed indicators, the original NSGA-II achieved a

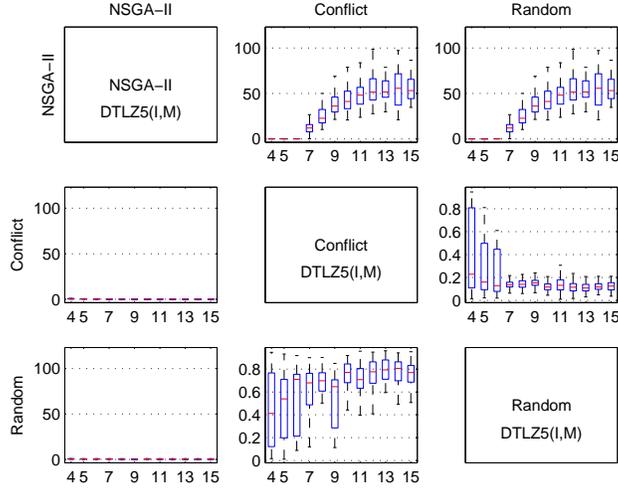


Figure 11: DTLZ5($I = 4, M$): ϵ -indicator results. The vertical axis of each subplot denotes the corresponding ϵ value, and the horizontal axis the boxplot for each number of objectives considered.

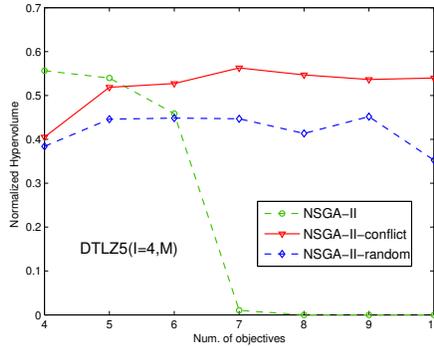


Figure 12: Normalized Hypervolume results for DTLZ5($I = 4, M = 8$).

poor performance in terms of the hypervolume indicator. However, it is worth noting a recurrent behavior using the hypervolume and other indicators. That is, for less than 5 or 6 objectives, NSGA-II presents a better or similar performance than that achieved by using a partition strategy. There are two facts that explain this behavior. Firstly, that the NSGA-II is still able to deal with that lower number of objectives. Second, since there are 4 conflicting objectives for 4 to 6 objectives, using 2 subspaces it is not possible that all the conflicting objectives are grouped into one subspace. Therefore, the trade-offs between objectives in different subspaces are not well represented. This suggests that it is convenient

to assign all the highly correlated objectives to a single subspace. However, a large subspace might surpass the capacities of the underlying MOEA. In the next section we will analyze the effect of the size of the subspaces in the partition.

5.6. Effect of the Size of the Subspaces

In this section we analyze if it is better to have all the conflicting objectives together although in a large subspace, or small subspaces although the conflicting objectives are in different subspaces. To this end, we used $DTLZ5(I, M)$ $M = 24$ objectives and $I = 12$ objectives in conflict. Then, we compare two different partitions, namely, one with two subspaces with 12 objectives each, and another one with 6 subspaces with 4 objectives each.

First, we want to show that both types of partitions are able to identify the conflicting objectives. However, in most cases, the partition with two subspaces achieved a better identification of the conflicting objectives. Figs. 13(a) and 13(b) show an example of the partitions created using 2 and 6 subspaces, respectively. In order to easily verify if the objectives in the partition with 6 subspaces were correctly identified, the objectives in subspaces 1 to 3 are marked with a cross, and those in subspaces 4 to 6 (conflicting objectives) with circles.

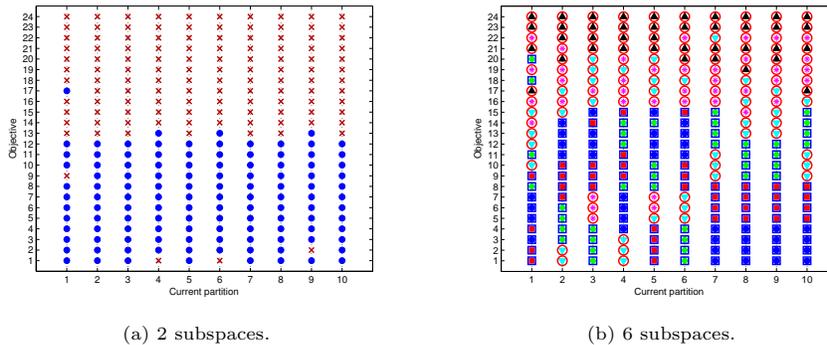


Figure 13: $DTLZ5(I = 12, M = 24)$: Subspaces generated using 2 subspaces with 12 objectives each, and 6 subspaces with 4 objectives each.

Figure 14 shows the progress of the generational distance indicator during the search process. Similarly to previous experiments, due to the dominance resistant solutions, NSGA-II diverges with respect to GD. However, what we want to emphasize is the fact that each partition strategy achieved a better convergence using 6 subspaces with 4 objectives. This suggests that it is preferable to have subspaces of moderate size, even if highly conflicting objectives have to be assigned to different subspaces. The optimal size of the subspaces depends on the capacities of the underlying MOEA. For example, based on the experimental results observed so far, an appropriate size of the subspaces for NSGA-II would be between 4 and 6 objectives. However, for other MOEAs, like the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [43] or the Pareto

Archived Evolution Strategy (PAES) [28], the optimal subspace size might be different.

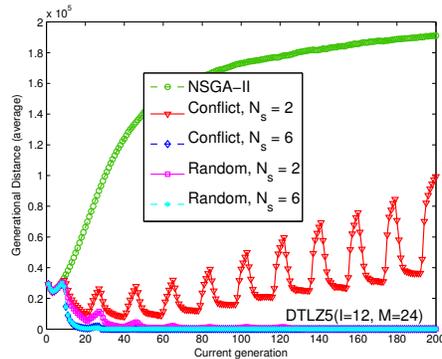


Figure 14: DTLZ5($I = 12, M = 24$): Online Generational Distance using a partition with 2 subspaces and another one with 6 subspaces.

Regarding the conflict and random strategies, it can be seen that the conflict-based strategy also diverges when using 2 subspaces. However, the random-based strategy with the same number of subspaces obtains good results. In order to find the reason for this behavior, we will analyze the performance using other quality indicators and plots. The parallel coordinate plot shown in Figure 15 indicates that by using a random strategy the solutions converge to the extremes of only a pair of objectives. In this plot only the conflicting objectives are plotted (from objective 13 to 24). In the ideal case, lines crossing from objective 13 to 24 in the range $[0, 1]$ should appear. In contrast to the random strategy, the conflict-based strategy finds solutions that optimize more objectives and cover the mid trade-off regions of the Pareto front.

In order to quantitatively assess the distribution, we compare the algorithms using the inverted generational distance, whose results are shown in Table 3. Although the obtained generational distance of the conflict and random strategies are similar using 6 subspaces (see Figure 14), the results of the inverted generational distance shown in Table 3 suggest that the conflict strategy with 6 subspaces achieved a better distribution of the solutions than the random strategy with 6 subspaces. In fact, the random strategy with 2 subspaces achieved a better IGD than the one yield using 6 objectives.

Based on the ϵ -indicator results shown in Table 4, we can confirm that both partition strategies have a better performance using partitions with 6 subspaces. In the same way, the conflict strategy outperformed the other algorithms in terms of the ϵ -indicator. The negative results in the column of NSGA-II indicate that, on average, the Pareto front approximations yielded by the conflict strategy with 6 subspaces and by both random strategies, dominate the Pareto front approximations obtained by NSGA-II.

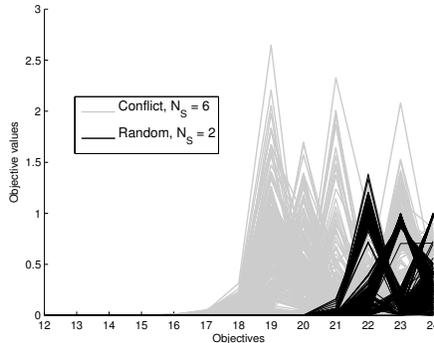


Figure 15: DTLZ5($I = 12, M = 24$): Parallel coordinate plot of the Pareto front approximations obtained by the conflict-based strategy with 6 subspaces, and the random strategy with 2 subspaces.

	NSGA-II	Conflict		Random	
		$N_S = 2$	$N_S = 6$	$N_S = 2$	$N_S = 6$
Mean	0.18005	0.00838	0.00570	0.00682	0.00769
Std. Dev.	0.04695	0.00010	0.00047	0.00092	0.00029
Worst	0.26993	0.00849	0.00758	0.00791	0.00819
Best	0.07868	0.00788	0.00454	0.00591	0.00716

Table 3: IGD values for DTLZ5($I = 12, M = 24$) using 2 and 6 subspaces in each of the partitioning strategies, namely, random- and conflict-based partitions.

5.7. Problems With A Priori Unknown Conflict

In this section, we analyze the performance of the conflict and random partition strategies in problems in which the conflict relation among objectives is not known a priori. That is, the DTLZ2_{BZ}, WFG1 and Knapsack problems.

Based on the symmetrical geometry of DTLZ2_{BZ}'s Pareto front (which is a sphere), it seems that the conflict between every pair of objectives is very similar. Therefore, we would expect that both partition strategies present a similar performance. Figure 16 shows the results for the generational distance and the inverted generational distance obtained in problem DTLZ2_{BZ}. As we expected, the experimental results show that both partition strategies obtained a similar performance in both indicators. However, the conflict-based strategy achieved a slightly better performance.

In a similar way, both algorithms achieved similar results with respect to the hypervolume and the ϵ -indicator (see Figs. 17 and 18).

Although in DTLZ2_{BZ}, the conflict information was not useful to create the partitions, as we will see, in the Knapsack problem there is an interesting conflict relation among the objectives that allows the conflict-based strategy to perform better than the random strategy. Figure 19 shows the subspaces generated by the conflict strategy on the Knapsack problem with 9 objectives.

$I_{\epsilon+}(A, B)$	NSGA-II	Cft	Cft	Rnd	Rnd
		$N_S = 2$	$N_S = 6$	$N_S = 2$	$N_S = 6$
NSGA-II	x	14.4030	14.4030	14.4030	14.4030
Cft, $N_S = 2$	0.1977	x	0.9644	0.9644	0.9623
Cft, $N_S = 6$	-6.73e-6	0.0372	x	0.1974	0.1012
Rnd, $N_S = 2$	-6.73e-6	0.0107	0.4845	x	0.1407
Rnd, $N_S = 6$	-6.73e-6	0.6009	0.6010	0.6011	x

Table 4: $I_{\epsilon+}$ values for DTLZ5($I = 12, M = 24$) using 2 and 6 subspaces in each of the partitioning strategies, namely, random- and conflict-based partitions.

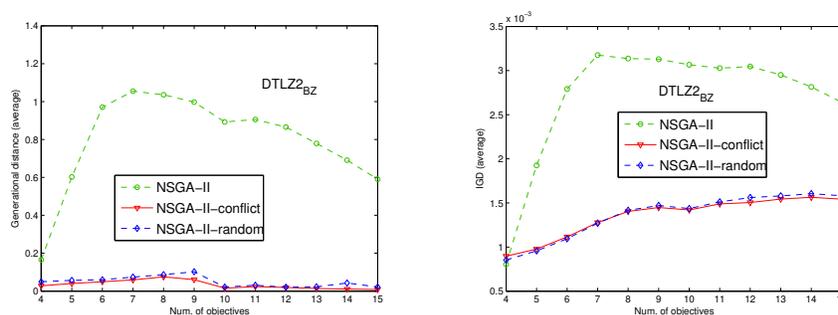


Figure 16: Generational distance and inverted generational distance in problem DTLZ2_{BZ}.

As can be seen, as the search progresses, a particular partition is formed recurrently, namely $\Psi_3 = \{\{4, 5, 8\}, \{1, 3, 9\}, \{2, 6, 7\}\}$, where $\{4, 5, 8\}$ is the least conflicting subspace, and $\{2, 6, 7\}$ is the most conflicting one. This suggests that the conflict between certain objectives is considerably larger than the conflict between other objectives.

In order to measure the contribution of each subspace to the total conflict in the problem, we compute the following measure. For each subspace we compute the sum of the conflict between each pair of its objectives. We consider this sum as the conflict degree of each subspace. The sum of the conflict degree of each subspace is the total conflict of the problem. The ratio of the conflict degree of each subspace and the total conflict is called the conflict contribution. In Figure 20, we can clearly see that subspace 3 has a larger conflict contribution with respect to the other subspaces.

From the results obtained in the generational distance and in the inverted generational distance (see Figure 21) we can say that the conflict-based partition strategy achieved better Pareto front approximations than the random-based strategy in terms of both convergence and distribution.

The results obtained with the hypervolume indicator (see Figure 18) confirm that the conflict-based strategy outperformed the random strategy. We can conclude that the differences in the degrees of conflict between each pair

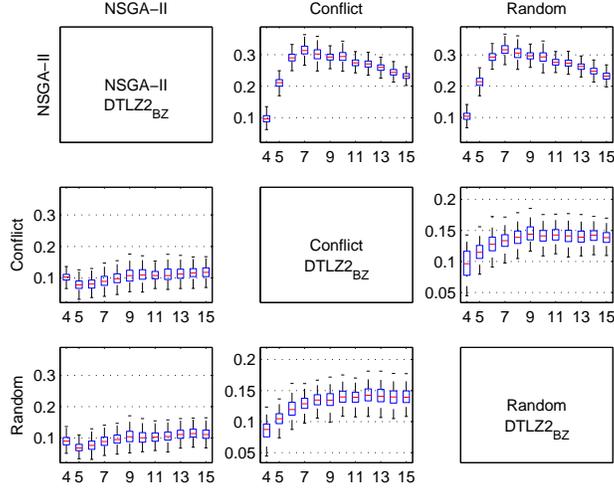


Figure 17: $DTLZ2_{BZ}$: ϵ -indicator results. The vertical axis of each subplot denotes the corresponding ϵ value, and the horizontal axis the boxplot for each number of objectives considered.

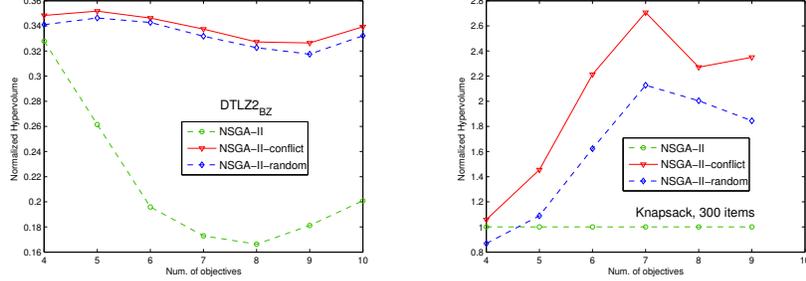


Figure 18: Normalized Hypervolume of $DTLZ2_{BZ}$ (left) and the Knapsack problem (right). For the Knapsack problem the hypervolume values were normalized with respect to the hypervolume achieved by the NSGA-II.

objectives was used by the conflict-based strategy to obtain better results than those obtained using a random partition.

Finally, we present the results of the performance of the three variants of NSGA-II on problem WFG1. This is a very hard problem since it presents a strong bias towards a variable space's region away from the Pareto optimal set. Similarly to $DTLZ2_{BZ}$, the symmetrical shape of WFG1's Pareto front suggest that objective conflict information will not represent a decisive advantage for the conflict-based strategy. However, searching in different subspaces might still

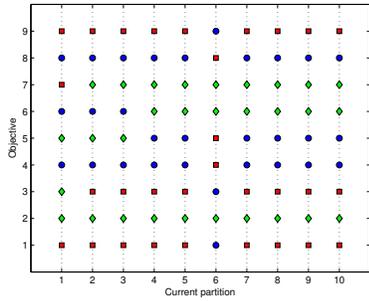


Figure 19: Three generated subspaces by the conflict-based partition strategy on the Knapsack problem.

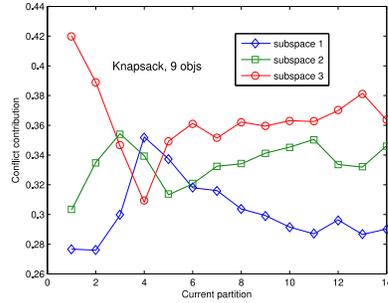


Figure 20: Conflict contribution of each of the three subspaces generated using the conflict partition strategy.

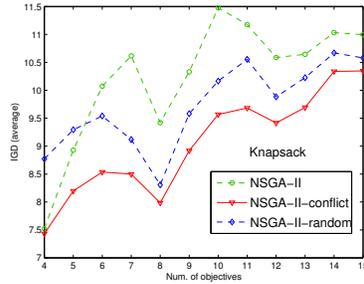
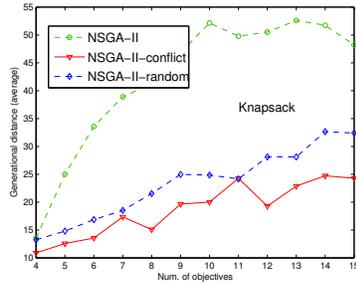


Figure 21: Generational distance and inverted generational distance in the Knapsack problem.

help NSGA-II to converge faster towards the Pareto front, especially when the number of objective is high.

As expected, the generational distance shown in Figure 22 indicates that the performance of both partitioning strategies is very similar for all the objectives considered and considerably better than NSGA-II when the partitioning scheme is not used. On the other hand, the results of the inverted generational distance (see Figure 22) suggest that conflict-based strategy achieved a better distribution over the Pareto front.

6. Conclusions and Future Work

In this paper, we have proposed a new strategy to partition the objective space into small subproblems in order to deal with many-objective problems. The new strategy creates objective subspaces based on the analysis of the conflict information obtained from the Pareto front approximation. Additionally, we introduced a new version of the general partitioning scheme in order to improve

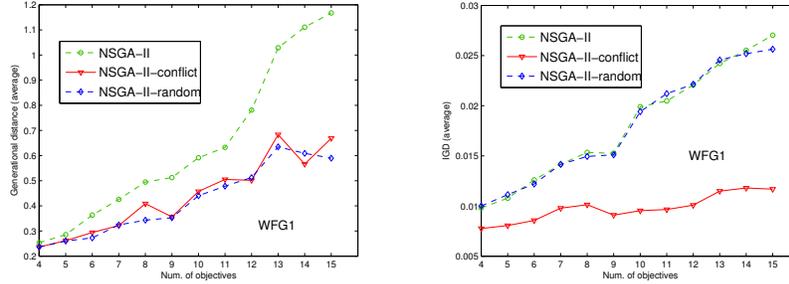


Figure 22: Generational distance and inverted generational distance in the WFG1 problem.

the distribution of the solution over the entire Pareto front. The general partitioning scheme was implemented within NSGA-II’s framework. In order to evaluate the effectiveness of the new conflict-based partition strategy, we compared its performance against three algorithms: a partitioning method based on the Kendall correlation, a random partitioning strategy and the original NSGA-II.

The experimental results showed that both the conflict-based and random partition strategies outperformed NSGA-II in all the test problems considered in this study. While NSGA-II even diverged in some test problems, the NSGA-II using any of the partition strategies maintained a good convergence regardless of the number of objectives. Regarding the two partition strategies, the conflict-based partition strategy achieved a better distribution of the solutions than that achieved by the random strategy. In some problems, by using the random strategy, convergence was concentrated on the extremes of the Pareto front. Finally, the conflict-based strategy outperformed the Kendall-based method on a separable problem.

In problems in which the degree of conflict between pairs of objectives was different, the conflict-based strategy presented a better performance. It is important to note, that in the case of the Knapsack problem, in which the conflict relation among the objectives is not known a priori, the conflict-based strategy was able to detect important dependencies among the objectives in terms of the conflict. The extracted conflict information allowed our proposed conflict-based partition strategy to achieve better results than the other algorithms.

Initially, one may think that grouping all the highly conflicting objectives in one subspace is a better choice. However, the experimental results showed that the best size of the subspaces considerably depends on the scalability of the underlying MOEA. For instance, if the underlying MOEA has good performance up to 5 objectives, the size of each subspace should not exceed that limit.

From the experimental results we realized that in some problems the contribution of some subspaces to the overall conflict of the problem was very small. Therefore, an equal distribution of the resources (e.g., proportion of parents, number of generations) to the subspaces might not be a good idea. In this

sense, as part of our future work, we plan to exploit the conflict information in order to automatically adapt the proportion of resources granted to each subspace. Similarly, we want to use the conflict information to determine the best size of each subspace in the objective partition. Although the proposed partitioning framework achieved promising results, it would be interesting to compare our approach against other algorithms that have also shown a good performance on many-objective optimization problems (e.g. [24]).

- [1] Agrell, P.J., 1997. On redundancy in multi criteria decision making. *European Journal of Operational Research* 98, 571–586.
- [2] Aguirre, H.E., Tanaka, K., 2009a. Many-Objective Optimization by Space Partitioning and Adaptive e-Ranking on MNK-Landscapes, in: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.K., Sevaux, M. (Eds.), *EMO 2009*. Springer, Nantes, France. volume 5467 of *LNCS*, pp. 407–422.
- [3] Aguirre, H.E., Tanaka, K., 2009b. Space partitioning with adaptive e-ranking and substitute distance assignments: a comparative study on many-objective MNK-landscapes, in: *GECCO'2009*, ACM Press, Montreal, Canada. pp. 547–554.
- [4] Brockhoff, D., Zitzler, E., 2006a. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization, in: *Parallel Problem Solving from Nature IX*, Springer-Verlag. pp. 533–542.
- [5] Brockhoff, D., Zitzler, E., 2006b. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization, in: Runarsson, T.P., Beyer, H.G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (Eds.), *PPSN IX, 9th International Conference*. Springer, Reykjavík, Iceland. volume 4193 of *LNCS*, pp. 533–542.
- [6] Brockhoff, D., Zitzler, E., 2007. Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods, in: *CEC'2007*, IEEE Press, Singapore. pp. 2086–2093.
- [7] Brualdi, R.A., 1977. *Introductory Combinatorics*. North-Holland.
- [8] Carlsson, C., Fullér, R., 1995. Multiple criteria decision making: The case for interdependence. *Computers and Operations Research* 22, 251–260.
- [9] Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A., 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York. second edition.
- [10] Deb, K., Agrawal, S., Pratab, A., Meyarivan, T., 2000. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, in: Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.P. (Eds.), *Parallel Problem Solving from Nature VI*, Springer, Paris, France. pp. 849–858.
- [11] Deb, K., Kumar, A., 2007. Light Beam Search Based Multi-objective Optimization using Evolutionary Algorithms, in: *CEC'2007*, IEEE Press, Singapore. pp. 2125–2132.

- [12] Deb, K., Saxena, D.K., 2006. Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems, in: CEC'2006, IEEE Press, Vancouver, BC, Canada. pp. 3353–3360.
- [13] Deb, K., Thiele, L., Laumanns, M., Zitzler, E., 2002. Scalable Multi-Objective Optimization Test Problems, in: CEC'2002, IEEE Press, Piscataway, New Jersey. pp. 825–830.
- [14] di Pierro, F., 2006. Many-Objective Evolutionary Algorithms and Applications to Water Resources Engineering. Ph.D. thesis. School of Engineering, Computer Science and Mathematics. UK.
- [15] Edgeworth, F.Y., 1881. Mathematical Physics. P. Keagan.
- [16] Farina, M., Amato, P., 2002. On the Optimal Solution Definition for Many-criteria Optimization Problems, in: Proceedings of the NAFIPS-FLINT International Conference'2002, IEEE Press, Piscataway, New Jersey. pp. 233–238.
- [17] Figueira, J.R., Liefooghe, A., Talbi, E.G., Wierzbicki, A.P., 2010. A parallel multiple reference point approach for multi-objective optimization. European Journal of Operational Research 205, 390–400.
- [18] Fonseca, C.M., Fleming, P.J., 1998. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part II: A Application Example. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28, 38–47.
- [19] Gal, T., Hanne, T., 1999. Consequences of dropping nonessential objectives for the application of mcdm methods. European Journal of Operational Research 119, 373–378.
- [20] Gal, T., Leberling, H., 1977. Redundant objective functions in linear vector maximum problems and their determination. European Journal of Operational Research 1, 176–184.
- [21] Gunawan, S., Farhang-Mehr, A., Azarm, S., 2003. Multi-level multi-objective genetic algorithm using entropy to preserve diversity, in: Fonseca, C., Fleming, P., Zitzler, E., Thiele, L., Deb, K. (Eds.), Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg. volume 2632 of *LNCS*, pp. 148–161.
- [22] Huband, S., Barone, L., While, L., Hingston, P., 2005. A Scalable Multi-objective Test Problem Toolkit, in: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (Eds.), EMO 2005, Springer, Guanajuato, México. pp. 280–295.
- [23] Hughes, E.J., 2005. Evolutionary Many-Objective Optimisation: Many Once or One Many?, in: CEC'2005, IEEE Press, Edinburgh, Scotland. pp. 222–227.
- [24] Hughes, E.J., 2007a. MSOPS-II: A General-Purpose Many-Objective Optimiser, in: CEC'2007, IEEE Press, Singapore. pp. 3944–3951.
- [25] Hughes, E.J., 2007b. Radar Waveform Optimisation as a Many-Objective Application Benchmark, in: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (Eds.), EMO 2007, Springer, Matshushima, Japan. pp. 700–714.

- [26] Ikeda, K., Kita, H., Kobayashi, S., 2001. Failure of Pareto-based MOEAs: Does Non-dominated Really Mean Near to Optimal?, in: CEC'2001, IEEE Press, Piscataway, New Jersey. pp. 957–962.
- [27] Knowles, J., Corne, D., 2007. Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization, in: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (Eds.), EMO 2007, Springer, Matshushima, Japan. pp. 757–771.
- [28] Knowles, J.D., Corne, D.W., 2000. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation* 8, 149–172.
- [29] López Jaimes, A., Coello Coello, C.A., Chakraborty, D., 2008. Objective Reduction Using a Feature Selection Technique, in: GECCO'2008, ACM Press, Atlanta, USA. pp. 674–680.
- [30] López Jaimes, A., Coello Coello, C.A., Urías Barrientos, J.E., 2009. Online Objective Reduction to Deal with Many-Objective Problems, in: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.K., Sevaux, M. (Eds.), EMO 2009. Springer, Nantes, France. volume 5467 of *LNCS*, pp. 423–437.
- [31] Pareto, V., 1896. *Cours D'Economie Politique*. F. Rouge.
- [32] Praditwong, K., Yao, X., 2007. How well do multi-objective evolutionary algorithms scale to large problems, in: CEC'2007, IEEE Press. pp. 3959–3966.
- [33] Purshouse, R.C., Fleming, P.J., 2003. An Adaptive Divide-and-Conquer Methodology for Evolutionary Multi-criterion Optimisation, in: Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., Deb, K. (Eds.), *Evolutionary Multi-Criterion Optimization*, Springer Berlin Heidelberg. pp. 133–147.
- [34] Purshouse, R.C., Fleming, P.J., 2007. On the Evolutionary Optimization of Many Conflicting Objectives. *IEEE Transactions on Evolutionary Algorithms* 11, 770–784.
- [35] Sato, H., Aguirre, H.E., Tanaka, K., 2007. Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs, in: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (Eds.), EMO 2007, Springer, Matshushima, Japan. pp. 5–20.
- [36] Saxena, D., Duro, J., Tiwari, A., Deb, K., Zhang, Q., 2013. Objective Reduction in Many-Objective Optimization: Linear and Nonlinear Algorithms. *Evolutionary Computation*, *IEEE Transactions on* 17, 77–99.
- [37] Schaffer, J.D., 1985. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, in: *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum. pp. 93–100.
- [38] Schütze, O., Lara, A., Coello Coello, C.A., 2011. On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *Evolutionary Computation*, *IEEE Transactions on* 15, 444–455.

- [39] Stewart, T., Bandte, O., Braun, H., Chakraborti, N., Ehrgott, M., Göbelt, M., Jin, Y., Nakayama, H., Poles, S., Stefano, D.D., 2009. Real-World Applications of Multiobjective Optimization, in: Branke, J., Deb, K., Miettinen, K., Slowinski, R. (Eds.), Multiobjective Optimization. Interactive and Evolutionary Approaches. Springer, Berlin, Germany. volume 5252 of *LNCS*, pp. 285–327.
- [40] Teytaud, O., 2007. On the hardness of offline multi-objective optimization. *Evolutionary Computation* 15, 475–491.
- [41] Thiele, L., Miettinen, K., Korhonen, P.J., Molina, J., 2009. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation* 17, 411–436.
- [42] Wagner, T., Beume, N., Naujoks, B., 2007. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization, in: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (Eds.), *EMO 2007*, Springer, Matshushima, Japan. pp. 742–756.
- [43] Zitzler, E., Laumanns, M., Thiele, L., 2002. SPEA2: Improving the Strength Pareto Evolutionary Algorithm, in: Giannakoglou, K., Tsahalis, D., Periaux, J., Papailou, P., Fogarty, T. (Eds.), *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece. pp. 95–100.
- [44] Zitzler, E., Thiele, L., 1999. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation* 3, 257–271.
- [45] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G., 2003. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation* 7, 117–132.