Online Objective Reduction to Deal with Many-Objective Problems

Antonio López Jaimes, Carlos A. Coello Coello*, and Jesús E. Urías Barrientos

CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco
México, D.F. 07360, México
email: tonio.jaimes@gmail.com, ccoello@cs.cinvestav.mx,
eduardo.uriasb@gmail.com

Abstract. In this paper, we propose and analyze two schemes to integrate an objective reduction technique into a multi-objective evolutionary algorithm (MOEA) in order to cope with many-objective problems. One scheme reduces periodically the number objectives during the search until the required objective subset size has been reached and, towards the end of the search, the original objective set is used again. The second approach is a more conservative scheme that alternately uses the reduced and the entire set of objectives to carry out the search. Besides improving computational efficiency by removing some objectives, the experimental results showed that both objective reduction schemes also considerably improve the convergence of a MOEA in many-objective problems.

Key words: Many-objective optimization, dimensionality reduction, objective reduction.

1 Introduction

Since the first implementation of a Multi-objective Evolutionary Algorithm (MOEA) in the mid 1980s [1], a wide variety of new MOEAs have been proposed, gradually improving in both their effectiveness and efficiency to solve multi-objective problems (MOPs) [2]. However, most of these algorithms have 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 [3, 4].

Recent experimental [5–7] and analytical [8, 9] studies have shown that MOEAs based on Pareto optimality [10] scale poorly in MOPs with a high number of objectives (4 or more). Although this limitation seems to affect only to Pareto-based MOEAs, optimization problems with a large number of objectives (also known as many-objective problems) introduce some difficulties common to any other multi-objective optimizer. Three of the most serious difficulties due to high dimensionality are the following:

* The second author is also affiliated to the UMI LAFMIA 3175 CNRS at CINVESTAV-IPN.

1. *Deterioration of the Search Ability.* One of the reasons for this problem is that the proportion of nondominated solutions (*i.e.*, equally good solutions) in a population increases rapidly with the number of objectives [11]. According to Bentley *et al.* [12] the number of nondominated k-dimensional vectors on a set of size n is $O(\ln^{k-1} n)$. As a consequence, in a many-objective problem, the selection of solutions is carried out almost at random or guided by diversity criteria. In fact, Mostaghim and Schmeck [13] have shown that a random search optimizer achieves better results than the NSGA-II [14] in a problem with 10 objectives.
2. *Dimensionality of the Pareto front.* Due to the 'curse of dimensionality', the number of points required to represent accurately a Pareto front increases exponentially with the number of objectives. Formally, the number of points necessary to represent a Pareto front with k objectives and resolution r is given by kr^{k-1} (*e.g.*, see [15]). This poses a challenge both to the data structures to efficiently manage that number of points and to the density estimators to achieve an even distribution of the solutions along the Pareto front.
3. *Visualization of the Pareto front.* Clearly, with more than three objectives 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. Parallel coordinates [16] and self-organizing maps [17] are some of the methods proposed to ease the decision making in high dimensional problems. However, more research in this field is required.

Currently, there are mainly two 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 [11], preference order ranking [18], and a method that controls the dominance area [19].
2. Reduce the number of objectives of the problem during the search process or, *a posteriori*, during the decision making process [20–22]. 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 relation[1] induced by the original objective set.

In the current paper we propose to incorporate an objective reduction method into a Pareto-based MOEA in order to cope with many-objective problems. By selecting a computationally efficient objective reduction method we can expect that the resulting MOEA improves its efficiency, since a smaller number of objective functions are evaluated. While this may be true, the omission of some objective implies some loss of information that could be important to converge to the real Pareto front. On the other hand, this omission can be useful to cope with

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

the deterioration of the search ability of Pareto-based MOEAs in many-objective problems. With this in mind we propose two schemes to integrate an efficient reduction method into a MOEA in such a way that the resulting MOEA can be useful even in problems with inexpensive objective functions. Additionally, one of the goals of this work is to investigate if an objective reduction method represents a benefit or a damage to the search ability. The results show that the proposed schemes improve the computational efficiency of a common MOEA even in problems with low computational-cost functions. More important, the experiments show that the reduction techniques employed also improve the search ability of the MOEA. Therefore, the benefit of reducing the objective set is greater than the negative effect caused by the loss of information. In [23] is also incorporated an objective reduction method into a MOEA, however the objective in that work is to improve the efficiency of hypervolume-based MOEAs which have exponential complexity in the number of objectives.

The remainder of this paper has the following structure. Section 2 presents the objective reduction technique selected to be incorporated into a MOEA. In Section 3 we describe two schemes to incorporate the reduction method during the search. The assessment of the proposed reduction schemes is presented in Section 4. Finally, in Section 5 we draw some conclusions about the proposed reduction schemes, as well as some possible paths for future research.

2 An Objective Reduction Technique Based On Correlation

The success of an objective reduction method during the search mainly depends on the balance between the overhead incurred by the reduction method itself, and the time saved by omitting some objective function evaluations. For this reason, an efficient reduction method is more likely beneficial in a wide variety of problems. In the following, we shortly describe three objective reduction methods recently proposed in the specialized literature.

Saxena and Deb [20] proposed a method for reducing the number of objectives based on principal component analysis. This method consists of an iterative scheme where the nondominated set obtained by the NSGA-II [14] is analyzed in order to gradually obtain a smaller objective set. The time complexity of each iteration[2] of this algorithm is $O(ms^2 + s^3) + O(gm^2s)$, where the second term corresponds to NSGA-II's complexity, s is the number of objectives, m is the size of the nondominated set, and g the number of generations for each run of NSGA-II.

Brockhoff and Zitzler [21] proposed two greedy algorithms to reduce the number of objectives. One of them finds a minimum objective subset that yields a given error δ (degree of change of the dominance relation). The other algorithm finds a k-sized objective subset with the minimum possible error. Both algorithms use the ϵ-dominance relation to measure the change of the domi-

[2] The total number of iterations depends on a threshold cut parameter and on the particular nondominated sets generated by the NSGA-II.

nance relation when objectives are discarded. The time complexity for these algorithms is $O(\min\{m^2s^3, m^4s^2\})$ and $O(m^2s^3)$, respectively.

Similar to Brockhoff and Zitzler, López Jaimes *et al.* [22] proposed two schemes to reduce the number of objectives. The first algorithm is intended to determine a minimum subset of objectives that yields the minimum possible error, while the second one finds a subset of objectives of a given size that yields the minimum error. 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 $O(ms^2)$.

Since López Jaimes *et al.*'s algorithms have a lower time complexity, they are suitable to be integrated into a MOEA since the chances that their computational time savings overcome their overhead are larger than those of the other methods described here. However, in this study we have only chosen the algorithm that finds a subset of objectives of a given size.

2.1 Details of the Selected Objective Reductions Method

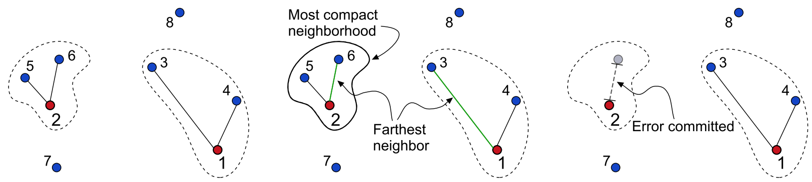
The algorithm that finds a k-sized objective subset (KOSSA) uses a correlation matrix to estimate the conflict between each pair of objectives. This matrix is computed using the nondominated set generated by some MOEA. A negative correlation between a pair of objectives means that one objective increases while the other decreases and vice versa. This way, we could interpret that the more negative the correlation between two objectives, the more conflict between them.

Since the interest is in the negative correlation, we use $1 - \rho(f_1, f_2) \in [0, 2]$ to measure the degree of negative correlation (where $\rho(f_1, f_2)$ is the correlation between objectives f_1 and f_2). Thus, a value of 2 indicates that objectives f_1 and f_2 are completely negatively correlated (totally in conflict) and a result of zero indicates that the objectives are completely positively correlated (without any conflict).

The central part of the objective reduction algorithm is divided in three steps:

1. Divide the objective set into homogeneous neighborhoods of size q around each objective. The conflict between objectives takes the role of the distance. That is, the more conflict between two objectives, the more distant they are in the "conflict" space. Figure 1(a) shows only two neighborhoods of a hypothetical situation with eight objectives and $q = 2$.
2. Select the most compact neighborhood. That is, the neighborhood with the minimum distance to its q-th nearest-neighbor. Figure 1(b) shows the farthest neighbor for each of the two neighborhoods. In the example, the neighborhood on the left is the most compact one.
3. Retain the center of that neighborhood and discard its q neighbors. In this process, the distance to the q-th neighbor can be thought of as the error committed by removing the q objectives (see Figure 1(c)).

The pseudocode of the reduction algorithm, KOSSA, is presented in Figure 2. In this pseudocode each entry, $r_{i,j}$, of the correlation matrix represents the conflict between objective f_i and f_j. In particular, $r_{i,q}$ denotes the conflict between objective f_i and its q-th nearest-neighbor.



(a) Divide the objective set into neighborhoods around each objective.

(b) Select the most compact neighborhood.

(c) Retain the center and remove the neighbors.

Fig. 1. Basic strategy of the objective reduction method employed.

3 Integration Schemes of the Objective Reduction Method into a MOEA

When some objectives are discarded from the original problem some information is being lost. The magnitude of this loss depends on the degree of redundancy among the objectives.

In any case, we have to balance the benefit of discarding some objectives along with the computational cost of the reduction algorithm. Two benefits are clear from removing some objectives, namely: *i*) the avoidance of the computation of some possible computational expensive objective functions, and *ii*) the speedup

Input: Nondominated set A.
Initial objective set $F = \{f_i, i = 1, ..., s\}$.
Number of neighbors $q \leq |F| - k$.
Size of the desired objective subset, k.
Step 0: Compute the correlation matrix using A.
Step 1: $F' \leftarrow F$.
Step 2: Find objective f_i^{min} which corresponds to
$r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$.
Step 3: Retain f_i^{min} and discard its q neighbors from F'.
Let $error \leftarrow r_{i,q}^{min}$.
Step 4: **If** $q > |F'| - k$ **then** $q \leftarrow |F'| - k$.
Step 5: **If** $|F'| = k$ **then** go to **Step 8** to stop.
Compute again $r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$.
Step 6: **While** $r_{i,q}^{min} > error$ and $q > 1$ **do**:
$q \leftarrow q - 1$.
$r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$.
Step 7: Go to **Step 2**.
Step 8: Return set F' as the reduced objective set.

Fig. 2. Pseudocode of the objective reduction algorithm KOSSA.

in execution of the MOEA, specially if its complexity time largely depends on the number of objectives.

Next, we will describe two schmes to incorporate the KOSSA method into a MOEA. First, we propose a simple scheme where the objective set is reduced successively during most of the search and only towards the end of the search all the objectives are integrated. This scheme is divided in three stages:

1. In the first stage the MOEA is executed for a number of generations using all the objectives. The MOEA obtains an initial approximation of the Pareto front which will be the first input of the objective reduction method, KOSSA.
2. The second stage is the main stage of the scheme where the objective set is gradually reduced through several generations. In this stage, every certain number of generations KOSSA is executed to reduce the objective set and then the execution of the MOEA is resumed. This process is repeated until the desired objective set size has been reached.
3. In the last stage all the objectives are taken up again to obtain the final approximation of the Pareto front.

The detailed scheme with successive reductions is described in Algorithm 1, where P denotes the best population obtained so far by the MOEA.

Algorithm 1 Pseudocode of the successive reduction scheme.

Input:
R: Number of reductions during the search.
k: Size of the minimum objective set allowed.
G_{max}: Total number of generations.
G_{pre}: Generations before the reduction stage.
G_{post}: Generations after the reduction stage.

$G \leftarrow G_{pre}$; $F' \leftarrow F$
$k' \leftarrow \lceil (|F| - k)/R \rceil$ ▷ Number of objectives discarded per reduction.
for $r \leftarrow 1$ until $R + 2$ **do**
  for $g \leftarrow 1$ until G **do**
    MOEA(P, F')
  if $r \neq R + 2$ **then**
    ▷ Reduce the current objective set F'.
    if $r \leq R$ **then**
      $F' \leftarrow$ KOSSA$(P, F', |F'| - k')$
      $G \leftarrow (G_{max} - G_{pre} - G_{post})/R$
    else
      ▷ Integrate all the objectives at the end of the search.
      $F' \leftarrow F$
      $G \leftarrow G_{post}$

In the current implementation of this scheme we decided to schedule the reduction phases equally distributed during the reduction stage. However, other schedules are possible, for instance the number of generations for the next reduction can be shortened each time, since the population converges faster after each reduction. A similar decision can be made with regard to the number of objectives discarded on each reduction. Currently, the same number of objectives is removed at each reduction as it can be seen in the third statement of Algorithm 1.

Although this scheme has the advantage (computationally speaking) of omitting the evaluation of many objectives during most of the search, it is possible that the loss of information diminishes the MOEA's convergence ability. Therefore, we also proposed a less aggressive scheme which integrates the entire objective set periodically during the search to counterbalance the loss of information. As in the scheme described previously, this mixed scheme starts the search using the whole objective set for some generations. However, it alternates the reduction process with the integration of the original objectives during the remainder of the search. Algorithm 2 presents the details of the mixed scheme.

Algorithm 2 Pseudocode of the mixed reduction scheme.

Input:

R: Number of reductions during the search.
k: Size of the minimum objective set allowed.
G_{max}: Total number of generations.
G_{pre}: Generations before the reduction stage.
p_{red}: Percentage of generations using the reduced objective set.

```
p_int ← 1 − p_red.
G_red ← p_red × (G_max − G_pre)/R
G_int ← p_int × (G_max − G_pre)/R
G ← G_pre
F' ← F
k' ← ⌈(|F| − k)/R⌉   ▷ Number of objectives discarded per reduction.

for r ← 1 until 2R + 1 do
    for g ← 1 until G do
        MOEA(P, F')

    if r ≠ 2R + 1 then
        ▷ Reduce the current objective set F'.
        if r mod 2 = 1 then
            F' ← KOSSA(P, F', |F'| − k')
            G ← G_red
        else
            ▷ Integrate all the objectives for the next generations.
            F' ← F
            G ← G_int
```

4 Assessment of the Objective Reduction Schemes Coupled with a MOEA

In order to evaluate the performance of the schemes presented in the previous section we chose the NSGA-II as a testbed. As we mention in previous sections, the worth of using an objective reduction method depends on its computational cost, the time complexity of the MOEA (specially if it depends on the number of objectives), the computational cost of the objective functions, and on the effect caused by the removal of objectives.

In order to investigate the effect of these factors, we carried out two types of experiments. The first group of experiments attempts to provide an overall assessment of all those factors in order to determine if the reduction method is advantageous. To do so, instead of using the number of evaluations as a stopping criterion, we use the real computational time instead. By doing so, we can decide if the overall benefits of the reduction method are greater than its possible damages. In the second group of experiments we want to investigate if a reduction method increases or decreases the number of generations required to reach a certain quality of the approximation set produced.

In both types of experiments we compare the NSGA-II equipped with the reduction method (REDGA) against the original NSGA-II. The following problems were adopted in all the experiments: the 0/1 multi-objective knapsack problem with 200 items, and a variation, proposed in [24], of the well-known problem DTLZ2 (denoted here by DTLZ2$_{BZ}$) with 30 variables. All the runs were executed in a single-core computer with a 2.13 GHz CPU.

In the first group of experiments the results were evaluated using the additive ϵ-Indicator [25], which is defined as

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

for two nondominated sets A and B, where $\boldsymbol{z}^1 \succeq_{\epsilon+} \boldsymbol{z}^2$ 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 $A \succeq B$. 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.

4.1 Overall Assessment of the Reduction Schemes

In these experiments we used four instances for each of the two test problems employed with 4, 6, 8 and 10 objectives. For each number of objectives we fixed the following time windows: 2, 4, 6 and 10 seconds. For all the 30 runs and problems we used a population of 300 individuals. For NSGA-II we employed a crossover probability of 0.9 and a mutation probability of $1/N$ (N is the number of variables). In the knapsack problem we used a binary representation with a mutation probability of $1/n$ (n is the length of the chromosome).

In order to study the successive reduction scheme we reduced in all cases the objective set until a size of $k = 3$ and the percentage of generations before and after the reduction stage was fixed to 20% and 5%, respectively. Here, we studied two scenarios: one that reduces all the required objectives in one reduction (REDGA-S-1), while the other one uses, among all possible number of reductions, an intermediate number of reductions considering a final set of size $k = 3$ (REDGA-S-m). That is, for 6, 8 and 10 objectives were used 2, 3, and 4 reductions, respectively. In the mixed reduction scheme we only used an intermediate number of reductions for every number of objectives (REDGA-X-m), and the other parameters were $k = 3$, $p_{red} = 0.85$ and 20% of the total generations were accomplished before the reduction stage. The results of the ϵ-Indicator for these scenarios on problem DTLZ2$_{BZ}$ are presented in Table 1. Since for four objectives REDGA-S-m and REDGA-X-m are equivalent to the REDGA-S-1 scheme, we only show the results of this scheme against NSGA-II.

Table 1. Results of the reduction schemes with respect to the ϵ-Indicator in the DTLZ2$_{BZ}$ problem using a fixed-time stopping criterion.

DTLZ2$_{BZ}$ with 4 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	-	-	0.04450	**0.04450**
REDGA-S-m	-	-	-	-	
REDGA-X-m	-	-	-	-	
NSGA-II	0.06469	-	-	-	0.06469
Average	**0.06469**			0.04450	
DTLZ2$_{BZ}$ with 6 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	0.05961	0.05723	0.06019	0.05901
REDGA-S-m	0.05259	-	0.05085	0.05849	**0.05398**
REDGA-X-m	0.05850	0.05614	-	0.05421	0.05628
NSGA-II	0.07447	0.07711	0.07972	-	0.07710
Average	0.06185	**0.06429**	0.06260	0.05763	
DTLZ2$_{BZ}$ with 8 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	0.08583	0.07711	0.07179	0.07824
REDGA-S-m	0.06905	-	0.08195	0.06341	**0.07147**
REDGA-X-m	0.07386	0.08171	-	0.06944	0.07500
NSGA-II	0.09882	0.10616	0.11782	-	0.10760
Average	0.08058	0.09123	**0.09229**	0.06821	
DTLZ2$_{BZ}$ with 10 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	0.09108	0.09182	0.08316	0.08869
REDGA-S-m	0.06916	-	0.07072	0.07926	**0.07305**
REDGA-X-m	0.07998	0.08554	-	0.06840	0.07797
NSGA-II	0.11608	0.12159	0.11480	-	0.11749
Average	0.08841	**0.09940**	0.09245	0.07694	

As we can clearly see in Table 1, all the reduction schemes perform better than NSGA-II for every number of objectives. Besides, the advantage of the reduction schemes over the NSGA-II increases with the number of objectives. On the other hand, except for 8 objectives, the scheme REDGA-S-m achieved better results than the REDGA-X-m which is the second best in this comparison. This means that the strategy of integrating all the objectives periodically did not improve the performance of the reduction scheme. As somewhat expected, the REDGA-S-1 scheme did not obtain results as good as the other reduction schemes. A possible explanation is that, in spite of the fact that REDGA-S-1 carries out more evaluations than the other schemes in the given time, this advantage is not enough to counteract the negative effect caused by the loss of information. In this sense, the REDGA-S-m scenario represents a better tradeoff between these factors.

Table 2. Results of the reduction schemes with respect to the ϵ-indicator in the 0/1 Knapsack problem using a fixed-time stopping criterion.

Knapsack with 4 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	-	-	205	**205**
REDGA-S-m	-	-	-	-	
REDGA-X-m	-	-	-	-	
NSGA-II	241	-	-	-	241
Average	**241**			205	
Knapsack with 6 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	408	264	318	**330.0**
REDGA-S-m	371	-	269	352	330.7
REDGA-X-m	372	403	-	306	360.3
NSGA-II	448	414	378	-	413.3
Average	397.0	**408.3**	303.7	325.3	
Knapsack with 8 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	646	478	505	543.0
REDGA-S-m	457	-	323	290	**356.7**
REDGA-X-m	441	465	-	345	417.0
NSGA-II	564	472	438	-	491.3
Average	487.3	**527.7**	413.0	380.0	
Knapsack with 10 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	455	424	423	434.0
REDGA-S-m	503	-	411	376	**430.0**
REDGA-X-m	760	667	-	493	640.0
NSGA-II	533	455	522	-	503.3
Average	**598.7**	525.7	452.3	430.7	

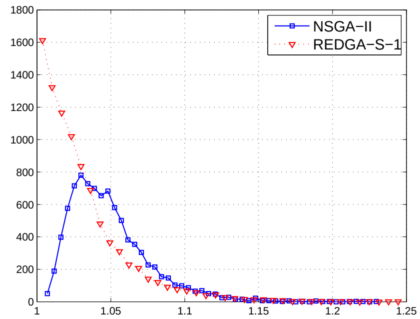
As in the previous problem, NSGA-II was the worst algorithm in the 0/1 knapsack problem regarding the ϵ-Indicator (see Table 2). Nonetheless, the REDGA-S-1 scheme presented a better performance than in DTLZ2$_{BZ}$, *i.e.*, with 4 objectives it was the second best and with 10 it was the best scheme. The reason is that knapsack's objective functions are more computationally expensive than those of the problem DTLZ2$_{BZ}$. This allowed that REDGA-S-1 could perform many more generations than any other scheme. This is a clear example that the balance between the computational cost of the objective functions and the overhead of the reduction scheme plays an important role on the success of the reduction scheme. Furthermore, it acts as a guide to decide what type of reduction scheme to choose. If the objective functions are expensive then it may be convenient to use an aggressive scheme such as REDGA-S-1; otherwise, the REDGA-S-m could be more appropriate.

4.2 Effect of the Reduction Schemes on MOEA's Search Ability

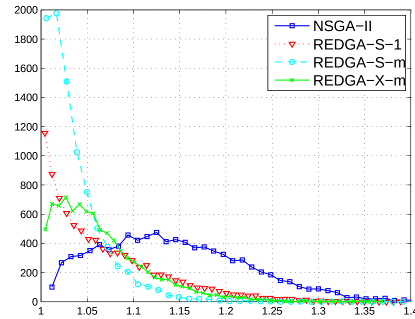
In order to investigate how a reduction scheme affects the MOEA's convergence ability we compare the reduction schemes using the number of generations as the stopping criterion. In these experiments we used a population of 300 individuals for every number of objectives, and all the algorithms were executed for 200 generations (60 000 evaluations). In this experiment we adopt only DTLZ2$_{BZ}$ since convergence can be easily measured given that the nondominated vectors of its true Pareto front have the property $D = \sum_{i=1}^{s} f_i^2 = 1$, where s is the number of objectives. The distribution of the values of D for each algorithm are shown in Figure 3. The horizontal axis represents the D values obtained by each algorithm and the vertical axis denotes the frequency of a given D value. As well as in other studies [13, 6], Figure 3 shows that the performance of NSGA-II decays as the number of objectives increases. In addition, all the reduction schemes perform better than NSGA-II in all cases. This means that the reduction schemes, besides reducing execution time also help Pareto-based MOEAs to recover the search ability deteriorated by the inability of Pareto optimality to discriminate solutions in many-objective problems. In concordance with the fixed-time experiments, the REDGA-S-m achieves the best convergence with respect to the average D value presented in Table 3. Like all the algorithms, its convergence decreases with the number of objectives. However REDGA-S-m is the scheme less affected by the number of objectives.

5 Conclusions and Future Work

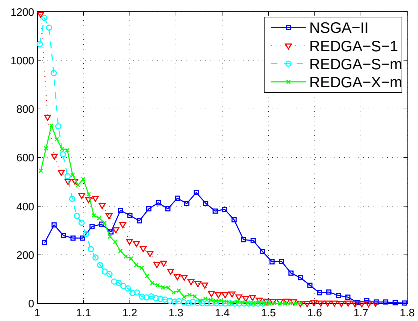
In this paper, we have presented two schemes to integrate an objective reduction method into a MOEA. One of these schemes reduces successively the number of objectives until the required size has been reached and only at the final generations the original objective set is used again (REDGA-S). The second scheme is intended to counterbalance the negative effect of the loss of information by



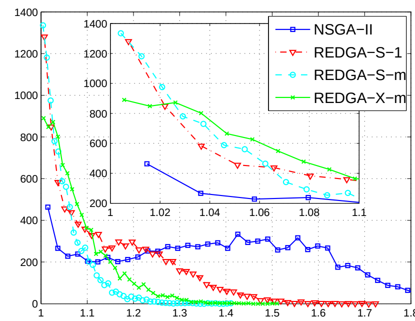
(a) 4 objectives



(b) 6 objectives



(c) 8 objectives



(d) 10 objectives

Fig. 3. D distribution on the problem DTLZ2_{BZ} for different number of objectives. $D = 1$ corresponds to the true Pareto front.

Table 3. Results of the reduction schemes with respect to the value D in the DTLZ2_{BZ} problem using a fixed-generations stopping criterion.

Obj		REDGA-S-1	REDGA-S-m	REDGA-X-m	NSGA-II
4	**Average**	**1.0305**	-	-	1.0488
	Std. Dev.	0.0289	-	-	**0.0266**
6	**Average**	1.0672	**1.0358**	1.0649	1.1445
	Std. Dev.	0.0609	**0.0334**	0.0496	0.0799
8	**Average**	1.1276	**1.0607**	1.1040	1.2863
	Std. Dev.	0.1113	**0.0561**	0.0805	0.1559
10	**Average**	1.1402	**1.0501**	1.0786	1.3787
	Std. Dev.	0.1234	**0.0487**	0.0690	0.2218

omitting some objectives (REDGA-X). This scheme uses alternately the reduced and the entire set of objectives to carry out the search.

The first group of experiments based on a fixed-time stopping criterion showed that the reduction of objectives during the search is beneficial in spite of the loss of information since it also saves computational time. This means that the overhead introduced by the objective reduction method was small enough to speed up the execution of the MOEA even with the inexpensive objective functions used in the study. Although in all the cases studied in the first group of experiments the MOEA coupled with the reduction scheme achieved better results than the MOEA alone, we have to carefully select the parameters of the reduction scheme. There is an equilibrium point in the number of objectives that need to be removed in order to achieve the best tradeoff possible between the benefits and damages obtained by the reduction scheme. To illustrate this, it is sufficient to consider that, although the REDGA-S scheme with only one reduction is the one that saves more time per generation, it did not present as good performance as a less aggressive configuration such as the REDGA-S-m. On the other hand, the periodic incorporation of the entire objective set did not improve the performance of the successive reduction scheme, which is simpler.

One important finding is that a reduction scheme besides reducing the execution time of a MOEA also helps to remedy the limitation of Pareto optimality for dealing with problems having a large number of objectives. The results showed that all the reduction schemes studied outperformed the original MOEA even when a stopping criterion based on a fixed number of generations was used. This shreds light into the usefulness of objective reduction schemes since they bring advantages both in efficiency and effectiveness.

As part of our future work, we want to study the performance of the objective reduction methods in problems with less conflict among their objectives. We would expect that in those problems the benefit of using a reduction method would be greater since the loss of information is smaller than in many-objective conflicting problems. Given their encouraging results, it would be interesting to compare the reduction schemes proposed against methods that have shown good performance in many-objective problems.

Acknowledgments.

The first author acknowledges support from CONACyT and from CINVESTAV-IPN to pursue graduate studies in computer science.

References

1. 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
2. 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.

3. Fonseca, C.M., Fleming, P.J.: 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**(1) (1998) 38–47
4. Hughes, E.J.: Radar Waveform Optimisation as a Many-Objective Application Benchmark. In Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T., eds.: Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matshushima, Japan, Springer. Lecture Notes in Computer Science Vol. 4403 (March 2007) 700–714
5. Hughes, E.J.: Evolutionary Many-Objective Optimisation: Many Once or One Many? In: 2005 IEEE Congress on Evolutionary Computation (CEC'2005). Volume 1., Edinburgh, Scotland, IEEE Service Center (September 2005) 222–227
6. 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.: Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matshushima, Japan, Springer. Lecture Notes in Computer Science Vol. 4403 (March 2007) 742–756
7. Praditwong, K., Yao, X.: How well do multi-objective evolutionary algorithms scale to large problems. Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (Sept. 2007) 3959–3966
8. Teytaud, O.: How entropy-theorems can show that approximating high-dim pareto-fronts is too hard. In: Bridging the Gap between Theory and Practice - Workshop PPSN-BTP, International Conference on Parallel Problem Solving from NaturePPSN. (2006)
9. 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.: Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matshushima, Japan, Springer. Lecture Notes in Computer Science Vol. 4403 (March 2007) 757–771
10. Pareto, V.: Cours D'Economie Politique. F. Rouge (1896)
11. 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
12. Bentley, J.L., Kung, H.T., Schkolnick, M., Thompson, C.D.: On the average number of maxima in a set of vectors and applications. J. ACM **25**(4) (1978) 536–543
13. Mostaghim, S., Schmeck, H.: Distance based ranking in many-objective particle swarm optimization. In Rudolph, G., Jansen, T., Lucas, S.M., Poloni, C., Beume, N., eds.: PPSN. Volume 5199 of Lecture Notes in Computer Science., Springer (2008) 753–762
14. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: 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, Paris, France, Springer. Lecture Notes in Computer Science No. 1917 (2000) 849–858
15. Sen, P., Yang, J.: Multiple criteria decision support in engineering design. Springer, London, UK (1998) ISBN 3-540-19932-2.
16. Wegman, E.J.: Hyperdimensional data analysis using parallel coordinates. Journal of the American Statistical Association **85** (1990) 664–675
17. Obayashi, S., Sasaki, D.: Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map. In Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele,

L., eds.: Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, Faro, Portugal, Springer. Lecture Notes in Computer Science. Volume 2632 (April 2003) 796–809
18. di Pierro, F.: Many-Objective Evolutionary Algorithms and Applications to Water Resources Engineering. PhD thesis, School of Engineering, Computer Science and Mathematics, UK (August 2006)
19. 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.: Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matshushima, Japan, Springer. Lecture Notes in Computer Science Vol. 4403 (March 2007) 5–20
20. Deb, K., Saxena, D.K.: Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, IEEE (July 2006) 3353–3360
21. 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
22. 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 ISBN 978-1-60558-131-6.
23. Brockhoff, D., Zitzler, E.: Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods. In: 2007 IEEE Congress on Evolutionary Computation (CEC'2007), Singapore, IEEE Press (September 2007) 2086–2093
24. Brockhoff, D., Zitzler, E.: Offline and Online Objective Reduction in Evolutionary Multiobjective Optimization Based on Objective Conflicts. TIK Report 269, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich (April 2007)
25. 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