Adaptive Objective Space Partitioning Using Conflict Information for Many-objective Optimization

Antonio López Jaimes[1], Carlos A. Coello Coello[1], Hernán Aguirre[2], and Kiyoshi Tanaka[2]

[1] CINVESTAV-IPN, Computer Science Department, Mexico City 07360, Mexico
tonio.jaimes@gmail.com,ccoello@cs.cinvestav.mx
[2] Shinshu University, Faculty of Engineering, Nagano 380-8553, Japan
{ahernan,ktanaka}@shinshu-u.ac.jp

Abstract. In a previous work we proposed a scheme for partitioning the objective space using the conflict information of the current Pareto front approximation found by an underlying multi-objective evolutionary algorithm. Since that scheme introduced additional parameters that have to be defined by the user, in this paper we propose important modifications in order to automatically set those parameters. Such parameters control the number of solutions devoted to explore each objective subspace, and the number of generations to create a new partition. Our experimental results show that the new adaptive scheme performs as good as the non-adaptive scheme, and in some cases it outperforms the original scheme.

1 Introduction

Multi-objective Evolutionary Algorithm (MOEA) have been successfully applied to solve many real world multi-objective problems (MOPs) (e.g.,[1]). However, recent experimental [2–5] and analytical [6, 7] studies have pointed out that MOEAs based on Pareto optimality have some drawbacks to solve problems with a large number of objectives (these problems are usually called *many-objective problems*). Approaches to deal with such problems have mainly focused on the use of alternative optimality relations [8, 9], reduction of the number of objectives of the problem, either during the search process [10, 11] or, at the decision making process [12–14], and the incorporation of preference information [5].

A general scheme for partitioning the objective space in several subspaces in order to deal with many-objective problems was introduced in [15]. In that approach, the solution ranking and parent selection are independently performed in each subspace to emphasize the search within smaller regions of objective function space. Later, in [16] we proposed a new partition strategy that creates objective subspaces based on the analysis of the conflict information was proposed. By grouping objectives in terms of the conflict among them, we aim to separate the MOP into several subproblems in such a way that each subproblem contains the information to preserve as much as possible the structure of

the original problem. Since the conflict among objectives may change along the search space, the search in the subspaces should be alternated with a search in the entire objective space in order to update the conflict information. That approach is more closely related to the objective reduction approaches, specially those adopted during the search (e.g., [12, 10, 11]). However, its main difference with respect to those approaches is that the partitioning scheme incorporates all the objectives in order to cover the entire Pareto front.

Although the partitioning strategy based on conflict information produced good results compared with NSGA-II and another partitioning strategy, the strategy introduced some additional parameters which values need to be defined by the user. These parameters are *i*) the number of generations to search on the partition's subspaces and on the whole objective space, and *ii*) the number of solutions assigned to explore each subspace. This represents a problem since the best values for those parameters may depend on each problem to be solved. Additionally, some subspaces have greater degree of conflict than others. Therefore, it seems reasonable to put more emphasis on them by assigning them more resources (in our case, a larger population).

In this paper we present important modifications in order to automate the determination of those parameters and to take advantage of the conflict information to focus the search on some subspaces. In particular, the conflict information is employed to assign the number of solutions according the contribution of each subspace to the total conflict of the problem. To automatically define the parameter value described in *ii*), we use a convergence detection method based on a statistical hypothesis tests applied on the values of a quality indicator.

The remainder of the paper is organized as follows. Section 2 presents the main concepts and notation used through the paper. The basic objective space partitioning framework is introduced in Section 3. The motivation and details of the automatization of the partitioning parameters are described in Section 4. In Section 5 is presented the experimental evaluation of the adaptive scheme. Finally, in Section 6 we draw some conclusions about the new proposed scheme, as well as some future research paths.

2 Basic Concepts and Notation

Definition 1 (Objective space Φ). *The objective space of a MOP is the set $\Phi = \{f_1, f_2, \ldots, f_M\}$ of the M objective functions to be optimized.*

Definition 2 (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 3 (Space partition Ψ). *A space Φ is said to be partitioned into N_S subspaces, denoted as Ψ, if $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{N_S} | \cup_{i=1}^{N_S} \psi_i = \Phi \wedge \cap_{i=1}^{N_S} \psi_i = \emptyset\}$.*

Definition 4 (Pareto dominance relation). *A solution $\mathbf{x}^1$ is said to Pareto dominate solution $\mathbf{x}^2$ in the objective space Φ, denoted by $\mathbf{x}^1 \prec \mathbf{x}^2$, if and only if (assuming minimization): $\forall f_i \in \Phi : f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \wedge \exists f_i \in \Phi : f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$.*

Definition 5 (Pareto optimal set). *The Pareto optimal set, P_{opt}, is defined as: $P_{\text{opt}} = \{\mathbf{x} \in \mathcal{X} \mid \nexists \mathbf{y} \in \mathcal{X} : \mathbf{y} \prec \mathbf{x}\}$, where $\mathcal{X} \subseteq \mathbb{R}^n$ is the feasible set in decision variable space.*

Definition 6 (Pareto front). *For a Pareto optimal set P_{opt}, the Pareto front, PF_{opt}, is defined as: $PF_{\text{opt}} = \{\mathbf{z} = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \mid \mathbf{x} \in P_{\text{opt}}\}$. We will denote by PF_{approx} the Pareto front approximation achieved by a MOEA.*

Definition 7 (Pareto front approximation). *A Pareto front approximation, denoted by PF_{approx}, is a finite set composed of nondominated solutions.*

Of course, in practice the goal of a MOEA is finding a PF_{approx} with the best quality from all the possible approximation sets. The quality is usually defined in terms of convergence and spread [17].

3 The Conflict-Based Partitioning Framework

3.1 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 in a different subspace. By partitioning the objective space into subspaces, we aim to emphasize the search within smaller regions of objective space. 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 with individuals having a good performance in each subspace. In our approach, we partition the M-dimensional space $\Phi = \{f_1, f_2, \ldots, f_M\}$ into N_S non-overlapping subspaces $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{N_S}\}$. We selected NSGA-II to implement our proposed partitioning framework. Thus, the nondominated sorting and truncation procedures of NSGA-II are modified in the following way. The union of the parents and offspring, $\mathcal{P} \cup \mathcal{Q}$, is sorted N_S times using a different subspace each time. Then, from each mixed 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.

3.2 Using Conflict Information to Partition the Objective Space

The number of all possible ways to partition Φ into N_S subspaces is very large. Therefore, it is not feasible to search in all the possible subspaces. Instead, we can define a schedule of subspace sampling by using a partition strategy. In [15], three strategies to partition Φ were investigated: random, fixed, and shift partition. Later, we proposed a new strategy using the conflict information among objectives [11]. In that strategy the first partition contains the least conflicting objectives, the second one the next least conflicting objectives, and so on. Therefore, instead of removing the least conflicting objectives, those objectives

are integrated to form subspaces in such a way that all the objectives are optimized. By grouping objectives in terms of the conflict among them, we are trying to separate the MOP into subproblems in such a way that each subspace contains information to preserve most of the structure of the original problem. The correlation among solutions in PF_{approx} is defined to estimate the conflict among objectives in the sense defined by Carlson and Fullér [18]. 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.

In order to implement the new partition strategy we should take into account two issues: *i*) the conflict relation among the objectives may change during the search, and *ii*) the tradeoffs between objectives in different subspaces are not taken into account. In a sense, the vector evaluated algorithm (VEGA) [19] can be thought as a particular case of the partitioning framework in which each objective is a partition. The issue *ii*) is related with the "speciation" phenomena observed in VEGA in which some individuals excel in some objectives (some subspaces in our case). To deal with both issues, in the final partitioning framework the search is divided in several stages. Each of these stages is divided in two phases, namely, an approximation phase followed by a partitioning phase. In the approximation phase all the objectives are used to select the new parent population (we can view this as a partition with a single partition formed by all the objectives). The goal of this phase is finding a good approximation of the current PF_{opt}. The interested reader is referred to [16] to find the details of the entire algorithm.

4 Automatic Setting of the Partitioning Parameters

As mentioned in the previous section, the conflict-based partitioning scheme needs the definition of some important parameter values. That is, the number of generations assigned to each phase of the strategy, and the number of total stages during the search. In this section we will introduce some modifications aimed to eliminate the need to define those values manually. In addition, as we will see in this section, some subspaces have greater conflict contribution than others, therefore it seems reasonable to put more emphasis on them by assigning them more solutions. In a similar way, in some subspaces the search could stagnate earlier than in others. Therefore it could be a good idea to stop searching them and reassign those resources to other subspaces where progress can still continue.

4.1 Proportional Assignment of the Resources

One of the findings of our previous work was the fact that the conflict among certain objectives is considerably larger than the conflict among others. In order to measure the contribution of each subspace to the total conflict in the problem,

we compute for each subspace its "conflict degree", i.e., the sum of the conflict between each pair of objectives. The ratio of the conflict degree of each subspace and the total conflict is called the *conflict contribution*. In Fig. 1, we can clearly see that subspace 3 has a larger conflict contribution with respect to the other subspaces.

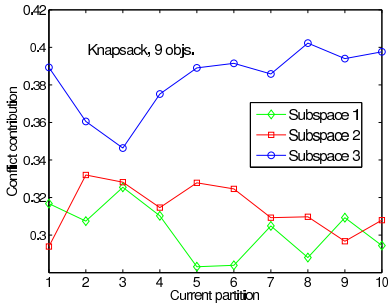

Fig. 1. Conflict contribution of each of the three subspaces generated using the conflict partition strategy.

In the original strategy proposed in [16], each subspace receives an equal number of parents regardless its conflict contribution. However, we can take advantage of the conflict information in order to distribute the number of parents accordingly. That is, the proportion of parents granted for each subspace should be proportional to its contribution to the total conflict on the problem. In order to illustrate this idea, let's take as an example the conflict contribution obtained for each of the 3 subspaces in the Knapsack problem (Fig. 2). Since subspace 3 contributes with 40% of the total contribution, then it will receive the same percentage of the total number of parents. In turn, subspaces 1 and 2 will receive 29% and 31% of the parents, respectively. The motivation behind this idea is the fact that the most conflicting objectives are the ones that change the most the Pareto optimal front [10], and, therefore, they contribute with a larger tradeoff surface to form the whole Pareto front. Thus, the most conflicting subspace should have more solutions to cover larger areas of the Pareto front. The modified procedure to carry out the non-dominated sorting taking into account the conflict contribution is presented in Algorithm 1.

In extreme cases in which some objectives do not have conflict at all with the other objectives, these objectives do not contribute at all to the structure of the Pareto front, and, therefore, can be removed. Since these objectives do not contribute to the total conflict of the problem, under the new proposed scheme, they will receive no parents. Thus, in these extreme cases, the new scheme can be regarded as a generalization of approaches that remove the least conflicting objectives during the search (see e.g., [11] and [13]).

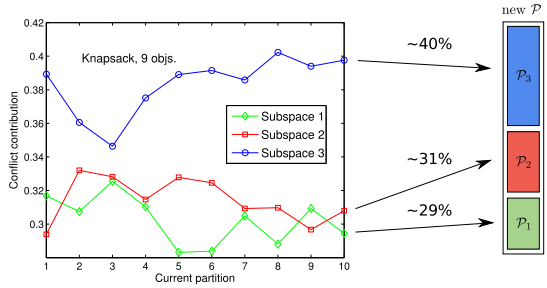

Fig. 2. Proportional assignment of parents according to the conflict contribution of the subspaces.

4.2 Automatic Transition Between Phases

In the scheme presented in [16], the number of generations in which the partitioning and integration phases are used is determined by a parameter defined by the user. However, the best value for this parameter might be different for different MOPs. In order to free the user from the task of finding the optimal parameter value we propose to determine the transition between the partitioning and integration phase automatically. In our approach we employ the progress of the current PF_{approx} in order to decide if the stage should be changed. In other words, if the search during the current phase does not have a significant progress, then it is time to change to the other phase. We adopt an approach similar to the convergence detection method proposed in [20]. That approach applies a statistical hypothesis test on a sample of quality indicators values to determine if a MOEA should be stopped. In our case, the convergence detection method will not be used to stop the search, but to switch from one phase to another.

The indicator that we propose to measure the progress of the search is based on the additive ϵ-indicator, $I_{\epsilon+}(A, B)$ [17]. This binary indicator is defined as

$$I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{\forall \mathbf{z}^2 \in B \ \exists \mathbf{z}^1 \in A : \mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2\}$$

Algorithm 1 Non-dominated sorting with proportional assignment of parents.

procedure SORT&TRUNCATION($\mathcal{R}, \mathcal{P}, \Psi$)
  $\mathcal{P}^* \leftarrow \emptyset$
  for $i \leftarrow 1$ until $|\Psi|$ **do**
    $\mathcal{F}^{\psi_i} \leftarrow$ NONDOMINATEDSORT($\mathcal{R}, \psi_i$) ▷ Sort using only the objectives in ψ_i
    CROWDING($\mathcal{F}^{\psi_i}, \psi_i$)
    $N_P \leftarrow |\mathcal{P}| \times$ conflict contribution of subspace ψ_i
    $\mathcal{P}^{\psi_i} \leftarrow$ TRUNCATION($\mathcal{F}^{\psi_i}, N_P$) ▷ Select the best N_P solutions wrt ψ_i
    $\mathcal{P}^* \leftarrow \mathcal{P}^* \cup \mathcal{P}^{\psi_i}$
  return $\mathcal{P}^*$ ▷ $|\mathcal{P}^*| = |\mathcal{P}|$

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 other words, $I_{\epsilon+}(A, B)$ is the minimum value such that aggregated to any objective vector in B, then $A \preceq B$.

The proposed indicator is intended to measure the improvement of the current approximation set PF_t with respect to the previous one, PF_{t-1}. Thus, we define this improvement as

$$\mathcal{D}_\epsilon(PF_t, PF_{t-1}) = |I_\epsilon(PF_t, PF_{t-1}) - I_\epsilon(PF_{t-1}, PF_t)|.$$

If $\mathcal{D}_\epsilon(PF_t, PF_{t-1})$ is close to zero, that implies that there is no significant improvement from iteration $t-1$ to generation t. In order to apply a hypothesis test we need to record the values of this indicator through the optimization process to get a sample of $\mathcal{D}_\epsilon$ values. The detailed process is described next. We will use two different tests, from which one is adopted to detect if the search has reached a certain threshold in terms of the $\mathcal{D}_\epsilon$ indicator. In some preliminary experiments we realized that in some cases, the MOEA diverged (specially for more than 5 objectives) making difficult to reach the desired threshold. Therefore, we also employ a hypothesis test on the slope of the regression line obtained from a sample of D_ϵ values. The first test can be formulated as follows:

$$H_0 : \mu_{\mathcal{D}_\epsilon} = \mu_0 \qquad \text{vs.} \qquad H_1 : \mu_{\mathcal{D}_\epsilon} < \mu_0$$

where μ_0 is the degree of improvement required. If H_0 is rejected in support of H_1, we can consider that the search has no significant progress, and, therefore, the phase should be changed. Since the type of distribution of the random variable $\mathcal{D}_\epsilon$ is not known, we employ a non-parametric test, namely, the Wilcoxon signed-rank test [21] to check that hypothesis.

The test for the analysis regression is formulated in the following manner:

$$H_0 : \beta_{\mathcal{D}_\epsilon} = 0 \qquad \text{vs.} \qquad H_1 : \beta_{\mathcal{D}_\epsilon} \neq 0$$

In contrast to the previous test, in order to determine if there is a significant descending linear trend of $\mathcal{D}_\epsilon$, we need to check when the hypothesis H_0 can no longer be rejected. For this test we use a two-tailed t-test [21].

Since the integration phase is only intended to obtain a new approximation of the Pareto front, and the goal of the partitioning phase is to improve the search ability when the MOEA is stuck in a local optima, we only test the linear trend during the integration phase. In addition, this helps us to determine the value of μ_0 automatically for the first test $H_0 : \mu_{\mathcal{D}_\epsilon} = \mu_0$. In contrast, both tests are used in the partitioning phase in the following manner. The test $H_0 : \beta_{\mathcal{D}_\epsilon} = 0$ is checked first, and if the null hypothesis H_0 is rejected, then the partitioning phase is maintained. If that hypothesis is not rejected, then we test $H_0 : \mu_{\mathcal{D}_\epsilon} = \mu_0$. The partitioning phase is changed only if H_0 is rejected. Written as a pseudocode, we get the following:

If $H_0 : \beta_{\mathcal{D}_\epsilon} = 0$ is not rejected (i.e., search has stalled) **and**
  $H_1 : \mu_{\mathcal{D}_\epsilon} < \mu_0$ is supported (i.e., no significant improvement) **then**
    Switch to the integration phase.
Otherwise
    Stay in the partitioning phase.

In this process, $\mu_0 = \hat{\mu}_{\mathcal{D}_\epsilon}$, that is the mean of the sample of $\mathcal{D}_\epsilon$ taken in the previous integration phase.

In a similar way to [20], a sample of $\mathcal{D}_\epsilon$ is taken from the last S_n observations of the indicator values. At the beginning of each phase the first test is carried out until S_n observations have been collected. After that, the test is checked during each subsequent generation. Additionally, in the partitioning phase, we manage the progress of each subspace independently to reallocate resources when the search in some of the subspaces has no significant improvement. Therefore, for each subspace in a partition, a sample of the last S_n observation of the D_ϵ indicator is stored. When the search in a certain subspace has stalled, then that subspace is removed from the partition, so that in the next selection of parents, only the remaining subspaces receive parents. This way all the resources are focused on subspaces with a significant progress towards the Pareto front. Fig. 3 shows an example of this procedure applied on a partition with 3 subspaces, i.e., $\Psi = \{\psi_1, \psi_2, \psi_3\}$. Initially, the subspaces receive 30%, 20% and 50% of the parents, respectively. After subspace ψ_2 has been stopped, the proportion of parents assigned to subspaces ψ_1 and ψ_3 is updated as it is shown in the figure. Thus, it is expected to have a convergence speedup in the remaining subspaces. Then, ψ_1 subspace ends, and partition ψ_3 receives all the parents. Finally, ψ_3 is stopped signaling the end of the partitioning phase and the beginning of the integration phase.

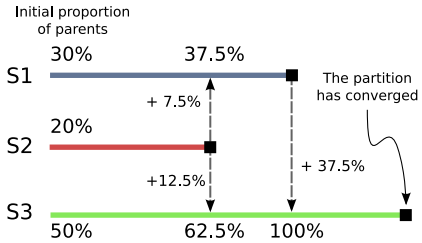

Fig. 3. Dynamic reallocation of the resources granted for each subspace.

5 Experimental Results

5.1 Algorithms, Metrics and Parameter Settings

In order to discover the advantages and disadvantages of the new adaptive partitioning scheme we compare 4 versions of the NSGA-II, i.e., the original version and three other versions using different partitioning strategies: random, conflict and adaptive-conflict. In all the algorithms we use a population of 200 individuals running during 200 generations. The results presented are the average over

30 runs of each NSGA-II variant. In the conflict-based strategy, the search is divided into 10 stages, and the values for G_Φ and G_Ψ represent the 30% and 70% of the generations of each stage, respectively. For the adaptive-conflict strategy we use a significance level $\alpha = 0.05$. The sample size for $\mathcal{D}_\epsilon$ was experimentally defined as 10.

In order to show how both conflict-based strategies work, we will use a test problem in which the conflicting objectives can be defined *a priori* by the user. Namely, the problem DTLZ5(I, M) [12], where M is the total number of objectives, and I is the number of objectives in conflict. Additionally, we employ the 0/1 Knapsack with 300 items since the conflict relation among its objectives is not known *a priori*. Unless specified otherwise, in our experiments we

Algorithm 2 Non-dominated sorting with proportional assignment of parents.

procedure CHECKCONVERGENCE($\mathcal{L}, \hat{\mu}_{\mathcal{D}_\epsilon}, \alpha$)
  for each $i \in \{1, \ldots, |\Psi|\}$ **do**
    $pSlope \leftarrow$ SLOPETEST($\mathcal{L}_i$) ▷ Test using the sample of subspace ψ_i
    $pWilcoxon \leftarrow$ WILCOXONTEST($\mathcal{L}_i, \hat{\mu}_{\mathcal{D}_\epsilon}$)
    if $pSlope > \alpha$ and $pWilcoxon \leq \alpha$ **then**
      $convergence_i \leftarrow TRUE$
    else
      $convergence_i \leftarrow FALSE$
  Return *convergence*

Algorithm 3 Adaptive Partitioning MOEA.

$\Psi \leftarrow \{\{f_1, .., f_M\}\}$ ▷ All the objectives in a single subspace.
$phase \leftarrow$ INTEGRATION
$\mathcal{L}_i \leftarrow \emptyset$ for all $i \in \{1, \ldots, |\Psi|\}$
for $t \leftarrow 1$ until $G_{\max}$ **do**
  $\mathcal{Q}_t \leftarrow$ NEWPOP($\mathcal{P}_t$) ▷ selection, crossover, mutation.
  $\mathcal{R}_t \leftarrow \mathcal{P}_t \cup \mathcal{Q}_t$
  $\mathcal{P}_{t+1} \leftarrow$ SORT&TRUNCATION($\mathcal{R}_t, \Psi$) ▷ Using current Ψ.
  for each $\psi_i \in \Psi$ **do**
    $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \mathcal{D}_\epsilon(\mathcal{P}_{t+1}, \mathcal{P}_t, \psi_i)$
  if $|\mathcal{L}_1| \geq S_N$ **then**
    CHECKCONVERGENCE($\mathcal{L}, \hat{\mu}_{\mathcal{D}_\epsilon}, \alpha$)
    if all the subspaces in Ψ have converged **then**
      if $phase =$ INTEGRATION **then**
        Compute new $\hat{\mu}_{\mathcal{D}_\epsilon}$ from $\mathcal{L}_1$
        $\Psi \leftarrow$ CREATEPARTITION($\mathcal{P}_{t+1}, \Phi, N_S$)
        $phase \leftarrow$ PARTITIONING
      else
        $\Psi \leftarrow \{\{f_1, .., f_M\}\}$
        $phase \leftarrow$ INTEGRATION
      $\mathcal{L}_i \leftarrow \emptyset$ for all $i \in \{1, \ldots, |\Psi|\}$
    else if at least one subspace have converged **then**
      Update conflict contribution in the current partition Ψ.

use from 4 to 15 objectives in each test problem. For 4-9 objectives we use 2 subspaces, and for 10-15 objectives, we use 3 subspaces. In order to assess convergence we adopt generational distance (GD). For DTZL5(I, M) we use the exact generational distance, namely $GD = \frac{1}{m}\sum_{\mathbf{z}\in PF_{\text{approx}}}\sum_{j=1}^{M}(z_j)^2 - 1$, where $m = |PF_{\text{approx}}|$. Additionally, to directly compare the convergence of the MOEAs, we utilize the additive ϵ-indicator [17]. Finally, to assess both convergence and diversity, we adopt the hypervolume indicator. For DTLZ5(I, M) the reference point was $\mathbf{z}^{\text{ref}} = 1.5^M$. For the Knapsack problem, the reference point was formed using the worst value in each objective of all the PF_{approx} generated by all the algorithms.

5.2 Application of the New Adaptive Scheme

Fig. 4 shows the progress of $\mathcal{D}_\epsilon$ in the Knapsack problem for the partitions of the integration and the partitioning phases. As can be seen, in the last two partitioning phases, the least conflicting subspace (ψ_1) converges first, whereas the most conflicting subspace (ψ_3) converges at the end. This is somewhat expected since the most conflicting subspace has a larger region of the Pareto front to cover.

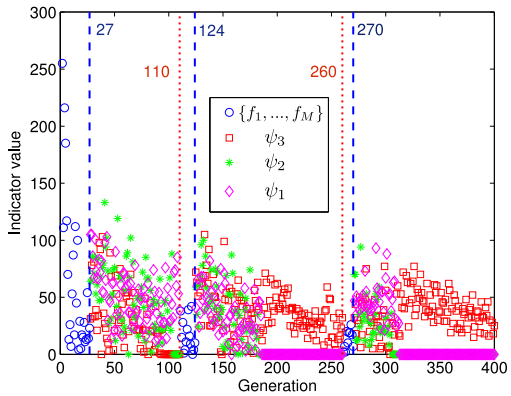

Fig. 4. Example of the convergence of the partitions $\{\{f_1, \ldots, f_M\}\}$ (integration phase) and $\{\psi_1, \psi_2, \psi_3\}$ (partitioning phase) through the search process. Dashed lines mark the end of the integration phase, while dotted lines that of the partitioning phase.

The assignment of parents for a 3-subspace partition is illustrated in Figure 5. We can observe that the most conflicting subspace receives about 50% of the parents at the beginning of each partitioning phase. Additionally, we can see that after the first subspace converges, the remaining subspaces converge

faster than the most conflicting subspace. This can be explained because the remaining subspaces receive more parents to explore their corresponding objective subspace.

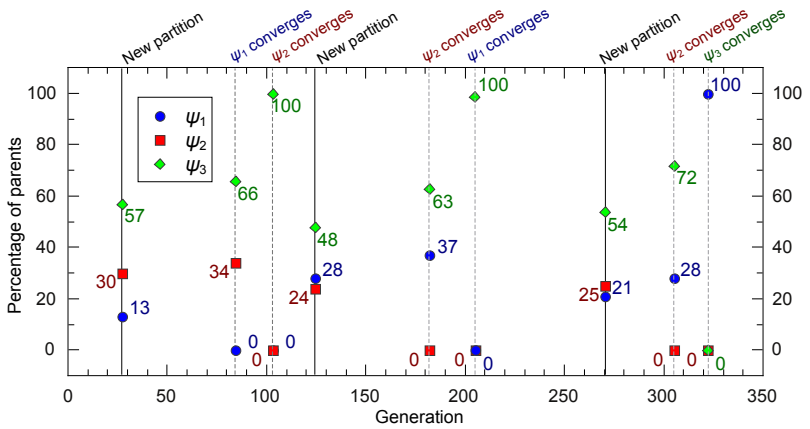

Fig. 5. Example of the proportional assignment of the parents. Solid lines indicate the generation at which a new partition is created, whereas dashed lines indicate the generation at which some subspace has converged. Next to the symbol of each subspace we indicate the number of parents assigned to the subspace.

5.3 DTLZ5(I, M): Conflict Known *a priori*

With respect to DTLZ5(I, M), the experiments show that the adaptive-conflict strategy only marginally outperforms to the conflict strategy in terms of both generational distance and hypervolume (Figs. 6 and 7, resp.). One interesting observation with respect to the hypervolume results is that for less than 5 objectives, NSGA-II performs better than any of the partitioning strategies. This implies that for those number of objectives, NSGA-II still achieves a good performance, and therefore there is no need to partitioning the objective space. Although, of course, the number of objectives in which NSGA-II performs well might depend on the given MOP. Regarding the ϵ-indicator (Fig. 8) the adaptive-conflict strategy clearly outperforms NSGA-II and the conflict strategy. $I_{\epsilon+}(A, B)$ is the subplot located in row A and column B of the matrix. The performance of both conflict-based strategies are very similar. From this experiment we can conclude about the performance of the adaptive-conflict strategy is that the proportional assignment of parents accelerates the convergence of the MOEA. Additionally, the automatic transition between phases does not seem to affect the search ability of NSGA-II.

5.4 Knapsack Problem: Unknown Conflict *a priori*

One important difference between DTLZ5(I, M) and the Knapsack problem is that none of the objectives of the Knapsack problem is totally redundant, i.e., there is conflict between every pair of objectives. However, as it was shown in [16], the objectives have different conflict degrees among them. Analyzing the boxplots of Fig. 9 we can realize that for a large number of objectives (e.g., 12-15) the adaptive-conflict strategy outperforms the original conflict strategy in terms of the ϵ-indicator 10. This could mean that as the problems get harder, the proportional assignment of parents is more useful. The adaptive-conflict strategy clearly outperforms the other variants of NSGA-II with respect to the hypervolume. This suggests that the adaptive-conflict strategy also contributes to get a better distribution of the solutions.

6 Conclusions and Future Work

In this paper we have introduced some improvements on a previously proposed scheme intended for partitioning the objective space. The main goal of these improvements is to automate the setting of some parameters of the original scheme, namely: the assignment the individuals to explore each subspace, and the transition between each phase of the scheme.

According to the experimental comparison, the new adaptive-conflict partitioning scheme performs at least as well as the two previously defined schemes. Furthermore, in some cases, the adaptive scheme outperformed those schemes. This means that, besides freeing the user from setting the above mentioned parameters, the adaptive scheme improves the search ability of the MOEA.

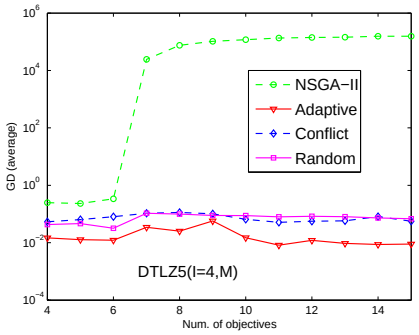

Fig. 6. GD achieved by NSGA-II and the three variants of the partitioning scheme in DTLZ5(I, M) when the number of objectives is varied from 4 to 15 objectives.

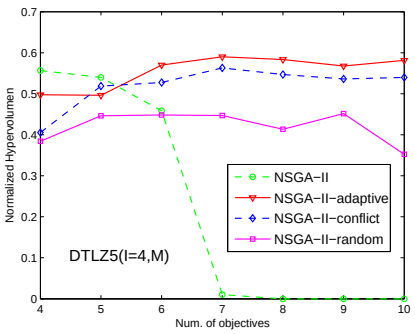

Fig. 7. Hypervolume obtained by NSGA-II and the three variants of the partitioning scheme using the problem DTLZ5(I, M).

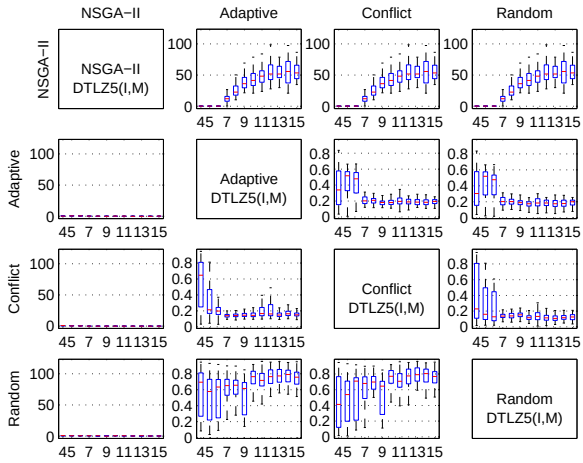

Fig. 8. Comparison of NSGA-II and the variants of the partitioning scheme with respect to I_ϵ using the problem DTLZ5(I, M).

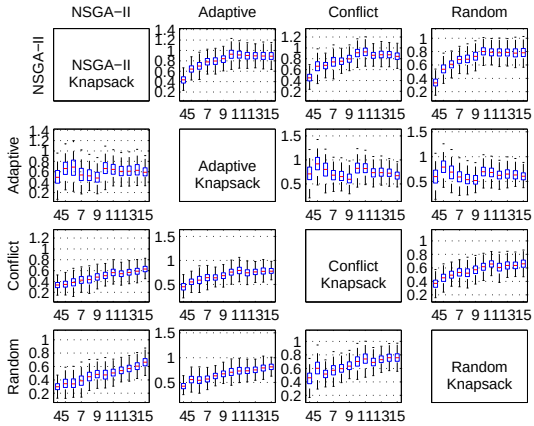

Fig. 9. Comparison of NSGA-II and the variants of the partitioning scheme with respect to I_ϵ using the Knapsack problem.

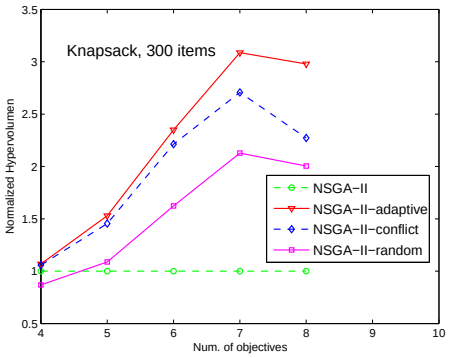

Fig. 10. Hypervolume obtained by NSGA-II and the three variants of the partitioning scheme using the Knapsack problem.

As a future work, it would be interesting to study the correlation of the transitions between phases and the number of objectives. Additionally, we plan to use the conflict information to determine automatically the size of the subspaces of the created partitions.

Acknowledgements. The first author acknowledges support from CONACyT to pursue graduate studies in Computer Science at CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 103570.

References

1. Coello Coello, C.A., Lamont, G.B.: An Introduction to Multi-Objective Evolutionary Algorithms and Their Applications. In Coello Coello, C.A., Lamont, G.B., eds.: Applications of Multi-Objective Evolutionary Algorithms. World Scientific, Singapore (2004) 1–28
2. Purshouse, R.C., Fleming, P.J.: On the Evolutionary Optimization of Many Conflicting Objectives. IEEE Transactions on Evolutionary Algorithms **11**(6) (December 2007) 770–784
3. Praditwong, K., Yao, X.: How well do multi-objective evolutionary algorithms scale to large problems. CEC 2007 (Sept. 2007) 3959–3966
4. Wagner, T., Beume, N., Naujoks, B.: Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T., eds.: EMO 2007, Matshushima, Japan, Springer. LNCS Vol. 4403 (March 2007) 742–756
5. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: CEC 2008, Hong Kong, IEEE Service Center (June 2008) 2424–2431
6. Teytaud, O.: On the hardness of offline multi-objective optimization. Evolutionary Computation **15**(4) (2007) 475–491

7. Knowles, J., Corne, D.: 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, Matshushima, Japan, Springer. LNCS Vol. 4403 (March 2007) 757–771
8. Farina, M., Amato, P.: On the Optimal Solution Definition for Many-criteria Optimization Problems. In: Proceedings of the NAFIPS-FLINT International Conference'2002, Piscataway, New Jersey, IEEE Service Center (June 2002) 233–238
9. Sato, H., Aguirre, H.E., Tanaka, K.: 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, Matshushima, Japan, Springer. Lecture Notes in Computer Science Vol. 4403 (March 2007) 5–20
10. Brockhoff, D., Zitzler, E.: Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods. In: CEC 2007, Singapore, IEEE Press (September 2007) 2086–2093
11. López Jaimes, A., Coello Coello, C.A., Urías Barrientos, J.E.: 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. Lecture Notes in Computer Science Vol. 5467, Nantes, France (April 2009) 423–437
12. Deb, K., Saxena, D.K.: Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: CEC 2006, Vancouver, BC, Canada, IEEE Press (July 2006) 3353–3360
13. Brockhoff, D., Zitzler, E.: Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In: Parallel Problem Solving from Nature IX, Springer-Verlag (2006) 533–542
14. López Jaimes, A., Coello Coello, C.A., Chakraborty, D.: Objective Reduction Using a Feature Selection Technique. In: 2008 Genetic and Evolutionary Computation Conference (GECCO'2008), Atlanta, USA, ACM Press (July 2008) 674–680
15. Aguirre, H.E., Tanaka, K.: 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. Lecture Notes in Computer Science Vol. 5467, Nantes, France (April 2009) 407–422
16. López Jaimes, A., Aguirre, H., Tanaka, K., Coello Coello, C.: Objective space partitioning using conflict information for many-objective optimization. In Schaefer, R., Cotta, C., Kolodziej, J., Rudolph, G., eds.: PPSN XI. Volume 6238 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 657–666
17. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation **7**(2) (April 2003) 117–132
18. Carlsson, C., Fullér, R.: Multiple criteria decision making: The case for interdependence. Computers and Operations Research **22**(3) (1995) 251–260
19. Schaffer, J.D.: 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 (1985) 93–100
20. Wagner, T., Trautmann, H., Naujoks, B.: OCD: Online Convergence Detection for Evolutionary Multi-Objective Algorithms Based ob Statistical Testing. In Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.K., Sevaux, M., eds.: EMO 2009. Springer. Lecture Notes in Computer Science Vol. 5467, Nantes, France (April 2009) 198–215
21. Sheskin, D.: Handbook of parametric and nonparametric statistical procedures. Chapman & Hall/CRC (2004)