Multi-objective optimization approach based on Minimum Population Search algorithm

Darian Reyes-Fernández-de-Bulnes
darian.reyes@tectijuana.edu.mx
Instituto Tecnológico de Tijuana, Mexico

Antonio Bolufé-Röhler
aboluferohler@upei.ca
University of Prince Edward Island, Canada

Dania Tamayo-Vera
dania@thinkingbig.net
Thinking Big Inc., Canada

ABSTRACT

Minimum Population Search is a recently developed metaheuristic for optimization of mono-objective continuous problems, which has proven to be a very effective optimizing large scale and multi-modal problems. One of its key characteristic is the ability to perform an efficient exploration of large dimensional spaces. We assume that this feature may prove useful when optimizing multi-objective problems, thus this paper presents a study of how it can be adapted to a multi-objective approach. We performed experiments and comparisons with five multi-objective selection processes and we test the effectiveness of Thresheld Convergence on this class of problems. Following this analysis we suggest a Multi-objective variant of the algorithm. The proposed algorithm is compared with multi-objective evolutionary algorithms IBEA, NSGA2 and SPEA2 on several well-known test problems. Subsequently, we present two hybrid approaches with the IBEA and NSGA-II, these hybrids allow to further improve the achieved results.

KEYWORDS: Evolutionary Algorithm, Minimum Population Search, Thresheld Convergence, Multi-objective Optimization

INTRODUCTION

Many industrial domains are concerned about large and complex optimization problems involving multiple criteria. Indeed, optimization problems encountered in practice are seldom mono-objective. In general, there are many conflicting objectives to handle. Multi-objective Optimization (MOO) is dedicated to solve problems in which a set of objective functions $f_1(x), f_2(x), \ldots, f_m(x)$ must be optimized simultaneously (1):

$$\min F(x) = \langle f_1(x), f_2(x), \ldots, f_m(x) \rangle, x \in D \quad (1)$$

where D is known as the decision space. The image set O, which results from projecting $F: D \rightarrow O$ is called the objective space, which is the space where the objective vectors belong.

Multi-objective Optimization Evolutionary Algorithms (MOEAs) are among the state of the art in solving Multi-objective Optimization Problems (MOPs), however, these algorithms still face important challenges. For instance, the scalability of algorithms by increasing the dimensions of both the decision space and the objective space, or the need to increase efficiency with respect to the number of evaluations of objective functions (Zitzler and Thiele, 1999). For that reasons, the development of new MOEAs remains an active area of research (Talbi, 2009). Following this approach, this paper proposes a new MOEA based on the heuristic Minimum Population Search (MPS) (Bolufé-Röhler and Chen, 2013).

Minimum Population Search is a recently developed metaheuristic for optimization of mono-objective continuous problems. A novel and key contribution of MPS is that its design is based on the hypothesis that the effectiveness of exploration can be affected by the concurrence with exploitation, thus, MPS attempts to separate these two processes (Bolufé-Röhler and Chen, 2014). To achieve this, Minimum Population Search is the first metaheuristic to use Thresheld Convergence (TC) as an integral part of its design. Thresheld Convergence is a diversification technique which attempts to separate these processes through the use of a *“threshold”* function to establish a minimum search step, managing this step makes it possible to influence the transition from exploration to exploitation, convergence is thus *“held”* back until the last stages of the search process (the word *thresheld* is a combination of threshold and held).

The goal of a controlled transition is to avoid an early concentration of the population around a few attraction basins which can bias exploration towards these regions, causing a loss of diversity and premature convergence. Thresheld Convergence has been successfully applied to several population-based metaheuristics such as Differential Evolution (Bolufé-Röhler et al., 2013), Evolution Strategies (Piad et al., 2015) and Estimation of Distribution Algorithm (Tamayo-Vera et al., 2016), but MPS is the first evolutionary algorithm to have TC built into its original design. This allows MPS to provide an effective and unbiased exploration of the search space, which has proven to be very effective as part of hybrid algorithms. A comprehensive third party study (Glorieux et al., 2017) has rated MPS-CMAES hybrid as the second best of 15 tested methods for Large Scale Global Optimization.

In this paper we hypothesize that the strong exploration of MPS will lead to good results in multi-objective optimization. Five multi-objective selection methods are tested on MPS and computational results are analyzed using quality indicators and nonparametric statistical tests. The proposed Multi-objective Minimum Population Search (MMPS) is compared against other MOEAs considering several MOPs of a standard benchmark.

However, we believe that the original design of Threhseld Convergence, designed to achieve a complete convergence upon a single point, may affect the diversification of solution throughout the Pareto front. Thus we will test some modification to TC, in an attempt to adjust this technique to the requirements of MOO. We also test whether hybrid algorithms that use MPS for an initial exploration, and then rely on well-known MOEA for the final diversification, allow to further improve results.

The remainder of the paper is organized as follows. In the next two sections several concepts of multi-objective optimization and the MPS algoithm are presented. Then, the Multi-objective MPS approach is described, as well as two hybrid algorithms based on MMPS. In the final section the computational results are described and analyzed. Finally, conclusions and future work are discussed.

MULTI-OBJECTIVE OPTIMIZATION

A Multi-objective Optimization Problem (MOP) involves a number of objective functions which are to be either minimized or maximized. As in a mono-objective optimization problem, the MOP may contain a number of constraints which any feasible solution (including all optimal solutions) must satisfy (Deb, 2001), (Deb, 2011). The MOEAs solution quality is commonly expressed in terms of Pareto dominance (Talbi, 2009). An objective vector $u = (u_{1},..., u_{m})$ is said to dominate $v = (v_{1},..., v_{m})$ if and only if no component of v is smaller than the corresponding component of u and at least one component of u is strictly smaller, that is (2):

$$\forall i \in \{1, \dots, m\}: u_{i} \leq v_{i} \wedge \exists i \in \{1, \dots, m\}: u_{i} < v_{i} \quad (2)$$

Multi-objective optimization problems

MOPs can be divided into two categories: those whose solutions are encoded with real-valued variables, also known as continuous MOPs, and those where the solutions are encoded using discrete variables such as combinatorial MOPs. This paper focuses in continuous MOPs.

Some continuous test functions have been proposed to easily carry out MOEAs experiments (Zitzler et al., 2000), (Deb et al., 2005). Test problems allow a fairer evaluation of the efficiency and the effectiveness of MOEA. Some well-known test problems are used in the literature for continuous MOPs, such as the DTLZ and ZDT benchmarks (Deb et al., 2005). For some test problems, the optimal PF is known *a priori* which makes easier to assess the performance of MOEAs. Indeed, in the ZDT class of instances, a simple procedure has been designed to construct bi-objective optimization problems in which different characteristics of the PF landscape and difficulties are introduced (Talbi, 2009). The test presented on this paper were performed with DTLZ and ZDT benchmark problems, using the Platform and programming language independent Interface for Search Algorithms (PISA) framework (Bleuler et al., 2003) which contains implementations of these standard MOPs in C++ language.

Multi-objective optimization evolutionary algorithms

Evolutionary algorithms are population-based stochastic search procedures which iteratively emphasize the best solutions. The best solutions are recombined and locally perturbed in the hope of creating a new and better population until a predefined termination criterion is met. There are many MOEAs in the literature. However, there are a few MOEAs that have stood out for their good performance and that are used on this paper for comparison and hybridization:

- Indicator-Based multi-objective Evolutionary Algorithm (IBEA): Its fitness assignment scheme is based on a pairwise comparison of solutions contained in a population by using a binary quality indicator. The selection scheme is a binary tournament between randomly chosen individuals. The replacement strategy is an environmental one that consists in deleting, one-by-one, the worst individuals and in updating the fitness values of the remaining solutions each time there is a deletion; this step is iterated until the required population size is reached (Zitzler and Kunzli, 2004).
- Non-dominated Sorting Genetic Algorithm II (NSGA-II): Individuals are ranked according two criteria. First, using the non-dominance concept. Within a given rank, solutions are ranked again according to the crowding distance. Solutions with high crowding distance are considered better solutions, as they introduce more diversity in the population (Deb et al., 2002).
- Strength Pareto Evolutionary Algorithm 2 (SPEA2): It maintains a fixed number of Pareto solutions in a separate archive. This elite archive participates exclusively in the generation of the solutions and is used in the fitness assignment procedure. A clustering algorithm is used to control the cardinality of the archive by maintaining the diversity among the Pareto solutions of the archive (Zitzler et al., 2001).

Once the MOEAs are executed, it is necessary to measure the quality of the obtained PFs. This task is performed using quality indicators. We employ three quality indicators for measure convergence to the optimal PF and diversity of solutions along the PF: Epsilon (Zitzler et al., 2003), Hypervolume (Bader, 2010) and R (Hansen and Jaszkiewicz, 1999).

Nonparametric statistical tests perform paired comparisons of algorithms applied to the test problem and estimate by means of the p-value which one is the best. A *p*-value provides information about whether a statistical hypothesis of a test is significant or not (lower *p*-value, stronger is the evidence against hypothesis null $H_{\{0\}}$, which states that the means of the two samples analyzed are equivalents). We employ one statistical testing procedures for comparing quality indicator values samples: Fisher-Indep (Conover, 1999).

Quality indicators are implemented within PISA framework. PISA performance assessment is based on separating the optimization process into the optimization problem (Variator) and the selection process (Selector). It extends it by a set of statistical tools that allow assessing and comparing different optimization methods.

MINIMUM POPULATION SEARCH

MPS is a recently developed metaheuristic for optimization of mono-objective continuous problems. The key idea is to guarantee a full search into all dimensions using the minimum required population size (Bolufé-Röhler and Chen, 2014), (Bolufé-Röhler et al., 2014). To generate new solutions, line segments are used to search within the *d - 1* dimensional hyperplane, and full coverage of the search space is then achieved by taking a subsequent step that is orthogonal to this hyperplane. To preserve the diversity of the (small) population and avoid premature convergence, the Thresheld convergence (Chen et al., 2015) technique is used. By establishing a minimum search step, thresheld convergence disallows new solutions which are too close to members of the current population, and this ensures a strong exploration during the early stages of the search. The minimum step threshold decays as the search progresses and convergence is thus held back until the last stages of the search process.

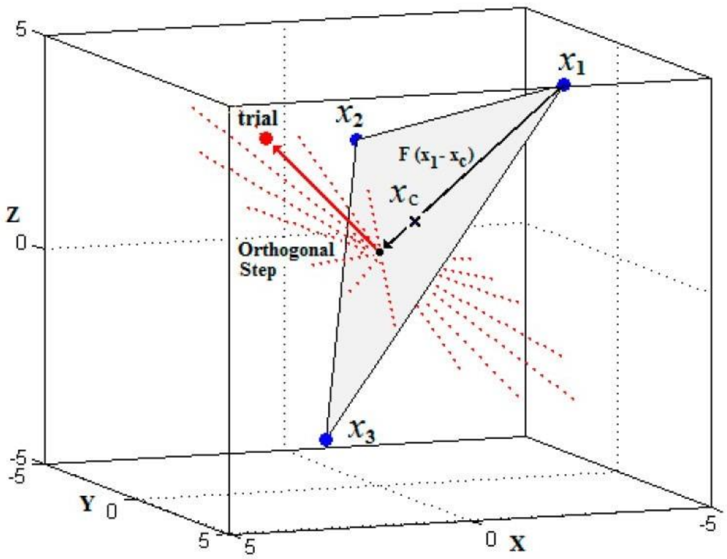
When multi-modal functions are optimized, the search process is commonly divided by two stages:

- Exploration: Consists of finding the best attraction basin.
- Exploitation: Consists of converging to the local optimum of a given attraction basin.

At every generation a new solution $trial_i$ is created from each population member x_i. First, from each parent solution x_i a step inside the *d - 1* hyperplane (formed by the *n* population members) is performed. Then, an orthogonal step is made to search into the missing dimension. The "in-plane" step is made using the (normalized) difference vector between the parent solution x_i and the centroid of the current population x_C. The orthogonal step is made taking a random vector orthogonal $VOrt_i$ to the parent-centroid difference vector (Figure 1). This two-step process for generating the new trial solutions $trial_i$ is represented in (3). The direction and size of the difference and the orthogonal vectors are determined by the scaling factor F_i and $FOrt_i$, respectively (Bolufé-Röhler and Chen, 2014).

$$trial_i = x_i + F_i * (x_i - x_C) + FOrt_i * VOrt_i \qquad (3)$$

Figure 1. Visualization of MPS search process in three dimensions. Centroid vector and orthogonal step.



To promote diversification, Thresheld convergence forces new solutions to be a minimum *min_step* threshold distance away from their parent solutions. To avoid new solutions from being sampled too far away from the best-found regions, MPS also enforces a maximum search threshold *max_step = 2 * min_step*. To guarantee that the difference vector step does not exceed the maximum allowed threshold distance, the F_i factor is drawn with a uniform distribution from *[-max_step, max_step]*. To ensure that the new solution $trial_i$ falls in the correct *[min_step, max_step]* threshold interval, the $FOrt_i$ factor is selected with a uniform distribution from $[min\_ort_i, max\backslash\_ort_i]$. The min_ort_i and $max\backslash\_ort_i$ values are calculated using equations (4) and (5), respectively. The difference vector $x_i - x_C$ and the orthogonal vector $VOrt_i$ are normalized before scaling. Once the new solutions are created, clamping is performed if necessary, and the best *n* solutions among the parents and offspring survive into the next generation (Bolufé-Röhler and Chen, 2014).

$$min\_ort_i = \sqrt{max(min\_step_i^2 - F_i^2, 0)} \qquad (4)$$

$$max\_ort_i = \sqrt{max(max\_step_i^2 - F_i^2, 0)} \qquad (5)$$

The *min_step* values are updated by a rule similar to that used in previous attempts to control convergence for Differential Evolution (Bolufé-Röhler et al., 2013) and Estimation of

Distribution Algorithms (Tamayo-Vera et al., 2016) in which an initial threshold is selected that then decays over the course of the search process. Equation (6) shows how *min_step* is calculated: α represents a fraction of the main space diagonal, *FEs* is the total available amount of function evaluations, *k* is the number of evaluations used so far, and γ is the parameter that controls the decay rate of the threshold (Chen et al., 2015).

$$\text{min\_step}_i = \alpha * \text{diagonal} * \left(\frac{[\text{FEs} - \text{k}]}{\text{FEs}}\right)^{\gamma} \tag{6}$$

To ensure good spacing in the initial population, the initial points are selected to be on the diagonals of the search space. Assuming that the search space is bounded by the same lower and upper bound in each dimension: s_k is the *k*-th population member, rs_i are random numbers which can be -0.5 or 0.5, and bound is the lower and upper bound in each dimension, see (7). This initialization method leads to a better distribution of the initial solutions in the search space than did uniform random solutions. A detailed pseudo-code is presented in Algorithm 1.

$$s_k = \left(\frac{rs_1 * \text{bound}}{2}, \frac{rs_2 * \text{bound}}{2}, \ldots, \frac{rs_n * \text{bound}}{2}\right) \tag{7}$$

Algorithm 1: Minimum Population Search (MPS)

```
MPS(α, γ, maxFEs)
// Ecuation (7)
X ← InitialPopulation();
while FEs < maxFEs do
    // Ecuation (6)
    min_step ←
     UpdateThreshold(FEs, maxFEs, α, γ);
    max_step ← 2 * min_step;
    x_c ← CalculateCentroid(X);
    for i = 1 : popsize do
        F_i ← UniformRandom(−max_step, max_step);
        // Normalized vector
        VOrt_i ← OrthogonalVector(x_i, x_c);
        // Ecuations (4) and (5)
        FOrt_i ← UniformRandom(min_ort, max_ort);
        // Clamping if necessary
        trial_i ← x_i + F_i * (x_i − x_c) + FOrt_i * VOrt_i;
    end
    X ← BestSolutions(X, trial);
end
```

Given the excellent performance in optimizing a single objective function and the ability to perform a methodical exploration of the search space, it is suggested in (Bolufé-Röhler and Chen, 2014) that MPS can also have a satisfactory performance in multi-objective continuous optimization. In next section, we present an analysis of how to create a Multi-objective MPS algorithm following MPS framework.

MULTI-OBJECTIVE MINIMUM POPULATION SEARCH

In addition to the concepts of mono-objective metaheuristic, a multi-objective metaheuristic contains three main search components (Talbi, 2009):

- Fitness assignment: The main role of this procedure is to guide the search towards Pareto optimal solutions for a better convergence. It assigns a scalar-valued fitness to a vector objective function.
- Diversity preservation: The emphasis here is to generate a diverse set of Pareto solutions in the objective and/or the decision space.
- Elitism: The preservation and use of elite solutions (e.g., Pareto optimal solutions) allows a robust, fast, and a monotonically improving performance of a metaheuristic.

These three components are implicit in the selection process, thus defining a multi-objective selection process for MPS is the key task for adapting MPS to a multi-objective approach. We value two general strategies recommended in (Talbi, 2009).

- Objective function decomposition: This strategy consists in partitioning the original objective function into several sub-objectives. The sub-objectives may be defined over a subset of the decision variables. This decomposition process will separate the potentially conflicting goals of the mono-objective function and will then reduce the number of local optima associated with the problem (Knowles et al., 2001).
- Helper objectives: This strategy consists in adding new objectives. The added objectives are generally correlated with the primary objective function (Talbi, 2009). These new secondary objectives may be viewed as helper objectives whose introduction will reduce the difficulty of the original mono-objective problem. For instance, in a landscape characterized by neutral networks (i.e., plateaus), adding helper objectives may break those plateaus into smooth networks in which the search is more easy for any metaheuristic.

As we intend to create a generic MOEA, all objectives have the same level of importance and it does not exist any primary objective function. For that reasons we apply the strategy objective function decomposition. Subsequently, we proceed to experiment with several multi-objective selection processes to analyze which is best coupled to MPS.

Selection process

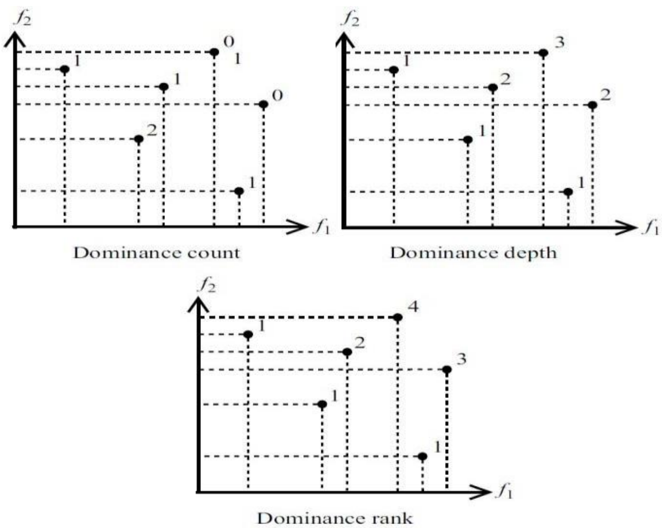
The main element that converts the algorithm into a multi-objective optimizer is the selection process it uses. We evaluate five different multi-objective selection processes for MPS in PISA:

- Dominance count: The dominance count of a solution is related to the number of solutions dominated by the solution. This measure can be used in conjunction with the other ones (Talbi, 2009) (Figure 2).
- Dominance-depth: The population of solutions is decomposed into several fronts. The nondominated solutions of the population receive rank 1 and form the first front E_1. The solutions that are not dominated except by solutions of E_1 receive rank 2; they form the second front E_2. In a general way, a solution receives the row *k* if it is only dominated by individuals of the population belonging to the unit.

$$E_1 \cup E_2 \cup \dots \cup E_{k-1} \tag{8}$$

  Then, the depth of a solution corresponds to the depth of the front to which it belongs. For instance, this strategy is applied by NSGA-II algorithm (Deb et al., 2002) (Figure 2).
- Dominance rank: In this strategy, the rank associated with a solution is related to the number of solutions in the population that dominates the considered solution (Talbi, 2009). In MOEAs, the fitness of a solution is equal to the number of population's solutions that dominate the considered solution, plus one (Figure 2). This strategy is applied by Multi-Objective Genetic Algorithm (MOGA) (Fonseca and Fleming, 1993).
- Dominance count: Dominance count is used in combination with Dominance rank. For instance, this strategy is applied by SPEA2 algorithm (Zitzler et al., 2001). For each member of the archive in SPEA2, a strength value proportional to the number of solutions this member dominates is computed. Then, the fitness of a solution is obtained according to the strength values of the individuals achieved that dominate it.
- Indicator based: This strategy is applied by IBEA algorithm (Zitzler and Kunzli, 2004). The fitness assignment scheme is based on a pairwise comparison of solutions contained in a population by using a binary quality indicator. Indeed, several quality indicators can be used.

Figure 2. Some dominance-based ranking methods.



Avoiding centrality

MPS was originally conceived for mono-objective problems. For that reason, it is designed to achieve a central convergence around the attraction basin of the best found solution. This is achieved through the use of the centroid vector, which is the centroid of all the individuals in the population, as a reference point to generate new solutions (Bolufé-Röhler et al., 2015). We considered that this centrality driven mechanism could become an inconvenient when aiming for diversity in MOPs. We performed experiments replacing the centroid vector by a random individual vector from the same population. A different random vector is chosen in each generation of the algorithm. The computational result section shows the achieved performance with both approaches.

HYBRID APPROACHES USING MULTI-OBJECTIVE MPS

Over the last years, interest in hybrid metaheuristics has risen considerably in the field of optimization. The best results found for many real-life or classical optimization problems are obtained by hybrid algorithms (Talbi, 2009). For that reason, we propose the hypothesis that if we hybridize MMPS with other standard MOEAs of the literature, satisfactory results could be obtained. The MOEAs IBEA (Zitzler and Kunzli, 2004) and NSGA-II (Deb et al., 2002) are selected for this purpose.

The general idea of hybridization is that MMPS initially acts up to $maxEvals * \phi$, where ϕ can be equal to 0.5, then MMPS is allowed to evaluate half of the budget of function evaluations. Thereafter, the second MOEA begins and take as initial population the last generation of MMPS. One of the advantages of MMPS is that it finds good solutions in good regions of the search space with very few generations. Pivot ϕ value could be lower than 0.5.

COMPUTATIONAL RESULTS

In this section we present and analyze all the results. Experimental parameters are presented in Table 1. Our implementation uses α = 0.3 and γ = 3 for thresheld convergence, as suggested in (Bolufé-Röhler and Chen, 2013). For all the experiments, the MOPs used were DTLZ and ZDT family.

Table 1. Experimental paremeters

Parameter	**Value**
Number of runs	30
Population size	100
Decision variables (dimension)	30
Objective functions	2
Number of generations	1000
Number of evaluations	100000
α (MPS core)	0.3
γ (MPS core)	3

Results of the different selection processes

Table 2 and Table 3 present the results of executing MMPS coupled with the five multi-objective selection processes, best performance is shown in bold. Table 2 shows the performance measured by the Epsilon indicator, it can be noticed that the Indicator based selector achieves the best result in nine of the twelve MOPs. Table 3 shows the performance measured by the Hypervolume indicator; in this case the Indicator based selector achieves the best results in all but one of the problems. Results with the R quality indicator values were also analyzed, but are not presented because they are very similar to the Epsilon indicator. The second best selection process is Dominance-depth, it achieves the best result in two MOPs in Table 2 and one of the best results in Table 3. It also consistently achieves the second best results in most of the problems.

Table 2. Epsilon means for MMPS with five selection processes

Problem	D. count	D. depth	D. rank	D. count	I. based
DTLZ1	1.41E-01	1.05E-02	1.00E+00	1.10E-02	**3.04E-03**
DTLZ2	1.21E-01	1.18E-03	7.98E-01	7.49E-04	**7.50E-04**
DTLZ3	1.13E-01	6.59E-03	9.94E-01	7.33E-03	**4.27E-03**
DTLZ4	9.96E-01	7.59E-03	1.00E+00	6.73E-02	**2.87E-03**
DTLZ5	1.21E-01	1.22E-03	8.49E-01	**7.22E-04**	8.03E-04
DTLZ6	3.33E-02	2.63E-02	1.00E+00	2.48E-02	**1.90E-02**
DTLZ7	9.03E-02	2.93E-03	1.00E+00	2.92E-03	**5.35E-03**
ZDT1	1.46E-01	3.58E-03	1.00E+00	3.10E-03	**2.80E-03**
ZDT2	1.00E-01	4.79E-03	1.00E+00	5.35E-03	**3.46E-03**
ZDT3	2.31E-01	**6.78E-03**	1.00E+00	8.86E-03	9.28E-03
ZDT4	1.74E-03	**3.49E-04**	1.00E+00	4.10E-04	3.69E-04
ZDT6	1.06E-01	8.34E-02	1.00E+00	7.89E-02	**3.95E-03**

Conversely, the Dominance-depth selector achieves better diversity in the Pareto Front than the Indicator based selector. Figure 3 illustrates this for the DTLZ5 problem, it can be noticed that when using the Indicator based selector diversity is lost at the border of the PF, while the Dominance-depth selector provides complete cover of the front. Summarizing, MMPS with the Indicator based selector achieves a better performance with all the measures, but the Dominance-depth selector provides more diversity on the PF. This is an expected performance since the Indicator based selector focuses more on quality indicator (Epsilon) than Dominance-depth. In other words, Indicator based is more responsive to the convergence of solutions. On the other hand, Dominance-depth holds both elements simultaneously: convergence and diversity.

In order to statistically to support these results, a non-parametric statistical test was performed. Table 4 shows the results of the Fisher-Indep test for Epsilon indicator on DTLZ6. Statistically significant *p*-values are shown in bold.

Table 3. Hypervolume means for MMPS with five selection processes

Problem	D. count	D. depth	D. rank	D. count	I. based
DTLZ1	1.55E-01	1.47E-03	1.13E+00	1.70E-03	**6.28E-04**
DTLZ2	1.22E-01	1.01E-04	1.06E+00	8.61E-05	**4.43E-05**
DTLZ3	1.25E-01	8.09E-04	1.14E+00	1.09E-03	**6.88E-04**
DTLZ4	1.00E+00	3.13E-03	1.01E+00	3.45E-02	**2.98E-04**
DTLZ5	1.22E-01	1.04E-04	1.07E+00	8.45E-05	**4.46E-05**
DTLZ6	3.37E-02	1.32E-02	1.11E+00	1.00E-02	**9.70E-03**
DTLZ7	4.78E-02	1.76E-03	1.15E+00	1.91E-03	**6.29E-04**
ZDT1	1.12E-01	2.24E-03	1.15E+00	2.01E-03	**1.30E-03**
ZDT2	4.33E-02	2.42E-03	1.13E+00	2.36E-03	**9.18E-04**
ZDT3	1.60E-01	3.43E-03	1.11E+00	4.55E-03	**1.91E-03**
ZDT4	1.33E-03	**1.60E-04**	1.20E+00	2.05E-04	4.40E-04
ZDT6	6.14E-02	4.33E-02	1.14E+00	3.68E-02	**8.47E-04**

The results for this problem are representative of the results achieved on most of the MOPs and allow to conclude that better performance achieved by the Indicator based selector is statistically significantly respect to the performance achieved by the other selectors. The difference between the other selectors is not statistically significant.

Figure 3. Pareto Fronts examples: MMPS-Dominance depth and MMPS-Indicator based.

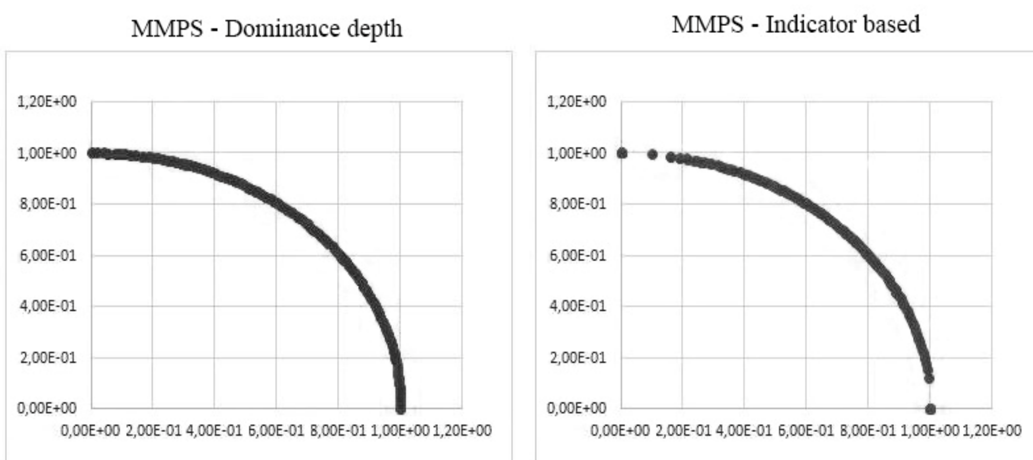


Table 4. Fisher-Indep test for MMPS with five selection processes with Epsilon on DTLZ6 problem

Fisher-Indep test
Dominance-depth is better than Dominance-count with a *p*-value < 3.10571e-032
Dominance-rank is better than Dominance-count with a *p*-value of 1
Indicator based is better than Dominance-count with a *p*-value < 3.10571e-032
Dominance-count is better than Dominance-count with a *p*-value < 3.10571e-032
Dominance-count is better than Dominance-depth with a *p*-value of 1
Dominance-rank is better than Dominance-depth with a *p*-value of 1
Indicator based is better than Dominance-depth with a *p*-value < 3.10571e-032
Dominance-count is better than Dominance-depth with a *p*-value of 0.18546
Dominance-count is better than Dominance-rank with a *p*-value < 3.10571e-032
Dominance-depth is better than Dominance-rank with a *p*-value < 3.10571e-032
Indicator based is better than Dominance-rank with a *p*-value < 3.10571e-032
Dominance-count is better than Dominance-rank with a *p*-value < 3.10571e-032
Dominance-count is better than Indicator based with a *p*-value of 1
Dominance-depth is better than Indicator based with a *p*-value of 0.99998
Dominance-rank is better than Indicator based with a *p*-value of 1
Dominance-count is better than Indicator based with a *p*-value of 0.99926
Dominance-count is better than Dominance-count with a *p*-value of 1
Dominance-depth is better than Dominance-count with a *p*-value of 0.81548
Dominance-rank is better than Dominance-count with a *p*-value of 1
Indicator based is better than Dominance-count with a *p*-value of 0.00068

Centrality results

In order to test how much the use of the centroid affects MMPS two versions were implemented: MMPS-Centroid which uses the original MPS with the centroid vector and MMPS-Random which uses a randomly selected individual from the population to generate the new solutions (as described in previous section). Both algorithms use the Indicator based selector. Table 5 shows a comparison of both algorithms in terms of Epsilon and Hypervolume quality indicators, respectively. R quality indicator values are very similar results to the Epsilon indicator. Best performance is shown in bold.

The results in Table 5 show that MMPS-Centroid outperforms MMPS-Random in both quality indicators. When measured by the Epsilon indicator MMPS-Centroid achieves the best results in 8 of the 12 problems. Results with the Hypervolume indicator favor MMPS-Centroid by 7 vs. 5 problems. In terms of the Pareto Front diversity MMPS-Centroid also shows a more diverse distribution along the front than MMPS-Random. After these results we conclude that MMPS's sampling mechanism based on the centroid vector does not affect its performance in multi-objective optimization.

Table 5. Epsilon and Hypervolume means for MMPS-Centroid and MMPS-Random

Problem	**Epsilon**	**means**	**Hypervolume**	**means**
	MPS-Centroid	**MPS-Random**	**MPS-Centroid**	**MPS-Random**
DTLZ1	3.04E-03	**1.99E-04**	6.28E-04	**5.45E-08**
DTLZ2	**7.50E-04**	7.51E-02	**4.43E-05**	4.36E-02
DTLZ3	4.27E-03	**3.91E-05**	6.88E-04	**4.68E-08**
DTLZ4	**2.87E-03**	1.18E-01	**2.98E-04**	4.29E-02
DTLZ5	**8.03E-04**	7.92E-02	**4.46E-05**	4.54E-02
DTLZ6	**1.90E-02**	3.53E-02	9.70E-03	**3.61E-03**
DTLZ7	**5.35E-03**	3.64E-03	**6.29E-04**	3.43E-03
ZDT1	**2.80E-03**	4.21E-02	**1.30E-03**	3.33E-02
ZDT2	3.46E-03	**3.28E-06**	9.18E-04	**3.57E-06**
ZDT3	**9.28E-03**	1.48E-01	**1.91E-03**	9.92E-02
ZDT4	3.69E-04	**3.67E-04**	4.40E-04	**3.83E-04**
ZDT6	**3.95E-03**	1.11E-01	**8.47E-04**	8.64E-02

Comparisons between MMPS, hybrid MMPS and three MOEAs

Several experiments were performed with the two hybrid variants of MMPS proposed in previous sections (MMPS-IBEA and MMPS-NSGA-II). In these hybrids, when MMPS reaches the maxEvals $*$ ϕ evaluations of the objective functions the threshold value μ is truncated. This value is still relatively large when truncation occurs, for instance in case $\phi = 0.3$ with 1000 total generations, the threshold is still greater than 1. This strategy aims to focus MMPS on exploration and allowing the second MOEA to perform exploitation.

The MMPS-IBEA and MMPS-NSGA-II hybrids were tested with different values of ϕ = 0.3, 0.5, 0.7. Among the hybrids the best results were achieved with MMPS-IBEA and ϕ = 0.3. Table 6 and Table 7 compare the results of MMPS and MMPS-IBEA for the means of the Epsilon and Hypervolume quality indicators, respectively. The R quality indicator was not included because of its similarity to the Epsilon indicator results. For a broader analysis, these tables also include the results achieved by three MOEAs: IBEA, NSGA-II and SPEA2. Best performance is shown in bold.

Table 6. Epsilon means for MMPS, hybrid MMPS-IBEA, IBEA, NSGA-II and SPEA2

Problem	MMPS	MMPS-IBEA	IBEA	NSGA-II	SPEA2
DTLZ1	**3.04E-03**	1.60E-02	3.34E-03	1.65E-02	1.30E-02
DTLZ2	**7.50E-04**	4.11E-03	6.18E-03	6.22E-03	4.24E-03
DTLZ3	4.27E-03	**8.26E-04**	4.37E-03	3.08E-02	1.96E-02
DTLZ4	**2.87E-03**	5.20E-03	3.04E-01	3.05E-01	6.78E-02
DTLZ5	**8.03E-04**	4.03E-03	6.18E-03	6.22E-03	4.24E-03
DTLZ6	1.90E-02	**3.81E-03**	1.87E-02	5.07E-02	4.97E-02
DTLZ7	5.35E-03	1.90E-02	1.83E-02	6.15E-03	**3.95E-03**
ZDT1	**2.80E-03**	4.05E-03	3.92E-03	7.46E-03	4.38E-03
ZDT2	**3.46E-03**	8.48E-03	7.00E-01	4.33E-01	5.75E-02
ZDT3	9.28E-03	1.86E-02	2.39E-02	5.20E-03	**4.20E-03**
ZDT4	**3.69E-04**	4.12E-03	2.18E-01	1.55E-02	1.26E-02
ZDT6	**3.95E-03**	4.68E-03	4.68E-03	1.67E-02	2.00E-02

Table 7. Hypervolume means for MMPS, hybrid MMPS-IBEA, IBEA, NSGA-II and SPEA2

Problem	MMPS	MMPS-IBEA	IBEA	NSGA-II	SPEA2
DTLZ1	6.28E-04	1.16E-02	**7.20E-05**	6.98E-04	4.44E-04
DTLZ2	**4.43E-05**	1.08E-03	2.43E-03	2.45E-03	2.06E-03
DTLZ3	6.88E-04	**1.05E-05**	1.00E-04	2.81E-03	8.40E-04
DTLZ4	**2.98E-04**	1.19E-03	1.01E-01	1.01E-01	3.50E-02
DTLZ5	**4.46E-05**	1.10E-03	2.43E-03	2.45E-03	2.06E-03
DTLZ6	9.70E-03	**3.94E-04**	6.17E-03	2.04E-02	1.97E-02
DTLZ7	**6.29E-04**	2.43E-03	2.16E-03	1.83E-03	1.76E-03
ZDT1	**1.30E-03**	2.45E-03	2.32E-03	3.45E-03	2.89E-03
ZDT2	**9.18E-04**	2.99E-03	3.03E-01	1.87E-01	2.53E-02
ZDT3	1.91E-03	4.10E-03	4.68E-03	**1.33E-03**	1.40E-03
ZDT4	**4.40E-04**	2.46E-03	2.37E-01	1.47E-02	1.19E-02
ZDT6	**8.47E-04**	2.88E-03	2.76E-03	1.74E-02	1.97E-02

For both quality indicators MMPS achieves the best results among all the algorithms in 8 of the 12 problems, while the MMPS-IBEA hybrid is the best in 2 (DTLZ3 and DTLZ6) of the other 4 problems and second best in most problems. These results show that MMPS hybrids can be effectively used for multi-objective optimization.

CONCLUSIONS AND FUTURE WORK

In this paper, we designed a multi-objective MPS algorithm (MMPS) following the MPS framework. The proposed MMPS is the result of testing five different multi-objective selection methods on the original MPS and critically testing the centroid vector mechanism. MMPS was compared with others MOEAs and the achieved results show that MMPS produces satisfactory results on all the benchmark problems considered in this paper. The proposed MMPS algorithm has as its key feature that it achieves solutions with an excellent convergence. Computational results between MMPS, hybrid MMPS and three state of the art MOEAs show that MMPS and the MMPS hybrids are competitive with the state of the art algorithms. Nevertheless, the PFs diversity must be improved in future research. The proposed hybrids take a step forward achieving solutions with a competitive quality in term of convergence and diversity. Moreover, future work should also include and exhaustive study on how the algorithms' parameters should be adjusted depending on the problem's characteristics.

In future work, we are also interested to review others selection processes approaches like criteria-based methods; we also want to parallelize the algorithm to reduce runtimes and to apply MMPS in a practical MOP such as the molecular docking problem. Thereafter, it should be possible to test MMPS optimizing many-objectives optimization problems, for example, adding NSGA-III selection process (Deb and Jain, 2014).

REFERENCES

Bader, J., (2010). Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods. Ph.D. dissertation, ETH Zurich, Switzerland.

Bleuler, S., M. Laumanns, L. Thiele and E. Zitzler (2003). PISA - A platform and programming language independent interface for search algorithms. LectureNotes in Computer Science, C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds. Berlin: Springer, 494–508.

Bolufé-Röhler, A. and S. Chen (2013) Minimum Population Search - Lessons from building a heuristic technique with two population members. IEEE Congress on Evolutionary Computation: 2061–2068.

Bolufé-Röhler, A., S. Estévez-Velarde, A. Piad-Morffis, S. Chen and J. Montgomery (2013). Differential evolution with thresheld convergence. IEEE Congress on Evolutionary Computation: 40–47.

Bolufé-Röhler, A., A. Coto-Santiesteban, M. Rosa-Soto and S. Chen (2014). Minimum Population Search, an Application to Molecular Docking. Revista GECONTEC, 2 (3).

Bolufé-Röhler, A. and S. Chen (2014). Extending Minimum Population Search towards Large Scale Global Optimization. IEEE Congress on Evolutionary Computation: 845–852.

Bolufé-Röhler, A., S. Fiol-Gonzalez and S. Chen (2015). A Minimum Population Search Hybrid for Large Scale Global Optimization. IEEE Congress on Evolutionary Computation: 1958–1965.

S. Chen, J. Montgomery, A. Bolufé-Röhler, and Y. Gonzalez-Fernandez (2015). A Review of Thresheld Convergence. Revista GECONTEC 3(1).

Conover, W. J. (1999). Practical Nonparametric Statistics, 3rd ed. John Wiley & Sons.

Deb, K. (2001) Multi-Objective Optimization using Evolutionary Algorithms. New York, EE.UU.: Wiley.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions Evolutionary Computation 6: 182–197.

Deb, K., L. Thiele, M. Laumanns, and Zitzler, E. (2005) Scalable test problems for evolutionary multi-objective optimization. Evolutionary Multi-objective Optimization. Springer, 105–145.

Deb, K. (2011) Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction. Indian Institute of Technology Kanpur, India, Tech.Rep.

Deb, K., and H. Jain (2014). An Evolutionary Many-Objective OptimizationAlgorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with

Box Constraints. IEEE Transactions on Evolutionary Computation, 18: 577–601.

Fonseca C. M. and P. J. Fleming (1993). Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. Proceedings of the 5th International Conference on Genetic Algorithms. California, USA: Morgan Kaufmann, 416–423.

Glorieux E., B. Svensson, F. Danielsson, and B. Lennartson (2017). Constructive cooperative coevolution for large-scale global optimisation. Journal of Heuristics, 23(6): 449–469.

Hansen M. P. and A. Jaszkiewicz (1999). Evaluating the Quality of Approximations to the Non-Dominated Set. Technical University of Denmark, Tech. Rep.

Knowles J. D., R. A. Watson, and D. W. Corne (2001). Reducing local optima in single objective problems by multi-objectivization. 1st International Conference on Evolutionary Multi-Criterion Optimization. Zurich, Switzerland: Springer, 269–283.

Piad-Morffis A., S. Estévez-Velarde, A. Bolufé-Röhler, J. Montgomery and S. Chen (2015). Evolution strategies with thresheld convergence. Evolutionary Computation (CEC), IEEE Congress: 2097–2104.

Talbi G. (2009). Metaheuristics: From design to implementation, 1st ed. John Wiley & Sons.

Tamayo-Vera, D., A. Bolufé-Röhler, and S. Chen (2016). Estimation multi-variate normal algorithm with thresheld convergence. Evolutionary Computation (CEC), IEEE Congress: 3425–3432.

Zitzler, E. and L. Thiele (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation, 3: 257–271.

Zitzler, E., M. Laumanns, and L. Thiele (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Tech. Rep.

Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca (2003). Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation, 7: 117–132.

Zitzler, E. K. Deb and L. Thiele (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, 8: 173–195.

Zitzler, E. and S. Kunzli (2004). Indicator-based selection in multi-objective search. Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), 832–842.

Editor in Chief: Prof. Dr. Luis Camilo Ortigueira-Sánchez