
Crossing over Evolutionary Algorithms: a clustered ϵ -NSGAI

Eloisa Bentivegna
Department of Physics
Penn State University
University Park, PA 16802

Abstract

The current landscape of second generation Multi-Objective Evolutionary Algorithms (MOEAs) counts numerous successful approaches, each exhibiting its own distinctive style. This class of algorithms represents a Pareto front in the most canonical sense: no algorithm outperforms all the others under all evaluation criteria, and each is advantageous over the others for a certain class of test problems. A legitimate question is then whether it is possible to generate other members of this Pareto front via the crossover of successful traits from different algorithms. A step in this direction is already represented by ϵ -NSGAI, which implements ϵ -MOEA's (see Deb et al. (2003)) archiving (*ϵ -archiving*) within the framework of the standard NSGAI. In this paper, we present the result of the implementation of SPEA's and MIDEA's clustering reduction in lieu of the crowded tournament selection of ϵ -NSGAI. We test the new algorithm over a set of optimization problems that are particularly challenging to ϵ -NSGAI.

1 OVERVIEW OF MOEAs

The field of multiobjective optimization via Evolutionary Algorithms (EAs) presents today a vast spectrum of approaches and methodologies.

The mere existence of conflicting objectives calls for a redefinition of the concept of optimization, which can no longer be defined as the extremization of a single function, but should rather be viewed as the search for the class of solutions that are optimal tradeoffs for a given problem, i.e. represent the best result of the

optimization of one of the objectives for a given value of all the others. This is the classical notion of Pareto dominance (see, e.g., Deb (2001)), and the class of such solutions is usually referred to as the Pareto front.

The lack of a unique fitness estimate for the solutions carries over to the algorithms themselves, whose performance is represented by several -often conflicting- factors. In particular, two ingredients have been singled out as the most relevant indicators of an algorithm's proficiency in reproducing a Pareto front: the *proximity* of the algorithm's final approximation set to the actual Pareto front, and the *diversity* of such set, represented by the fraction of the Pareto front that is covered by the set (see Bosman & Thierens (2003)). In the past ten years, several different approaches to ensure convergence and spread have been employed, and each algorithm represents a different balance in the achievement of these two goals.

The task of quantifying each particular tradeoff involves, as a first step, the choice of some performance metrics. If the Pareto front is known, two traditional indicators can be used:

1. The *convergence* of each algorithm can be estimated by calculating the average Euclidean distance of the points in the approximation set to the points in the Pareto front.
2. The *diversity* can be similarly evaluated by calculating the fraction of the Pareto front that is covered by the algorithm's approximation set.

A third indicator, usually referred as ϵ -performance, gives a flavor of both the convergence and the diversity of an approximation set. In a nutshell, for each point in the Pareto front a bounding box of dimension $\epsilon_1 \cdot \epsilon_2 \cdots \epsilon_M$, where M is the number of objectives, is built. The ϵ -performance indicator is then represented by the fraction of such boxes that contain at least one point

from the approximation set.

The experiments presented in this paper intend to explore how the convergence and diversity properties of ϵ -NSGAII are modified when the SPEA-like and MIDEA-like elitism replace the current type of environmental selection. We are going to give a more in-depth presentation of the types of environmental selection in the next section, and we will present the detail of the replacement, along with the results of our tests, in section 3.

2 ELITISM: CROWDED TOURNAMENT VS. CLUSTERING

Drawing from the lesson of single objective optimization, elitism (the inclusion of the best individuals in the parent population right on to the child population) is applied to enhance the convergence properties of the algorithms. Since now it is the convergence to a (hyper)surface, rather than to a single point, that has to be achieved, the selection of the class of individuals to carry over from one generation to the next is based not only on its proximity to the surface, but also on the amount of spread in the class, i.e. on how representative the class is of the entire tradeoff spectrum. The means by which such a class is obtained are referred to as *environmental selection*, as opposed to *mating selection*, the choice of the individuals that will partake in the reproductive step. While a clear population ranking can be established in single objective problems, for MOEAs we are again faced with the concept of non-domination. If two solutions are such that neither dominates the other one, there is no univocal way of establishing whether one would be preferable over the other, and a different criterion needs to be introduced in order to determine which solution to retain and which one to discard (unless, of course, both can be retained).

In the two following subsections, we will analyze the elitist approach of three current MOEAs, namely the ϵ -dominance Non-dominated Sorting Genetic Algorithm II (ϵ -NSGAII), the Strength Pareto Evolutionary Algorithm (SPEA) and the Mixture-based multi-objective Iterated Density-estimation Evolutionary Algorithm (MIDEA).

Two strategies to replace ϵ -NSGAII's elitism approach with SPEA's and MIDEA's paradigm will be illustrated in section 3, along with their implementation and the corresponding results.

2.1 Elitism in ϵ -NSGAII

Elitism in ϵ -NSGAII takes place through crowded tournament selection. Starting with a population of size N , the (mating) selection and mutation operators are applied to generate N children. The cumulative $2N$ individuals are ranked based on their non-domination sorting; following, the first front is copied into the new population, up to size N . If any space is left, the second front is copied, and so on, until size N is reached. If a front is too big to be included in the new population in its entirety, it will be reduced to size N by means of *crowding distance* ranking. First, all individuals in the front are ranked according to the value of the first objective. Ties are resolved by resorting to the second objective, and so on. Once the ordering is completed, a crowding distance is assigned to each individual equal to the sum of the objective differences of its two nearest neighbors, in each direction. The front is then ordered according to the crowding distance, the individuals with the larger values being selected for inclusion in the new population. Summarizing this procedure in a pseudocode:

- For each generation:
 - Get population of size N from previous generation;
 - Selection, mating and mutation generate N extra individuals;
 - The $2N$ individuals are sorted into fronts according to their nondomination;
 - Set $a = 0$. For each nondominated front (starting with the first one):
 - * If $k < N - a$ include all k individuals in new population;
 - * Else rank individuals according to their crowding distance, include the first $N - a$ individuals in the new population and break;
 - * $a += k$;
 - Copy the new population into the old population and move on to next generation.

2.2 Elitism in SPEA

The SPEA algorithm implements elitism through an external population, or archive \bar{P}_t of a predefined size \bar{N} . The algorithm is initialized with a random population P_0 , whose first non-dominated front is copied into \bar{P}_0 . If the front size exceeds \bar{N} , it will be truncated to size \bar{N} using a *k-means* clustering strategy

(i.e., the final number of clusters \bar{N} is specified *a priori*). The \bar{N} individuals now constitute the external population \bar{P}_1 for the next generation. After archiving is completed, tournament selection, mating and mutation are applied to both \bar{P}_0 and P_0 to generate the new population P_1 . This procedure is repeated for each following generation. Since \bar{P}_t will be non-empty for $t > 0$, a preliminary nondomination reduction will be performed after the first nondominated front of P_t is copied into \bar{P}_t and before clustering is applied.

Clustering selection proceeds along the following lines: all distances between individuals in the front are calculated, and the two individuals with the shortest relative distance are selected and merged into a cluster. The procedure is repeated until the desired number of clusters is reached; then, for each cluster, the individual closest to the cluster centroid is selected for propagation into the new population.

In pseudocode:

- For each generation:
 - Get population of size N from previous generation;
 - Merge nondominated solutions in population and archive, recalculate nondominated;
 - If the nondominated set exceeds the archive size;
 - * Create a cluster for each individual;
 - * While the number of clusters exceeds the archive size:
 - Calculate distances between clusters, merge the two closest ones and calculate new cluster centroid;
 - * In each cluster, select the individual closest to the centroid and copy it into the archive;

Several different definitions of distance are reported in the literature, ranging from a simple euclidean metric to more elaborate, covariance matrix based expressions (Bosman (2003)). It is also possible to choose whether to cluster in decision or objective space. In this paper, we have used a normalized euclidean distance in objective space:

$$d_{ij} = \sqrt{\sum_{k=1}^M \left(\frac{f_k^{(i)} - f_k^{(j)}}{\max\{f_k^{(m)}\}_{m \in \mathcal{N}} - \min\{f_k^{(m)}\}_{m \in \mathcal{N}}} \right)^2}$$

where M is the number of objectives, $f_k^{(i)}$ is the k -th objective value of individual i , and \mathcal{N} is the set of individuals that is being reduced.

2.3 Elitism in MIDEA

The truncation method in MIDEA is a variant of clustering reduction implementing the *leader partitioning algorithm*. This represents a more sophisticated clustering procedure, allowing for the specification of the maximum cluster extension. Essentially, one individual is randomly picked (becoming the cluster “leader”), and all the individuals within a certain distance d are clustered with it. The procedure is repeated until all the individuals belong to exactly one cluster. Finally, each cluster’s leader is chosen as a representative (remember that since the maximum size of the clusters is controlled, the specific way a representative is chosen is not expected to have a large impact on the reduction result).

Unlike the k -means method, the number of selected individuals in this case is not fixed. If the set needs any further reduction, a crowding distance ranking truncation is applied, making this approach a combination of the two previously described.

We again resort to a normalized euclidean distance. This expression is here furtherly divided by the maximum distance between two elements in the set, leading to what is usually referred to as the Bounding Box Euclidean Normalized Distance, or BEND. This scaled quantity seem to be more universal than the simple normalized euclidean distance, and a prescription of 0.3 for the upper bound to impose during the clustering routine is reported in the literature (Bosman (2003)). This is the value we used in our study.

A schematic representation of the the clustering and the crowded tournament types of reduction is provided in Figure 1 and 2 from Deb et al. (2003).

3 CROWDED TOURNAMENT VS. CLUSTERING IN ϵ -NSGAII

3.1 ϵ -NSGAII STRUCTURE

The main loop can be synthesized as follows:

1. First run: the initial population of size N is initialized, and its members are ranked based on non-domination and crowding distance;
2. The first nondominated rank is copied into an offline archive of size A (after, possibly, ϵ -dominance reduction);

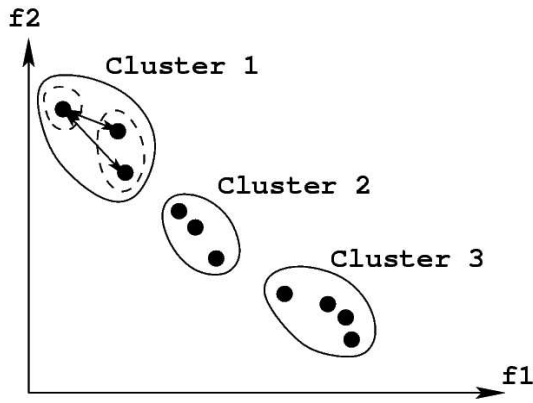


Figure 1: Clustering reduction of a nondominated front (Deb et al. (2003)).

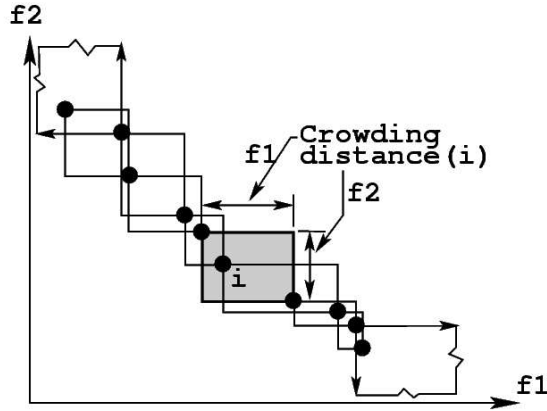


Figure 2: Crowded tournament reduction of a non-dominated front (Deb et al. (2003)).

3. Selection, mating and mutation produce N offsprings, which are merged with the N parent to form a size $2N$ mixed population;
4. The mixed population is truncated to size N by means of crowded tournament selection;
5. Steps 2 through 4 are repeated until some termination criterion is met;
6. Steps 1 through 5 are repeated with a population of size $4A$, comprising the A solutions in the archive from the previous run and $3A$ randomly generated individuals;
7. The population injection is repeated until one of the termination criteria is satisfied.

A more in-depth introduction to the structure of ϵ -NSGAII is provided in Kollat & Reed (2006).

We have added the two clustering-based archive truncation methods alternative to the crowded tournament selection in Step 4. The result of these substitutions and the comparison to the crowded tournament performance are described in the next section.

3.2 THE TEST PROBLEMS

A salient issue in the comparison of MOEAs is the design of test problems that are capable to probe the evolutionary dynamics of each algorithm, highlighting its advantages and its shortcomings. We have compared the new environmental selection to the old one over a number of challenging test functions drawn from two different classes.

The first is the ZDT family (Deb (2001)), which consists in the minimization of two objectives in the following general form:

$$\begin{aligned} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) &= g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x})) \end{aligned}$$

The family contains six cases, corresponding to six choices of the functions $f_1(\mathbf{x})$, $g(\mathbf{x})$ and $h(f_1(\mathbf{x}), g(\mathbf{x}))$, and of the length and range of the decision vector \mathbf{x} . For our tests, we have selected ZDT4 and ZDT6. The first is a 10 variable problem, with the first component of \mathbf{x} being restrained to the $[0, 1]$ interval, while all the others lie in $[-5, 5]$. The three functions are given by:

$$\begin{aligned} f_1(\mathbf{x}) &= x_1 \\ g(\mathbf{x}) &= 91 + \sum_{i=2}^{10} (x_i^2 - 10 \cos(4\pi x_i)) \\ h(f_1(\mathbf{x}), g(\mathbf{x})) &= 1 - \sqrt{\frac{f_1}{g}} \end{aligned}$$

ZDT6 is also a 10 variable problem, with all the components of \mathbf{x} in $[0, 1]$. The three functions are given by:

$$\begin{aligned} f_1(\mathbf{x}) &= 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ g(\mathbf{x}) &= 1 + 9 \left[\left(\sum_{i=2}^{10} x_i \right) / 9 \right]^{1/4} \\ h(f_1(\mathbf{x}), g(\mathbf{x})) &= 1 - \left(\frac{f_1}{g} \right)^2 \end{aligned}$$

The second family is constituted by the so called DTLZ test functions (Deb (2002)), from which we have selected problem 1 and 6. DTLZ1 is a 7 variable problem involving the minimization of the following three objectives:

$$f_1(\mathbf{x}) = \frac{1}{2}(1 + g(\mathbf{x}))x_1x_2$$

$$f_2(\mathbf{x}) = \frac{1}{2}(1 + g(\mathbf{x}))x_1(1 - x_2)$$

$$f_3(\mathbf{x}) = \frac{1}{2}(1 + g(\mathbf{x}))(1 - x_1)$$

with $x_i \in [0, 1]$ and

$$g(\mathbf{x}) = 100(5 + \sum_{i=3}^7 (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)))$$

Similarly, DTLZ6 is a 22 variable problem (again with $x_i \in [0, 1]$) involving the minimization of the three objectives:

$$f_1(\mathbf{x}) = \frac{1}{2}(1 + g(\mathbf{x})) \cos(\pi x_1/2) \cos(\theta_2)$$

$$f_2(\mathbf{x}) = \frac{1}{2}(1 + g(\mathbf{x})) \cos(\pi x_1/2) \sin(\theta_2)$$

$$f_3(\mathbf{x}) = \frac{1}{2}(1 + g(\mathbf{x})) \sin(\pi x_1/2)$$

with $\theta_2 = \pi(1 + 2g(\mathbf{x})x_2)/4(1 + g(\mathbf{x}))$ and

$$g(\mathbf{x}) = \sum_{i=3}^{22} x_i^{0.1}$$

We have run each test case for the two archive truncation methods and for ten random seeds. The convergence, diversity and ϵ -performance indicators are shown in Figures 3-6. Other primary details about ϵ -NSGAII's configuration for all the runs are reported in Table 1.

Table 1: Main ϵ -NSGAII parameter values used for the runs.

Generic	
Initial population size	12
Sizing scheme	Injection (25%)
ϵ -archiving value	0.0075
Δ	10%
δ	10%
SBX Crossover	
P_c	1.0
η_c	15
Polynomial Mutation	
P_m	0.0333
η_c	20
Performance Metrics	
ϵ -value	0.0075
Diversity grid specification	133

The data show that the clustered ϵ -NSGAII's average performance is comparable with that of the original algorithm.

4 CONCLUSIONS

A quick analysis of the performance plots shows that the clustered ϵ -NSGAII does not sensibly perform better than the crowded tournament counterpart. The three variants show comparable dynamics on the four test cases we have considered, leading to two possible conclusions:

- Additional improvements are needed in order to construct a competent clustering algorithm capable of tackling complex problems. Covariance matrix approaches seem very promising in this sense, but they are also more computationally demanding;
- The greed of ϵ -NSGAII is only marginally affected by the type of environmental selection implemented. This conclusion is partially supported by our results showing a minimal departure of the performance metrics from the original version of the algorithm. This may indicate that other directions for improvement are to be pursued. Recent results obtained with a parallel version of ϵ -NSGAII seem to confirm that the existence of more promising sources of improvement (P. Reed, private communication).

Acknowledgements

The author acknowledges P. Reed, J. Kollat and Y. Tang for extensive help and clarifications, as well as for providing the source code and the computational resources for carrying out the tests.

References

- P. Bosman (2003). *Design and applications of iterated density-estimation Evolutionary Algorithms*. Ph.D. thesis, Institute for Information and Computing Science (Utrecht).
- P. Bosman & D. Thierens (2003). *The balance between proximity and diversity in multiobjective evolutionary algorithms*. IEEE Transactions on Evolutionary Computation, Vol. 7, Number 2, 174.
- P. Bosman & D. Thierens (2001). *Advancing continuous IDEAs with mixture distributions and factorization selection metrics*. Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO-2001, M. Pelikan and K. Sastry Eds., San Mateo, CA: Morgan Kaufmann, p. 208.

K. Deb (2001). *Multi-objective optimization using evolutionary algorithms*. J. Wiley.

K. Deb, M. Mohan & S. Mishra (2003). *A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions*. KanGAL report 2003002. Indian Institute of Technology, Kanpur.

K. Deb et al. (2002). *Scalable multi-objective optimization test problems*. Proceedings of the 2002 Congress on Evolutionary Computation, p. 825.

J.B. Kollat & P.M. Reed (2006). *Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design*. Advances in Water Resources, Vol. 29, Issue 6, 792.

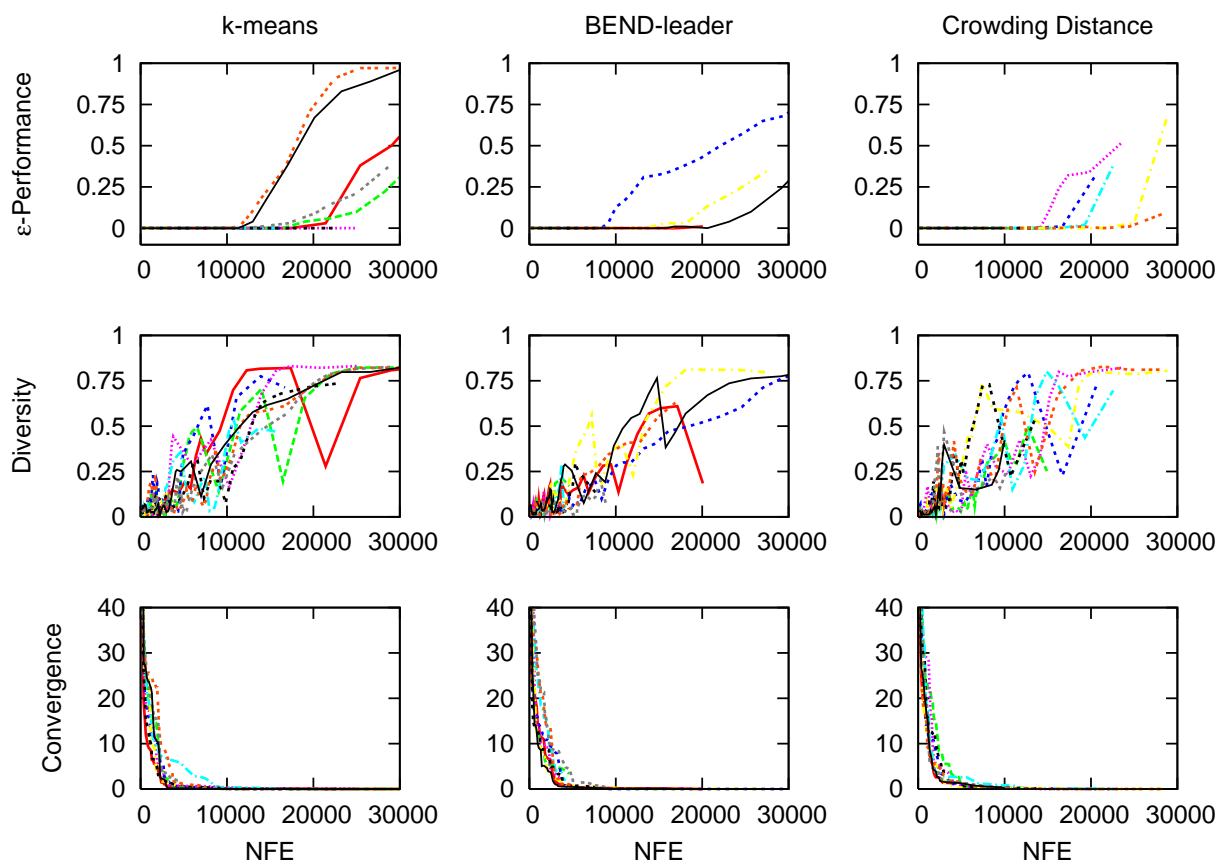


Figure 3: Performance metrics for ZDT4 (one curve per seed).

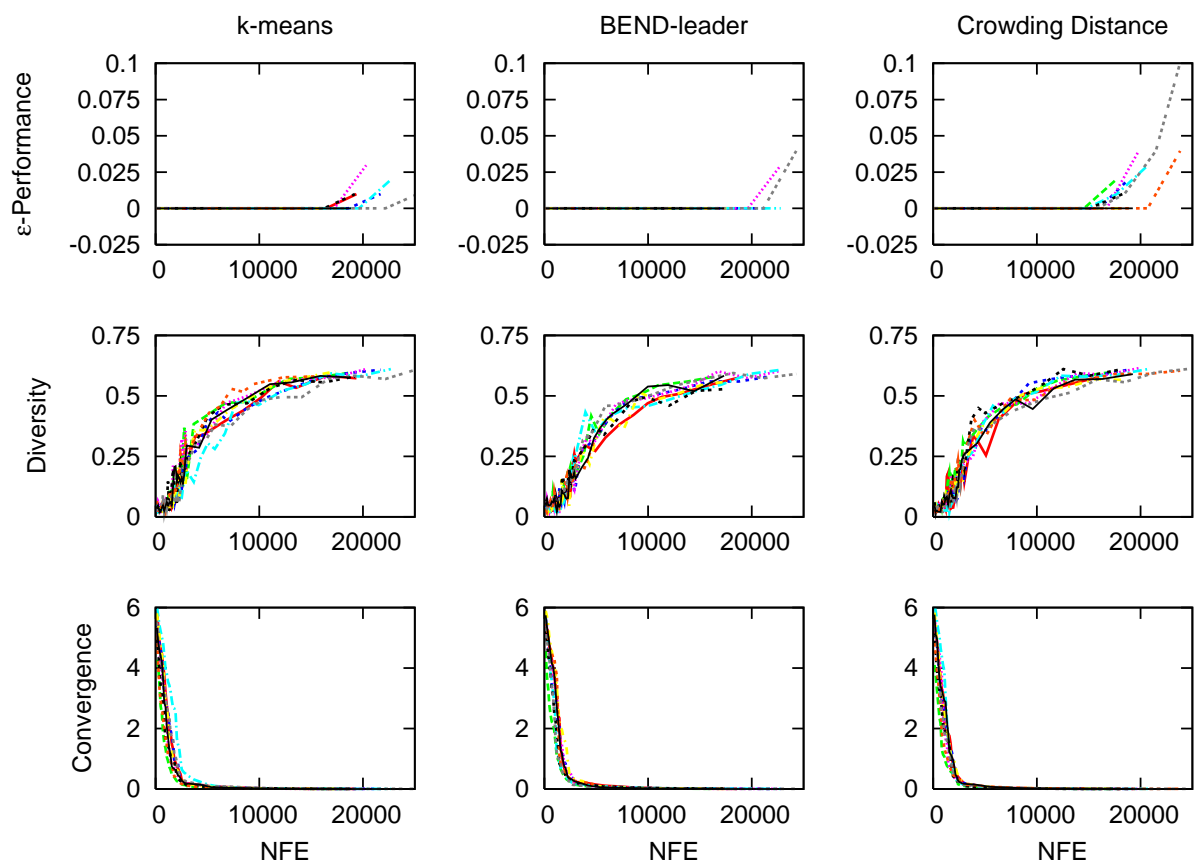


Figure 4: Performance metrics for ZDT6 (one curve per seed).

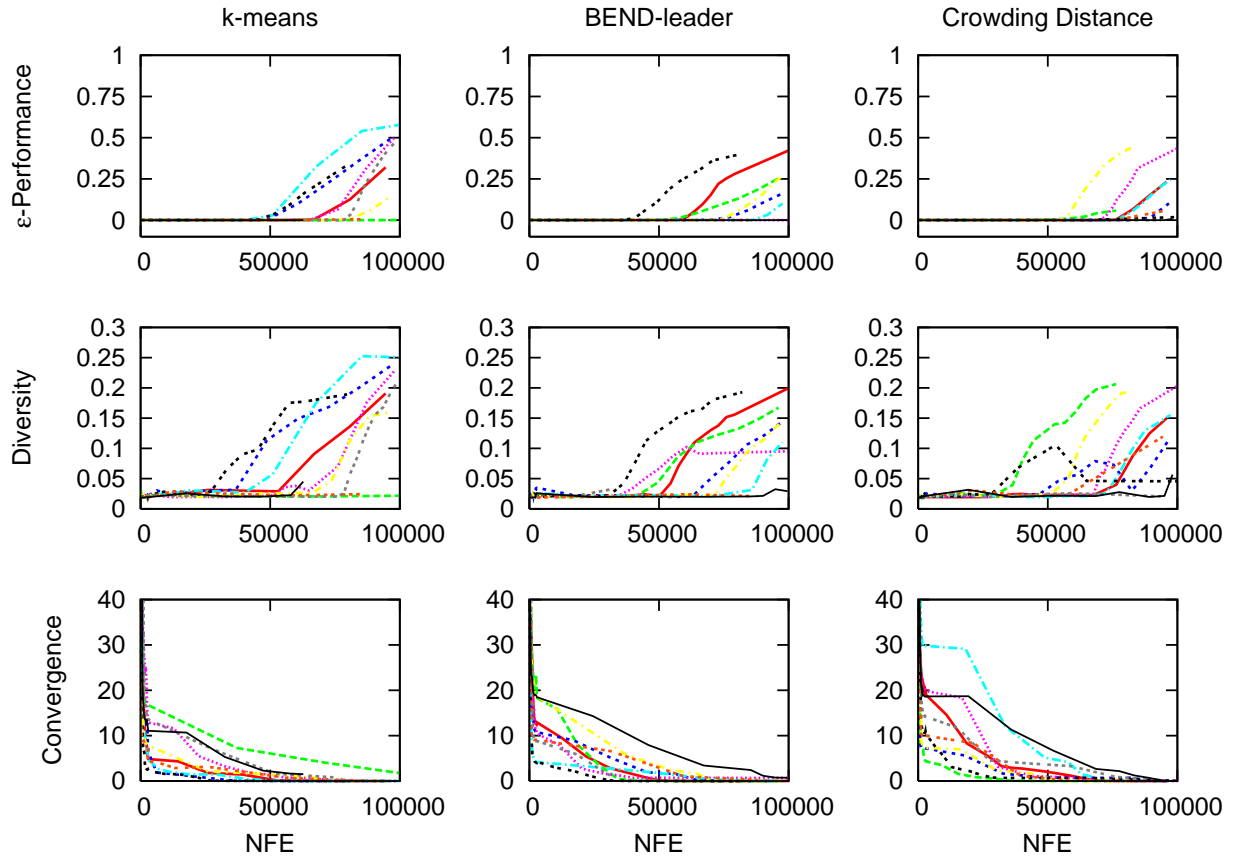


Figure 5: Performance metrics for DTLZ1 (one curve per seed).

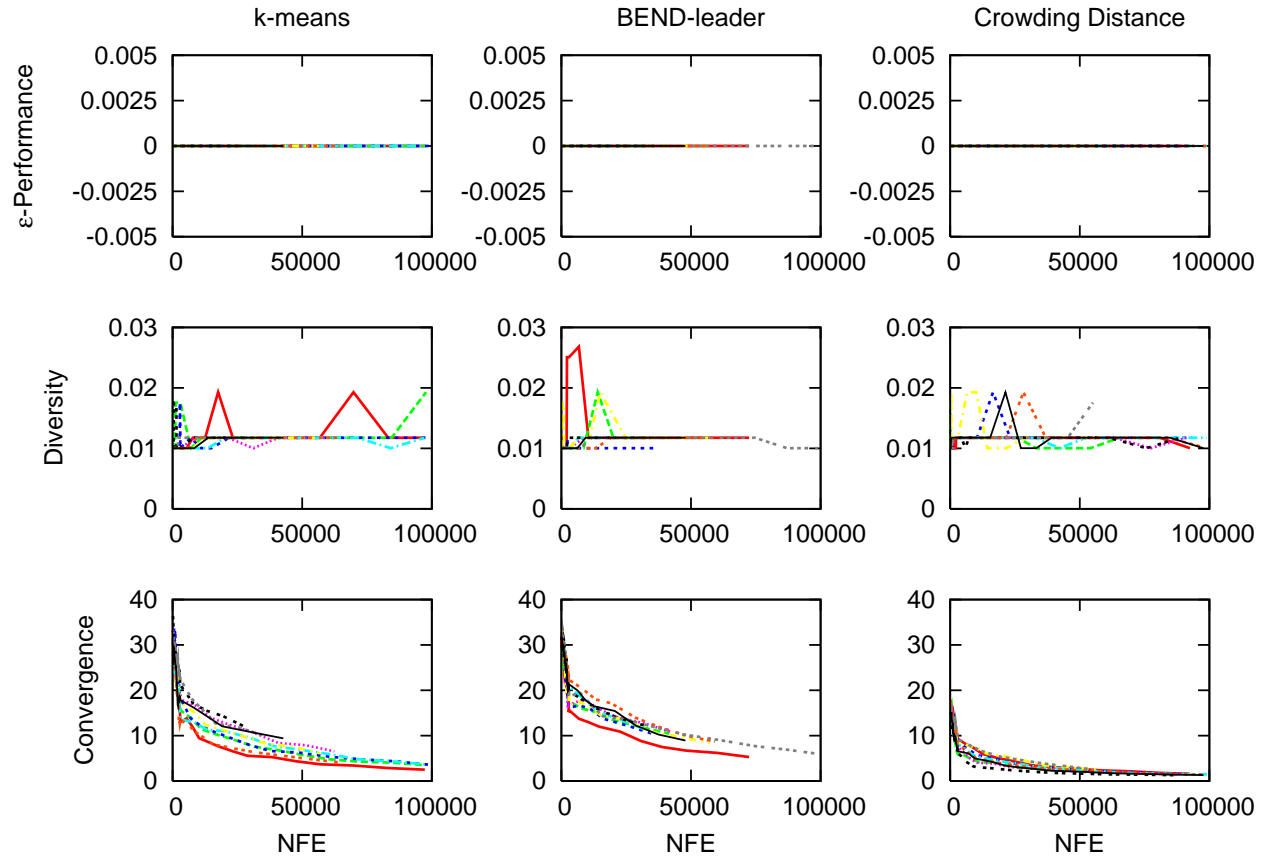


Figure 6: Performance metrics for DTLZ6 (one curve per seed).