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

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

[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.** Here, we present a partition strategy to generate objective subspaces based on the analysis of the conflict information obtained from the Pareto front approximation found by an underlying multi-objective evolutionary algorithm. By grouping objectives in terms of the conflict among them, we aim to separate the multi-objective 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. The ranking and parent selection is independently performed in each subspace. Our experimental results show that the proposed conflict-based partition strategy outperforms NSGA-II in all the test problems considered in this study. In problems in which the degree of conflict among the objectives is significantly different, the conflict-based strategy achieves its best performance.

## 1 Introduction

Since the first implementation of a Multi-objective Evolutionary Algorithm (MOEA) in the mid 1980s, a wide variety of approaches have been proposed, gradually improving in both their effectiveness and their efficiency to solve multi-objective problems (MOPs) [1]. However, recent experimental and analytical studies have shown that MOEAs based on Pareto optimality scale poorly when the number of objectives is increased (this is called a *many-objective problem*) [2]. Approaches to deal with such problem have mainly focused on the use of alternative optimality relations [3, 4], reduction of the number of objectives of the problem, either during the search process [5, 6] or, at the decision making process [7–9], and the incorporation of preference information [2].

A general scheme for partitioning the objective space in several subspaces in order to deal with many-objective problems was introduced in [10]. In this approach the solution ranking and parent selection are independently performed in each subspace to emphasize the search within smaller regions of objective function space. 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. 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.

Our approach is more closely related to the objective reduction approaches, specially those adopted during the search. However, its main difference with respect to them is the incorporation of all the objectives in order to cover the entire Pareto front. Deb and Saxena [7] proposed a method for reducing the number of objectives based on principal component analysis. Although some modifications can be made to this method in order to use it during the search, this method was designed as an *a posteriori* method. Brockhoff and Zitzler [5], and López Jaimes et al. [6] used similar objective reductions algorithms incorporated into a MOEA. However, in both cases, the non-conflicting objectives were discarded or aggregated to form a single objective.

## 2 Basic Concepts and Notation

**Definition 1 (Objective space $\Phi$).** *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 $\psi$).** *A subspace $\psi$ of $\Phi$ is a lower dimensional space that includes some of the objective functions in $\Phi$, i.e. $\psi \subset \Phi$.*

**Definition 3 (Space partition $\Psi$).** *A space $\Phi$ is said to be partitioned into $N_S$ subspaces, denoted as $\Psi$, 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 $\Phi$, 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, $P_{\mathrm{opt}}$, is defined as: $P_{\mathrm{opt}} = \{\mathbf{x} \in \mathcal{X} \,|\, \nexists\, \mathbf{y} \in \mathcal{X} : \mathbf{y} \prec \mathbf{x}\}$, where $\mathcal{X} \in \mathbb{R}^n$ is the variable space.*

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

**Definition 7 (Sample Correlation coefficient).** *The sample correlation coefficient, $r_{XY}$, is defined by $r_{XY} = \sum_{i=1}^{m}(X_i - \bar{X})(Y_i - \bar{Y})/(m-1)s_X s_Y$, where $s_X > 0$ and $s_Y > 0$ denote the sample standard deviations for the data sets $X$ and $Y$, respectively, and $m$ is the number of elements of each data set.*

## 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 $\Phi = \{f_1, f_2, \dots, f_M\}$ into $N_S$ non-overlapping subspaces $\Psi = \{\psi_1, \psi_2, \dots, \psi_{N_S}\}$. 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. Algorithm 1 shows this procedure.

### 3.2 A New Partition Strategy

The number of all possible ways to partition $\Phi$ 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 [10] three strategies to partition $\Phi$ were investigated: random, fixed, and shift partition. Here, we investigate a new strategy using the conflict information among objectives. Namely, the first partition would contain the least conflicting objectives, the second one the next least conflicting objectives, and so on. Therefore, instead of removing the least conflicting objectives, we integrate those objectives 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. We propose using the correlation among solutions in $PF_{\text{approx}}$ to estimate the conflict among objectives. 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 that the conflict relation among the objectives changes during the search. To deal

**Algorithm 1** Procedure of non-dominated sort and truncation.

```
procedure SORT&TRUNCATION(R, P, Ψ)
    P* ← ∅
    for i ← 1 until |Ψ| do
        F^ψi ← NONDOMINATEDSORT(R, ψi)
        CROWDING(F^ψi, ψi)
        P^ψi ← TRUNCATION(F^ψi, |P|/|Ψ|)    ▷ |P^ψi| = |P|/|Ψ|
        P* ← P* ∪ P^ψi
    return P*    ▷ |P*| = |P|
```

**Algorithm 2** Pseudocode of our proposed partitioning MOEA.

**Input:** Evolutionary operators values, $N_S$ (Num. of subspaces).
**Output:** Pareto front approximation.

```
P_0 ← randomPopulation()
evaluate(P_0)
crowding(P_0)
integrationPhase = TRUE
for t ← 1 until G_max do
    Q_t ← newPop(P_t)    ▷ selection, crossover and mutation.
    evaluate(Q_t)
    R_t ← P_t ∪ Q_t
    if integrationPhase = TRUE then
        P_{t+1} ← sort&Truncation(R_t, P_t, {Φ})
        if g ≥ G_Φ then
            integrationPhase = FALSE
            g ← 0
    else
        if g = 1 then
            Ψ ← conflictPartition(P, Φ, N_S)
        P_{t+1} ← sort&Truncation(R_t, P_t, Ψ)
        if g ≥ G_Ψ then
            integrationPhase = TRUE
            g ← 0
    g ← g + 1
```

**Algorithm 3** Partitioning Using Conflict Information.

```
procedure conflictPartition(P, Φ, N_S)
    cMatrix ← computeConflictMatrix(P)
    k ← (|Φ|/N_S) − 1
    Φ' ← Φ = {f_1, ..., f_M}
    for 1 until N_S − 1 do
        for each objective f_i in Φ' do
            V_{f_i} ← Ascending ordered list of k-nearest neighbors of f_i wrt conflict.
        V* ← V_{f_i} ∪ {f_i} : ∀f_j ∈ Φ', V_{f_i}[k] ≤ V_{f_j}[k]
        Ψ_{N_S} ← Ψ_{N_S} ∪ V*
        Φ' ← Φ' − V*.
    Ψ_{N_S} ← Ψ_{N_S} ∪ Φ'.
```

with this situation we suggest a new partitioning framework in which 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. The goal of this phase is finding a good approximation of the current $PF_{\text{opt}}$. The proposed procedure is described in Algorithm 2.

### 3.3 Partitioning Using Conflict Information

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 in the range $[0,2]$. A value of zero indicates that objectives $f_i$ and $f_j$ are not in conflict at all, and a value of 2 indicates that they are completely in conflict. The procedure to create the subspaces is presented in Algorithm 3.

## 4 Experimental Results

### 4.1 Algorithms, Metrics and Parameter Settings

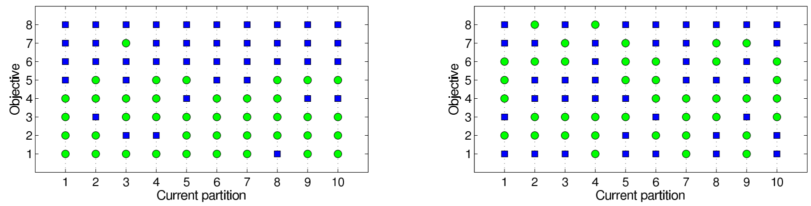
Since we wish to investigate the advantages and disadvantages of the conflict-based strategy with respect to a random strategy which creates the partitions at random, we compare the original NSGA-II with the NSGA-II using the conflict-based strategy and the random strategy. In all the algorithms we use a population of 200 individuals running during 200 generations. The results presented are the average over 30 runs of each MOEA. In the conflict-based strategy, the search is divided in 10 stages, and the values for $G_\Phi$ and $G_\Psi$ represent the 30% and 70% of the generations of each stage, respectively.

In order to show how the conflict-based strategy works, we will use a test problem in which the conflicting objectives can be defined *a priori* by the user. Namely, the problem DTLZ5$(I, M)$ [7], 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 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). In the case of 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 $m = |PF_{\text{approx}}|$. In the case of the Knapsack problem, the generational distance is computed using as our reference Pareto front, the non-dominated set resulting of the union of the $PF_{\text{approx}}$ sets obtained by the three algorithms in all the runs. Additionally, to directly compare the convergence of the MOEAs, we utilize the additive $\epsilon$-indicator [11]. In order to evaluate diversity, we adopt the inverted generational distance (IGD). 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 $PF_{\text{approx}}$ generated by all the algorithms.

### 4.2 DTLZ5$(I, M)$: Conflict Known *a priori*

In these experiments we use $I = 4$ conflicting objectives from a total of $M = 4, \ldots, 15$ objectives. For 4-9 objectives, 2 subspaces are used, whereas for 10-15 objectives, we employ 3 subspaces. First, we show that the conflict-based strategy is able to identify the conflicting objectives in most of the partitions generated during the search process. Fig. 1 shows the subspaces generated by the conflict-based and the random partition strategies during the search process. In this example, there is a total of $M = 8$ objectives. The conflicting objectives are objectives 6-8 and any other objective. The objectives in the most conflicting subspace are denoted by squares, and the other subspace is denoted by circles.

As the search progresses, the input $PF_{\text{approx}}$ used to estimate the conflict approaches the true Pareto front. Therefore, as can be seen in Fig. 1(a), in the last stages of the search, the conflict-based strategy was able to create the correct




(a) Conflict-based partition strategy.

(b) Random partition strategy.

**Fig. 1.** Subspaces generated using the conflict and random partition strategies in DTLZ5($I = 4, M = 8$). Objectives 6-8 and any other are the conflicting objectives.

partition. On the other hand, by using the random strategy (Fig. 1(b)), only one of the generated partitions 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. By inspecting the parallel coordinate plot presented in Fig. 2 we realize 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 in the middle. In contrast, the conflict-based strategy covers all the trade-offs among the objectives. In order to quantify this situation, we compute the IGD. Fig. 3 shows that the conflict-based partition strategy achieves better values in terms of IGD.

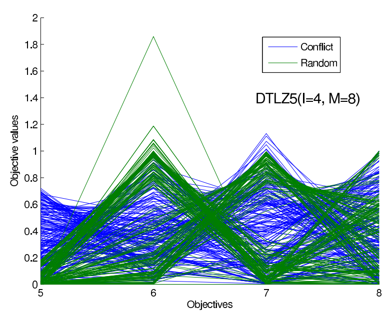



**Fig. 2.** Parallel coordinate plot of the $PF_{\text{approx}}$ obtained with the random and the conflict partition strategies.

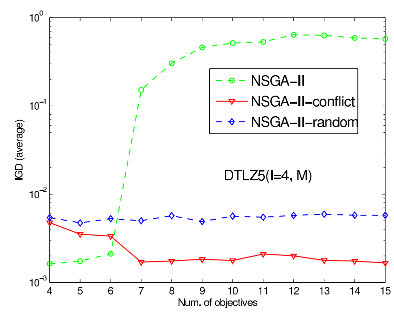



**Fig. 3.** IGD for DTLZ5(I=4,M). For 4-9 objs. we used a partition with 2 subspaces, and for 10-15 objs., one with 3.

This indicates a better distribution using the conflict-based partition strategy. In addition, the convergence of NSGA-II degrades dramatically when the number of objectives is more than 6. A possible reason of this behavior is the generation of dominance resistant solutions in DTLZ5($I, M$). In contrast, the IGD 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.

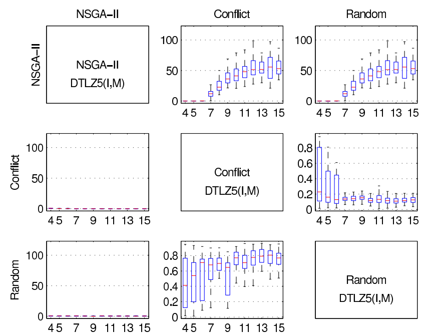



**Fig. 4.** $\epsilon$-indicator results. The horizontal axis denotes the number of objectives.

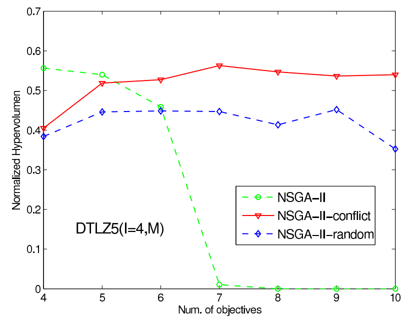



**Fig. 5.** Normalized hypervolume results on DTLZ5$(I, M)$.

The results of the $\epsilon$-indicator are presented in the matrices of subplots of Fig. 4. $I_{\epsilon+}(A, B)$ is the subplot located in row $A$ and column $B$ of the matrix. As we can see, NSGA-II is clearly outperformed by the NSGA-II using any of the partition strategies. In turn, we can observe that the conflict-based strategy is better than the random strategy, specially for 6 or more objectives. Since the hypervolume considers both convergence and distribution, as we can see in Fig. 5, the conflict-based partition strategy outperforms the random strategy. 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 causes for 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 4-6 objectives, using 2 subspaces is not possible that all the conflicting objectives are grouped in one subspace. This suggests that 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.

### 4.3 Effect of the Size of the Subspaces

In this section we analyze if it is better to have all the conflicting objectives together in a large subspace, or small subspaces in which the conflicting objectives are in different subspaces. To this end, we used DTLZ5$(I = 12, M = 24)$ to compare two partitions, namely, one with two subspaces with 12 objectives each, and another one with 6 subspaces with 4 objectives each. Fig. 6 shows the progress of GD during all the search process. We want to emphasize the fact that each partition strategy achieved a better convergence using 6 subspaces with 4 objectives. This suggests that 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, an appropriate size of the subspaces for NSGA-II would be between 4 and 6 objectives.

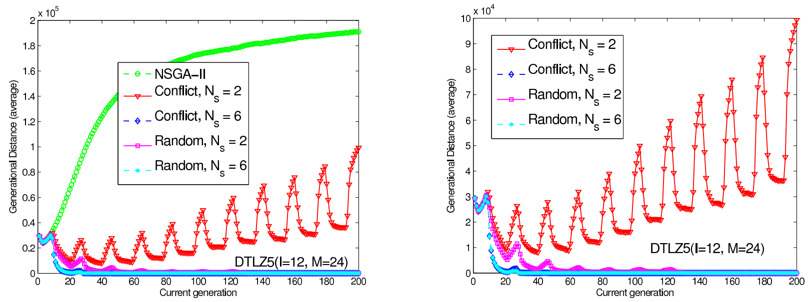



**Fig. 6.** Online GD using a partition with 2 subspaces and another one with 6 subspaces.

As in previous experiments, using parallel coordinate plots we realized that the solutions using the random strategy converge to the extremes of some objectives. To quantitatively assess the distribution, we compare the algorithms using IGD (Table 1). Although the obtained GD of the conflict and random strategies are similar using 6 subspaces (Fig. 6), the results of IGD suggest that the conflict strategy with 6 subspaces achieved a better distribution of the solutions than the random strategy with 6 subspaces.

**Table 1.** IGD values for using 2 and 6 subspaces in each of the partitioning strategies, namely, random- and conflict-based partitions.

| | **NSGA-II** | **Conflict** | | **Random** | |
|---|---|---|---|---|---|
| | | $N_S = 2$ | $N_S = 6$ | $N_S = 2$ | $N_S = 6$ |
| **Average** | 0.18005 | 0.00838 | **0.00570** | 0.00682 | 0.00769 |
| **Std. Dev.** | 0.04695 | 0.00010 | 0.00047 | 0.00092 | 0.00029 |

### 4.4 Knapsack Problem: Unknown Conflict *a priori*

In the Knapsack problem there is an interesting conflict relation among the objectives that allows the conflict-based strategy performing better than the random strategy. Fig. 7 shows the subspaces generated by the conflict strategy in the Knapsack problem. As we can see, as the search progresses, a particular partition is formed repeatedly, namely $\Psi_3 = \{\{4, 5, 8\}, \{1, 3, 9\}, \{2, 6, 7\}\}$. This suggests 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. 8, we can clearly see that subspace 3 has a larger conflict contribution with respect to the other subspaces. From the results

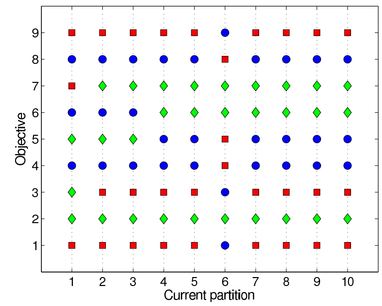



**Fig. 7.** Generated subspaces by the conflict-based partition strategy on the Knapsack problem.

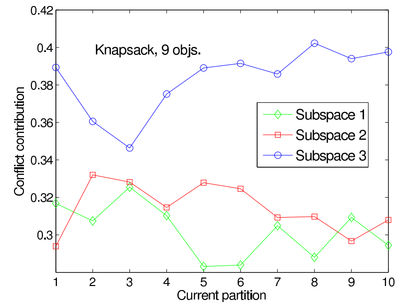



**Fig. 8.** Conflict contribution of each of the three subspaces generated using the conflict partition strategy.

obtained in GD and IGD (see Fig. 9) we can say that the conflict-based partition strategy achieved better Pareto front approximations than the random-based strategy both in terms of convergence and distribution. The results obtained with the hypervolume indicator (Fig. 10) confirm that the conflict-based strategy outperformed the random strategy. We can conclude that the differences in the degrees of conflict between each pair objectives was used by the conflict-based strategy to obtain better results than those obtained using a random partition.

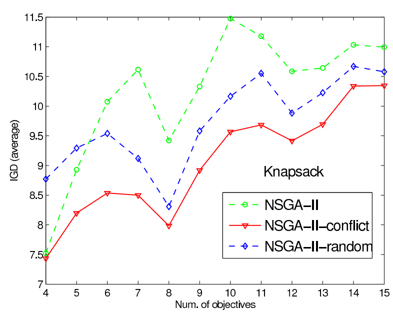



**Fig. 9.** Inverted generational distance in the Knapsack problem.

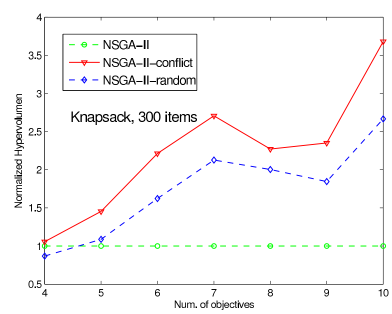



**Fig. 10.** Normalized Hypervolume wrt the one achieved by the NSGA-II.

## 5 Conclusions and Future Work

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 diverges in some test problems, the NSGA-II using any of the partition strategies maintains a good convergence despite the number of objectives. Regarding the two partition strategies, the conflict-based partition strategy achieved a better distribution of the solutions than the random strategy. In some problems, by using the random strategy, the solutions converged to the extremes of the Pareto front. In problems in which the degree of conflict among the objectives was different, the conflict-based strategy presented a better performance. It is important to note that in the Knapsack problem, where 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. Another finding is that the best size of the subspaces considerably depends on the scalability of the underlying MOEA. As part of our future work, we plan to exploit the conflict information to automatically adapt the proportion of resources granted to each subspace.

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

## References

1. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Second edn. Springer, New York (September 2007) ISBN 978-0-387-33254-3.
2. 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
3. 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
4. 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
5. 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
6. 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
7. 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
8. 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
9. 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
10. 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
11. 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