PSA – A New Scalable Space Partition Based Selection Algorithm for MOEAs

Shaul Salomon, Gideon Avigad, Alex Goldvard, and Oliver Schütze

Abstract. It has generally been acknowledged that both proximity to the Pareto front and a certain diversity along the front should be targeted when using evolutionary algorithms to evolve solutions to multi-objective optimization problems. Although many evolutionary algorithms are equipped with mechanisms to achieve both targets, most give priority to proximity over diversity. This priority is embedded within the algorithms through the selection of solutions to the elite population based on the concept of dominance. Although the current study does not change this embedded preference, it does utilize an improved diversity preservation mechanism that is based on a recently introduced partitioning algorithm for function selection. It is shown that this partitioning allows for the selection of a well-diversified set out of an arbitrary given set. Further, when embedded into an evolutionary search, this procedure significantly enhances the exploitation of diversity. The procedure is demonstrated on commonly used test cases for up to five objectives. The potential for further improving evolutionary algorithms through the use of the partitioning algorithm is highlighted.

1 Introduction

In many problem situations, several objectives must be optimized at the same time, leading to a *multi-objective optimization problem* (MOP). Mathematically, a MOP can be stated as follows:

$$\min_{x \in Q} \{F(x)\}, \tag{1}$$

Shaul Salomon · Gideon Avigad · Alex Goldvard
ORT Braude College of Engineering, Karmiel, Israel
e-mail: {shaulsal,gideona,goldvard}@braude.ac.il

Oliver Schütze
Computer Science Department, CINVESTAV-IPN, Av. IPN 2508,
Col. San Pedro Zacatenco, Mexico City, Mexico
e-mail: schuetze@cs.cinvestav.mx

O. Schütze et al. (Eds.): EVOLVE – A Bridge between Probability, AISC 175, pp. 137–151.
springerlink.com

where $Q \subset \mathbb{R}^d$ is the domain, F is defined as the vector of the k objective functions $F : Q \to \mathbb{R}^k$, $F(x) = (f_1(x), \ldots, f_k(x))$, and each objective is defined as $f_i : Q \to \mathbb{R}$. The set of optimal solutions of a MOP, the *Pareto set* $\mathscr{P}$, typically forms a $(k-1)$-dimensional object. The task of many set-oriented search procedures is to find a 'suitable' finite sized approximation of the *Pareto front* $F(\mathscr{P})$ (i.e., the image of the Pareto set), since this front represents the set of optimal compromises measured in objective space, the most interesting view in many applications.
Of the set-oriented search procedures for the numerical treatment of MOPs, multi-objective evolutionary algorithms (MOEAs) are widely used due to their global and universal approach and their high robustness ([6, 5]). Most MOEAs simultaneously attempt to account both for the proximity of the approximation set to the Pareto front (also termed as *convergence* in the community) and for its diversity (also termed as *spread*) [3]. Domination is the predominant approach used to exploit proximity. Diversity is exploited by different approaches that can be classified into three main categories. The first treats diversity as a property of a set and evolves sets with good diversity. The diversity can be measured according to the accumulated distances between the members of the set (see [12], [13]), or indirectly by the hypervolume measure [20] or the averaged Hausdorff distance [14]. Algorithms in the second category treat diversity as a property of each individual according to the density of solutions surrounding it. Fitness sharing of NPGA [10], crowding distance of NSGA-II [7], the diversity metric based on entropy [15] and the density estimation technique used in SPEA2 [22] are examples of this category. Algorithms of the third category decompose the multi-objective problem into a number of single objective problems (scalarization). Each of these problems ideally aims for a different zone on the Pareto front such that the set of solutions to the auxiliary problems form a diverse set of optimal solutions. MOEA/D [19] is probably the most famous method within this category.

With respect to both goals (proximity and diversity), selection plays an important role in evolving individuals towards the Pareto front [1]. In order to select promising (i.e., less dominated and less crowded) individuals, proper selection criteria must be formulated. These criteria are commonly utilized in MOEAs in order to choose the elite population, thus allowing the best solutions in the current generation to be copied into the next generation. An elite population may be selected by pure truncation selection. In truncation selection with threshold τ, the algorithm sorts all individuals based on their domination level and includes the first τ individuals as the elite population. Truncation selection is exploited in many MOEAs, such as NSGA-II [7] and SPEA2 [22]. In this case, exploitation of proximity takes over exploitation of diversity. Some efforts to further promote population diversity have been made over the last decade, among them Balanced Truncation Selection (BTS) [3], epsilon Dominance Selection [11], Indicator-Based Selection, IBS [21].

In this paper, a new algorithm, PSA, is proposed to select a diverse subset from a given set of points. The proposed algorithm has several advantages that make it suitable for integration into MOEAs: 1) It has low computational complexity. 2) It can select a diverse subset, even if the original set is poorly distributed. 3) There are no limitations on the size of the selected subset.

The remainder of this paper is organized as follows. The proposed algorithm is described in Section 2. Section 3 describes a straightforward method for integrating the algorithm into NSGA-II in order to improve the diversity of its population. The proposed algorithm and the original NSGA-II are compared in Section 4. Finally, the paper ends with some conclusions in Section 5.

2 PSA – Part and Select Algorithm

This section proposes an algorithm that aims at selecting m well-spread points from a set of n candidate solutions. The proposed method is an adaption of the algorithm for the selection of functions (see [2]) to the context of multi-objective optimization. It selects a diverse subset from a set of objective vectors (e.g., the image of the current population of a MOEA). The procedure has two steps: First, the set is divided into subsets so that similar members are grouped in the same subset. Next, a diverse subset is formed by selecting one member from each generated subset.

2.1 Partitioning a Set

The core of the PSA is the algorithm of partitioning a given set of points in the objective space into smaller subsets. In order to partition a set into m subsets, PSA performs $m-1$ divisions of one single set into two subsets. At each step, the set with the greatest dissimilarity among its members is the one that is divided. This is repeated until the desired stopping criterion is met. The criterion can be either a predefined number of subsets (i.e., the value of m) or a maximal dissimilarity among each of the subsets. The dissimilarity of a set A is defined by the measure $\varnothing A$ as follows:

Let $A := \{\mathbf{f}_1 = [f_{11}, \ldots, f_{1k}], \ldots, \mathbf{f}_n = [f_{n1}, \ldots, f_{nk}]\} \subset \mathbb{R}^k$ (i.e., n objective vectors $\mathbf{f}_i = F(x_i)$ for points $x_i \in Q$), and denote

$$a_j := \min_{i=1,\ldots n} f_{ij},\ \ b_j := \max_{i=1,\ldots n} f_{ij},\ \ \Delta_j := b_j - a_j, \quad j = 1, \ldots, k \tag{2}$$

$$\varnothing A := \max_{j=1,\ldots k} \Delta_j \tag{3}$$

In fact, $\varnothing A$ is the diameter of the set A in the Chebyshev metric. The size of $\varnothing A$ is a measure of the dissimilarity among the members of A, with a large $\varnothing A$ indicating a large dissimilarity among the members of A.

The pseudocode of PSA for a fixed value of m is shown in Algorithm 1. At every iteration the algorithm finds the subset with the largest diameter, and divide it into two subsets.

Figure 1 demonstrates the steps of the algorithm and highlights the results obtained by its use. Consider the set of 24 points in the bi-objective space depicted in the top left panel of Figure 1. Suppose that the purpose is to partition this set into $m = 5$ subsets. The gray rectangle represents the region in the objective space that contains the solutions of the set. According to Eq. (3), the diameter of the given set

Algorithm 1 Partitioning a set A into m subsets

1: $A_1 \leftarrow A$
2: Evaluate $\varnothing A_1$ according to Eq. (3) and store $\varnothing A_1$ in an archive.
3: $i \leftarrow 2$
4: **while** $i < m$ **do**
5: Find A_j and coordinate p_j such that $\varnothing A_j = \Delta_{p_j} = \max_{l=1,\ldots i-1} \varnothing A_l$
6: Part A_j to subsets A_{j_1}, A_{j_2}:
$A_{j1} \leftarrow \left\{\mathbf{f} = [f_1, \ldots, f_{p_j}, \ldots, f_k] \in A_j, f_{p_j} \leq a_{p_j} + \varnothing A_j/2\right\}$
$A_{j2} \leftarrow \left\{\mathbf{f} = [f_1, \ldots, f_{p_j}, \ldots, f_k] \in A_j, f_{p_j} > a_{p_j} + \varnothing A_j/2\right\}$
7: Evaluate $\varnothing A_{j1}$ and $\varnothing A_{j2}$ according to Eq. (3), and replace in the archive $\varnothing A_j$ and p_j with the pairs $\varnothing A_{j1}, \varnothing A_{j2}$ and p_{j1}, p_{j2} accordingly.
8: $S \leftarrow \left\{A_1, \ldots, A_{j_1}, A_{j_2}, \ldots, A_i\right\}$
9: $i \leftarrow i+1$
10: **end while**

is the length of the horizontal side of the rectangle. Therefore, the first partition is made by vertical incision (indicated by the vertical line in the middle of the rectangle). The results of this partition are depicted in the top middle panel of Figure 1. The left subset in this panel has the greatest diameter (in horizontal direction). Therefore, the next partition is made on this subset by vertical incision. The results of this partition are depicted in the top right panel of Figure 1. The bottom two panels of Figure 1 depict the results of the next two iterations of Algorithm 1.

Note that the results of the partitioning are different than the results of using a common grid in the original space. With a common grid, an initial interval in every dimension is divided into equal sections, resulting in the division of the hyperbox into smaller hypberboxes of equal space. Since the original set A does not necessarily 'cover' the entire space, each hyperbox in the grid might or might not contain a member of A. Hence, there is no way to predict which resulting grid will have the desired number of occupied boxes. In addition, there are certain limitations on the number of hyperboxes in the grid. For example, in a two-dimensional grid it is possible to create $m = \{1,2,4,6,9,12,\ldots\}$ boxes, while only a number of $m = n^2$, when n is a positive integer, will produce an even grid. With PSA, only the occupied space (marked as the gray rectangles in Figure 1) is considered. When a group A_i is divided into two subgroups A_{i_1} and A_{i_2}, the space considered from now on is given only by the two hyperboxes circumscribing A_{i_1} and A_{i_2}. The rest of the space in A_i is discarded. Every partition increases the number of subgroups by one, and therefore any desired number of subgroups can be created. Moreover, by dividing the subgroup with the largest dissimilarity, PSA ensures that the members of each group will be as similar as possible to each other.

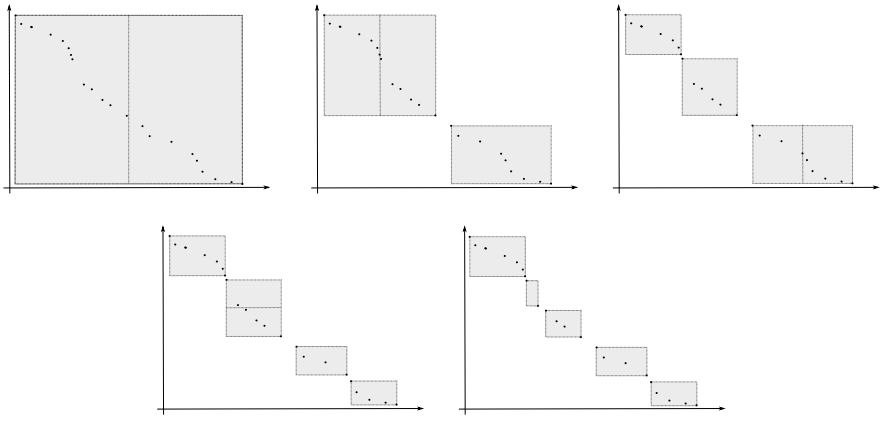


Fig. 1 Partitioning of 24 elements in bi-objective space into $m = 5$ subgroups (indicated by the gray boxes)

2.2 Selection of a Representative Subset

Once the set A has been divided into the m subsets $A_1, \ldots, A_m$, the 'most suitable' element from each subset must then be chosen in order to obtain a subset $A_{(r)}$ of A that contains m elements. This is of course problem dependent. Since this study aims for high diversity of the chosen elements, the following heuristic is suggested (denoted as center point selection): From each set A_i choose the member which is closest (in Euclidean metric) to the center of the hyperrectangle circumscribing A_i. If there exist more than one member closest to the center, one of them is chosen randomly.

Figure 2 illustrates this rule. The original set of 24 elements (compare to Figure 1) was partitioned by Algorithm 1 into five subsets. The centers of the grey rectangles are marked with a cross. In each subset the member closest to the center is circled (a random member is circled in the subset with only two members). The representative set $A_{(r)} = \{a_1, a_2, a_3, a_4, a_5\}$ is the set of all circled points.

Figure 3 illustrates the performance of PSA in selecting a subset from a randomly chosen (non-dominated) population in a three-objective space. A set of 500 randomly distributed points is depicted in Figure 3(a). The set is divided into 40 subsets, and the central member of each subset is selected as a representative point to form the representative subset depicted in Figure 3(f). According to Eq. (3), the diameter of the given set is the distance over f_2. Therefore, the first partition is made over f_2. At the second partition, the subset of the circles from Figure 3(b) has the largest dissimilarity and therefore is partitioned (over f_1). At the next partition the subset of gray stars is partitioned over f_1 to form the four subsets shown in Figure 3(d). The final stage of Algorithm 1 is shown in Figure 3(e). The subset shown in Figure 3(f) is obtained by selecting the point closest to the center of each

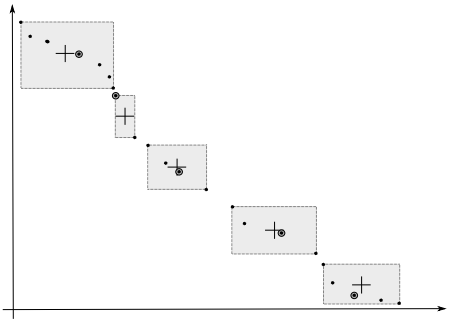


Fig. 2 Selection of a representative subset $A_{(r)}$ out of A using center point selection

of the 40 subsets. Figure 3(a) clearly shows that the distribution of the points in the original set is not uniform. Nevertheless, PSA managed to select a subset of fairly evenly distributed points from it.

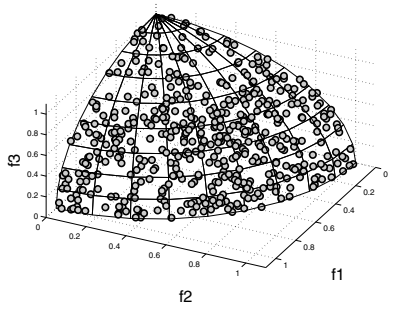
2.3 Complexity Analysis of the PSA

The following analysis refers to the procedure given in Algorithm 1. Let $A = (\mathbf{f}_1, \ldots, \mathbf{f}_n) \subset \mathbb{R}^k$ be the original set of n members, from which a subset of m members is to be selected.

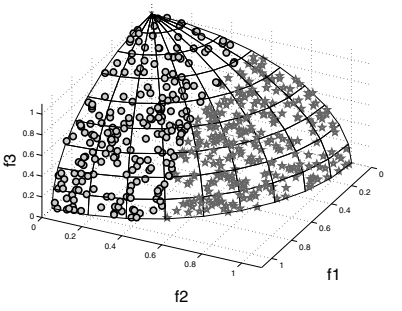
The evaluation of $\varnothing A_1$ (Step 2) requires a comparison $O(1)$ of n values to calculate Δ_j for each of the k coordinates. The complexity of this step is $O(nk)$. To find $\varnothing A_1 = \max_{j=1,\ldots,k} \Delta_j$, k comparisons of $O(1)$ are required. Thus, the total complexity of Step 2 is $O(nk)$. At Step 5 i comparisons are required in order to find p_j and $\varnothing A_j = \max_{l=1,\ldots,i} \varnothing A_l$, since it uses archiving. Given that $i \leq m$, the complexity of Step 5 is $O(m)$ in the worst case. To partition the set A_j into two subsets A_{j1} and A_{j2}, each of the members of A_j has to be compared in coordinate p_j with $a_j + \varnothing A_j/2$. Since $|A_j| = w_j \leq n$, the complexity of Step 6 is $O(n)$ in the worst case. The evaluation of $\varnothing A_{j1}$ and $\varnothing A_{j2}$ (Step 7) is similar to Step 2, but only w_j members have to be evaluated. In the worst case, the complexity of Step 7 is $O(nk)$. The complexity of Steps 5 to 7 is $O(m) + O(n) + O(nk) = O(nk)$. These steps are performed $m-1$ times. Therefore the entire complexity of Algorithm 1 is $O(nkm)$. The selection of one representative member of each set requires calculating the Euclidean distance of each member from the center of it's set. The complexity of this stage is $O(nk)$. The comparison between the distances has a complexity $O(n)$.

We summarize the above considerations.

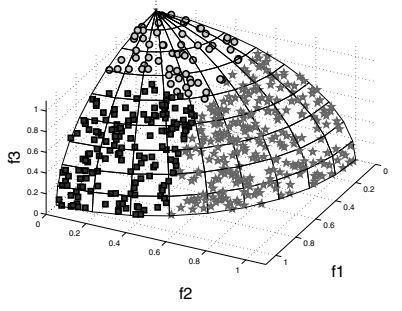
Proposition 1 *Given a set $A = (\mathbf{f}_1, \ldots, \mathbf{f}_n) \subset \mathbb{R}^k$, the overall complexity of choosing $m \leq n$ elements out of A using PSA and the center point selection is $O(nkm)$.*



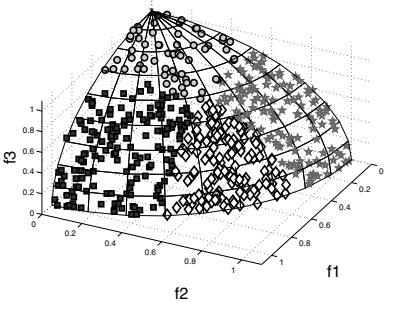
(a) original set of randomly distributed points



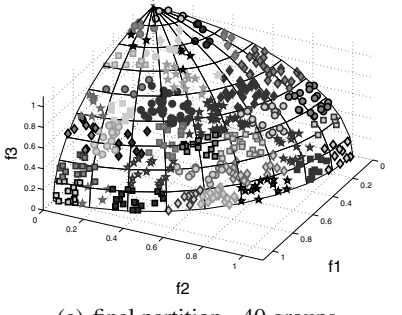
(b) two groups after first partition



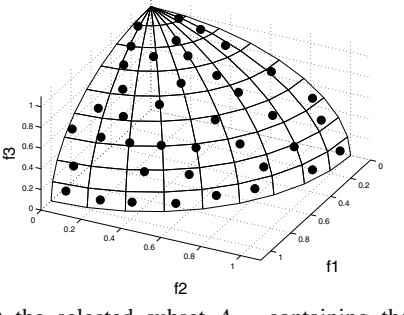
(c) three groups



(d) four groups



(e) final partition - 40 groups



(f) the selected subset $A_{(r)}$ containing the centers of the 40 groups

Fig. 3 Demonstration of PSA in a three-dimensional space: Selection of a representative subset of 40 points from a randomly distributed set of 500 points

3 Integration of PSA into NSGA-II

Due to the minimal requirements of PSA—the algorithm needs only a set of candidate solutions and returns a subset of cardinality m—the algorithm can basically be integrated into any MOEA and used as a selection mechanism and/or as a crowding assignment mechanism. The advantages of such an integration are demonstrated in the following using the well-known NSGA-II [7] as a base MOEA. NSGA-II is commonly used for solving MOPS. Nevertheless, it is known for its incompetence of finding a well-diversified set of solutions for problems with more than two objectives. It is shown in Section 4 that the suggested integration improves this drawback, and is able to find an approximated set with a better diversity. The purpose of the following is not to introduce an algorithm which is better than the state-of-the-art, but to demonstrate the simplicity of improving an existing MOEA by integrating it with PSA.

The modified NSGA-II will be referred to as "NSGA2-PSA". For the sake of simplicity, the implementation will be described using the same notations as in [7], page 6, "The Main Loop". Hence, assume that the combined current population $R_t = P_t \cup Q_t$ of the size $2N$ is sorted to the non-dominated fronts $\mathscr{F}_1, \ldots \mathscr{F}_l, \ldots$ and

$$|\mathscr{F}_1| + \ldots + |\mathscr{F}_{l-1}| = s < N, \quad |\mathscr{F}_1| + \ldots + |\mathscr{F}_l| > N.$$

The next parent population P_{t+1} is constructed from all the members of the sets $\mathscr{F}_1, \ldots, \mathscr{F}_{l-1}$ and from $N-s$ members of the set $\mathscr{F}_l$. The only difference from the classical NSGA-II is that instead of selecting $N-s$ members from the last front $\mathscr{F}_l$ according to the crowded-comparison operator $\prec_n$, selection is according to the PSA, as follows. $\mathscr{F}_l$ is partitioned to $N-s$ subsets according to Algorithm 1, and the central member of each subset is selected to P_{t+1}.

The crowding assignment of NSGA-II, used for the binary tournament is modified as well. Each set $\mathscr{F}_i, i = 1, \ldots l$ is partitioned according to Algorithm 1 into $|\mathscr{F}_i|/2$ subsets, and every member of $\mathscr{F}_i$ is assigned a crowding measure equal to the number of members of its subset.

To highlight the benefits of this integration, no other differences were made between NSGA-II and NSGA2-PSA. In particular, NSGA2-PSA uses the same tournament-selection, cross-over, and mutation operators as NSGA-II.

4 Comparison of NSGA2-PSA with NSGA-II

In this section, the proposed NSGA2-PSA is compared with the classic NSGA-II. For a fair comparison, both MOEAs start with the same initial population at every test, and use the same genetic settings.

4.1 Test Problems

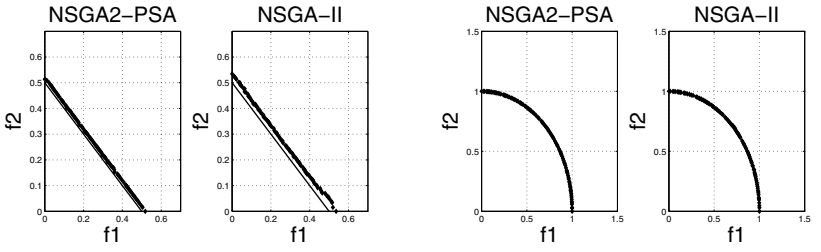
To demonstrate the scalability of the PSA, both algorithms are tested on the DTLZ 1 – 4 benchmark functions (see [8]). Each test case was tested for $p \in \{2,3,4,5\}$ objectives. The design space consists of $d = 7$ decision variables for all problems. The Pareto optimal front (PF) of DTLZ1 is the hyper-plane of all points that satisfy the equation $\sum f_i = 0.5, i = 1,\ldots,M$, where M is the number of objectives. This benchmark offers a difficult function for convergence with many local minima. The PF of DTLZ2 – 4 is the positive eighth of the hyper-sphere of all points that satisfy the equation $\sum f_i^2 = 1, i = 1,\ldots,M$. DTLZ3 uses the same function for convergence as in DTLZ1, and therefore it can check the ability of a MOEA to find solutions close to the real PF. In DTLZ4 a very small section in the decision space is mapped to a very large section in the objective space. This property makes it a good benchmark to test the ability of a MOEA to find a diverse set of solutions. Since the proposed NSGA2-PSA aims to enhance the diversity of the approximated set, the results for the DTLZ4 are of major importance.

4.2 Genetic Settings

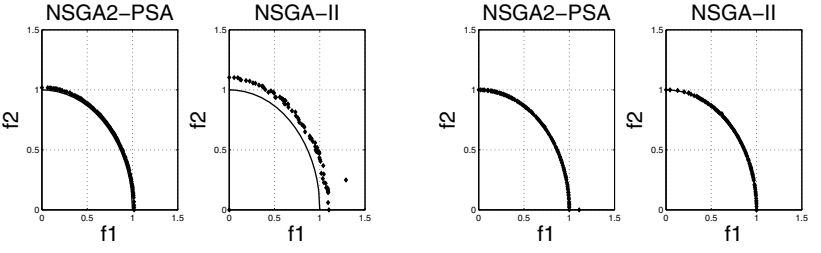
Both algorithms are given real-valued decision variables. They use the simulated binary crossover (SBX) operator and polynomial mutation [6], with distribution indexes of $\eta_c = 15$ and $\eta_m = 20$ respectively. A cross-over probability of $p_c = 1$ and a mutation probability of $p_c = 1/d$ are used. The population size is increased with the number of objectives. A population size of 100, 300, 600 and 1000 members is taken for 2, 3, 4 and 5 objectives, respectively. The number of generations for all the cases is set to 250.

4.3 Performance Metrics

The approximated sets obtained by both methods are evaluated according to the hyper-volume [20], generational distance [23] and inverted generational distance [16]. The generational distance (GD) measures the average distance of the members in the final approximated set to the real PF. Therefore, GD measures the convergence of the algorithm. Since the mathematical definition of the fronts of the test-cases are known, the distances are calculated analytically. The inverted generational distance (IGD) measures the average distance from a set of well-diversified points on the real PF to the final approximated set. Since the reference set is well spread along the PF, IGD provides information about the diversity of the approximated set. Smaller values of GD and IGD are preferred. The hyper-volume (HV) measures the size of the region dominated by the approximated set. A higher value of HV is associated with a good spread and convergence.

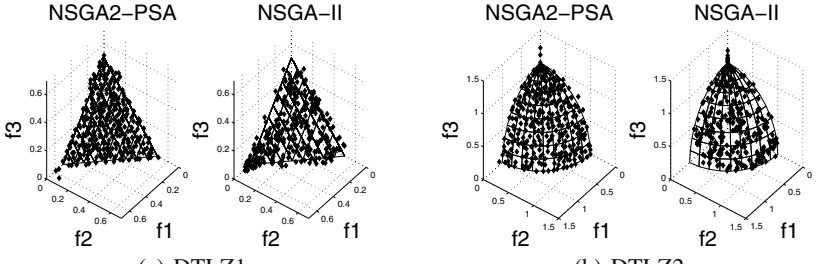


(a) DTLZ1 (b) DTLZ2

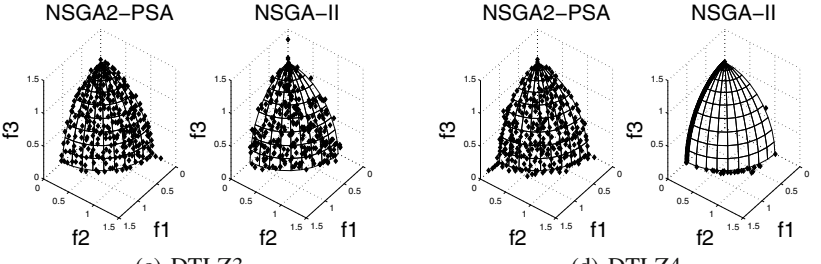


(c) DTLZ3 (d) DTLZ4

Fig. 4 The final approximated set obtained by NSGA2-PSA and NSGA-II on two objectives DTLZ1 – DTLZ4

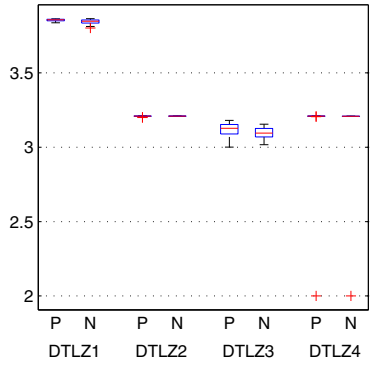


(a) DTLZ1 (b) DTLZ2

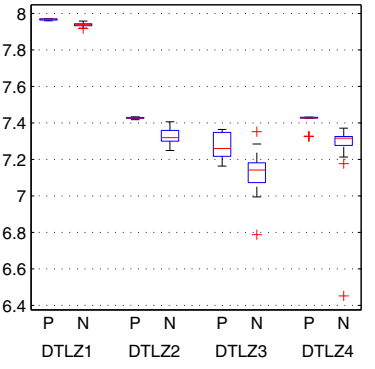


(c) DTLZ3 (d) DTLZ4

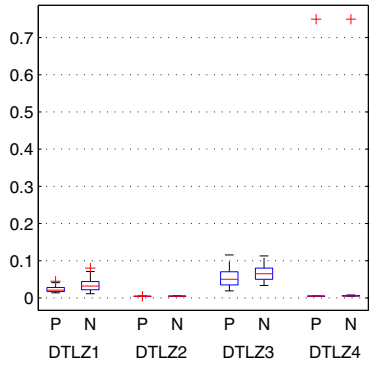
Fig. 5 The final approximated set obtained by NSGA2-PSA and NSGA-II on three objectives DTLZ1 – DTLZ4



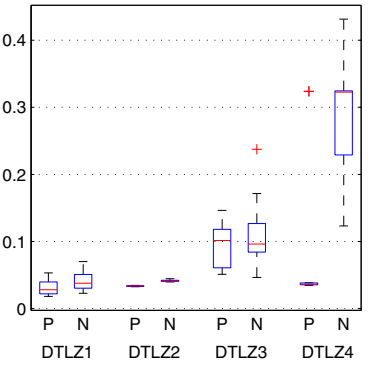
(a) HV results for 2 objectives



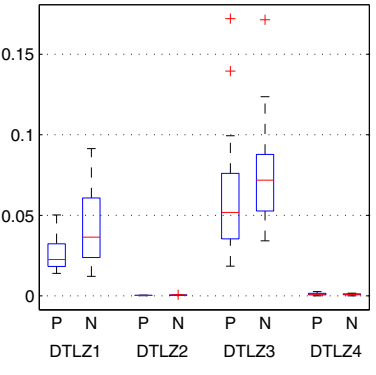
(b) HV results for 3 objectives



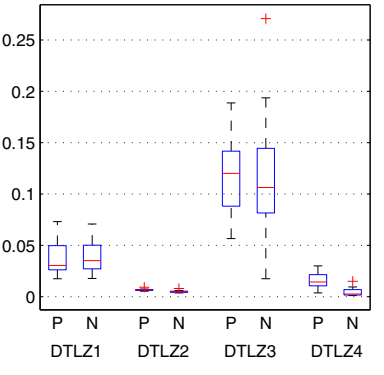
(c) IGD results for 2 objectives



(d) IGD results for 3 objectives

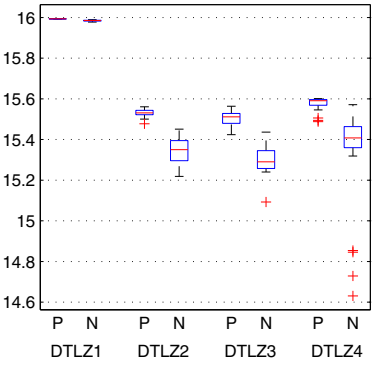


(e) GD results for 2 objectives

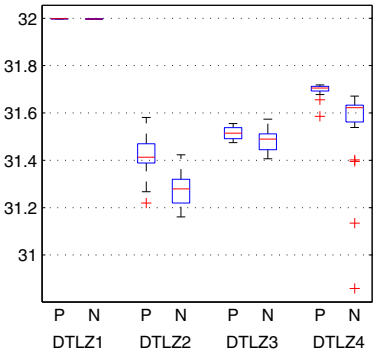


(f) GD results for 3 objectives

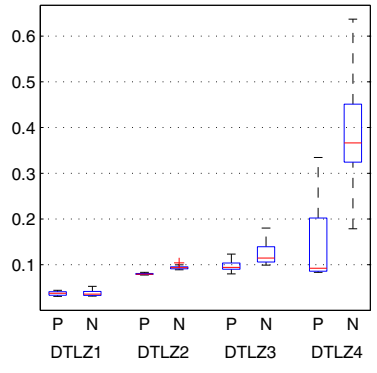
Fig. 6 Box plots of results from 20 independent runs of NSGA-II (denoted as N) and NSGA2-PSA (denoted as P) on the DTLZ 1 – DTLZ 4 test cases for two and three objectives



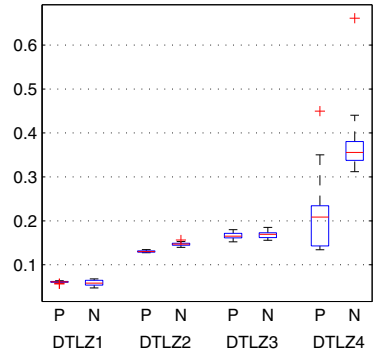
(a) HV results for 4 objectives



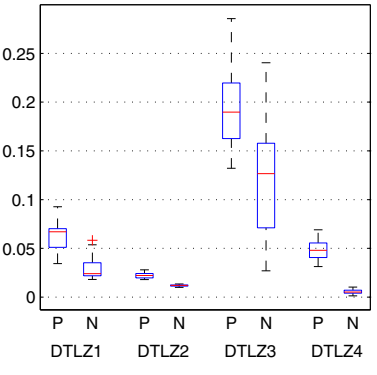
(b) HV results for 5 objectives



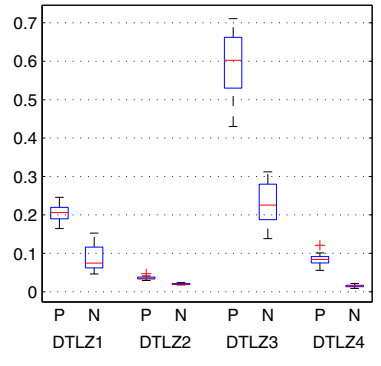
(c) IGD results for 4 objectives



(d) IGD results for 5 objectives



(e) GD results for 4 objectives



(f) GD results for 5 objectives

Fig. 7 Box plots of results from 20 independent runs of NSGA-II (denoted as N) and NSGA2-PSA (denoted as P) on the DTLZ 1 – DTLZ 4 test cases for four and five objectives

4.4 Simulation Results

To compare the methods, 20 independent runs were carried out on each of the problems. The results of a typical approximated set can be seen for two and three objectives in Figures 4 and 5, respectively. The statistical results for two – five objectives are shown in Figures 6 and 7. As seen in Figures 4 and 6(e), NSGA2-PSA has a slight advantage in convergence for problems with two objectives. That advantage can be seen in the results for the DTLZ1 and DTLZ3 problems. The differences between the methods begin to emerge for problems with three objectives and more. Figure 5 clearly shows that the spread of NSGA2-PSA is much better than that of NSGA-II for all test cases. The most outstanding difference occurs with the results of DTLZ4. As stated in [8], the final approximated set for DTLZ4 is highly dependent on the initial population. For three objectives, NSGA-II converged in some runs to the solutions on the f1-f2 plane, in others to the f1-f3 plane and in some runs it found solutions on the surface of the hyper-sphere. With the same initial population, NSGA2-PSA was able to find solutions on the surface of the hyper-sphere for almost every run. A good example of this can be seen in Figure 5(d).

The advantage of NSGA2-PSA over NSGA-II in spreading the approximated set is depicted in the statistical data in Figures 6 and 7. The IGD value of NSGA2-PSA is always better than that of NSGA-II. However, in problems with more than two objectives, this advantage comes at the cost of a loss of proximity, as manifested in the larger GD values of NSGA2-PSA. This highlights the unresolved tradeoff between proximity and diversity. The combination of small IGD values with large GD values can be explained by the existence of outliers. The higher HV measure of NSGA2-PSA indicates that the gain in diversity is more significant than the loss of proximity. The advantage of NSGA2-PSA makes the largest difference in the DTLZ4 test case, where a well-diversified set is hard to attain.

5 Conclusions and Future Work

In this paper, PSA (Part and Select Algorithm) has been proposed as a tool to select m elements from a given set A of candidate solutions such that the resulting subset $A_{(r)}$ is characterized by high diversity (and hence serves as a good representation of A). The idea of PSA was first used in [2] for function selection, but in this paper it is used and analyzed for the first time as a selection mechanism within MOEAs. The PSA together with center point selection is capable of choosing a well-spread subset $A_{(r)}$ of A for any given value of m. To demonstrate the potential benefit of the novel algorithm, PSA has been integrated into the well-known NSGA-II. The purpose of this was to utilize the high competency of the partitioning algorithm to select a diversified sub-set from an arbitrary set for enhancing diversity along the Pareto front. The partitioning algorithm has been used to assign a crowding measure to each of the solutions, and was used within the evolutionary algorithm. It has been shown that the selection based on the partitioning algorithm evolves a

front involving higher diversity (based on IGD and on hypervolume). Nevertheless, the inherent trade-off between proximity and diversity was not resolved.

As future work, the potential of the partitioning algorithm should be tapped to select a good spread of solutions, e.g., by integrating it in other MOEAs such as BTS [3] or Δ_p-EMOA [9]. The BTS algorithm aims directly at proximity and diversity while the Δ_p-EMOA is aimed at Hausdorff approximations of the Pareto front (i.e., a performance measure that is also related to proximity and divesity [14]). Moreover, due to the advantages of PSA in choosing a representative subset, its potential for supporting decision-makers should be explored.

Acknowledgements. This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme. The last author acknowledges support from CONACyT project no. 128554.

The authors would like to thank Alan R. R. Freitas for his help with the complexity analysis.

References

1. Ahn, C.W., Ramakrishna, R.S.: A diversity preserving selection in multiobjective evolutionary algorithms. Applied Intelligence 32(3), 231–248 (2010), doi:10.1007/s10489-008-0140-0
2. Avigad, G., Goldvard, A., Salomon, S.: Time-response-based evolutionary optimization. Tech. Rep. br352012, Ort Braude Academic College (2012), `http://brd.braude.ac.il/gideon`
3. Bosman, P.A.N., Thierens, D.: The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(2), 174–188 (2003)
4. Coello Coello, C.A., van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer (2007)
5. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., New York (2001)
6. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. 9, 115–148 (1995)
7. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimisation: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000), `http://dl.acm.org/citation.cfm?id=645825.668937`
8. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. Report 2 001 001, Kanpur Genetic Algorithms Lab (KanGAL), Indian Inst. Technol., Kanpur, India (2001)
9. Gerstl, K., Rudolph, G., Schütze, O., Trautmann, H.: Finding evenly spaced fronts for multiobjective control via averaging Hausdorff-measure. In: Int'l. Proc. Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2011), pp. 975–980 (2011)

10. Horn, J., Nafpliotis, N., Goldberg, D.E.: A niched Pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, pp. 82–87. IEEE Press (1994)
11. Laumanns, M., Očenášek, J.: Bayesian Optimization Algorithms for Multi-objective Optimization. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 298–307. Springer, Heidelberg (2002)
12. Li, M., Zheng, J., Xiao, G.: An efficient multi-objective evolutionary algorithm based on minimum spanning tree. In: Evolutionary Computation, CEC 2008 (IEEE World Congress on Computational Intelligence), pp. 617–624 (2008), doi:10.1109/CEC.2008.4630860
13. Wineberg, M., Oppacher, F.: The Underlying Similarity of Diversity Measures Used in Evolutionary Computation. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 1493–1504. Springer, Heidelberg (2003)
14. Schütze, O., Esquivel, X., Lara, A., Coello Coello, C.A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. IEEE Transactions on Evolutionary Computation 1 (to appear, 2012), doi:10.1109/TEVC.2011.2161872
15. Shen, X., Zhang, M., Li, T.: A multi-objective optimization evolutionary algorithm addressing diversity maintenance. In: International Joint Conference on Computational Sciences and Optimization, CSO 2009, vol. 1, pp. 524–527 (2009), doi:10.1109/CSO.2009.28
16. Sierra, M.R., Coello, C.A.C.: Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ϵ-Dominance. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 505–519. Springer, Heidelberg (2005)
17. Solow, A., Polasky, S.: Measuring biological diversity. Environmental and Ecological Statistics 1, 95–103 (1994), http://dx.doi.org/10.1007/BF02426650, doi:10.1007/BF02426650
18. van Veldhuizen, D.A.: Multiobjective evolutionary algorithms: Classifications, analyses and new innovations. Ph.D. thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA (1999)
19. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007), doi:10.1109/TEVC.2007.892759
20. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications. Ph.D. thesis, ETH, Zurich, Switzerland (1999)
21. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
22. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K.C., Tsahalis, D.T., Périaux, J., Papailiou, K.D., Fogarty, T. (eds.) Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems. International Center for Numerical Methods in Engineering, Athens, Greece, pp. 95–100 (2001)