

---

# MOCS: Multi-Objective Clustering Selection Evolutionary Algorithm

---

Thomas E. Koch and Andreas Zell  
University of Tübingen, WSI/RA  
Sand 1, D-72076 Tübingen, Germany  
{koch,zell}@informatik.uni-tuebingen.de

## Abstract

In this paper, we describe a multi-objective evolutionary algorithm, that uses clustering selection and does not need any additional parameter like others. It clusters the population into a flexible number of clusters employing x-means from [Pelleg and Moore, 2000]. First, the selective fitness is assigned to clusters and in second place to individuals of clusters. We show three hybrid variants incorporating additional mechanisms from other elitist multi-objective evolutionary algorithms in order to increase selection pressure. Using the test functions from Deb’s T suite (T1-T6), from Schaffer, Kursawe and Quagliarella we evaluate the performance and the quality of our approach against the most recent and performant elitist multi-objective evolutionary algorithms, NSGA2, SPEA2 and PESA2. The comparison yields promising results for region-based selection using clustering in combination with additional crowding strategies.

## 1 Introduction

Multi-Objective Optimisation Problems (MOOP) arise when at least two competing objectives (or criteria) have to be optimised. If no preferences for any objectives are given or known a priori, the task is to optimise all objectives at the same time, which will produce a set of optimal trade-off solutions rather than one single optimal solution. In general a MOOP is defined by a function  $f$

$$(y_1, y_2, \dots, y_m) = f(x_1, x_2, \dots, x_n) \text{ with } m > 1, n > 0$$

which maps a vector of  $n$  decision variables to a vector of  $m$  objective variables, which has to be optimised.

Solutions may be better, equal or worse than others. Better or *dominating* solutions are meant to be better in at least one objective and not worse in all others. All non-dominated solutions are called *Pareto-optimal* and belong to the *Pareto set*.

Most Evolutionary Algorithms (EA) try to solve MOOPs by using a priori knowledge —given or not— to weight the different objectives in order to construct a single objective problem. But these approaches produce just one solution of the Pareto-optimal set. In contrast, Evolutionary Multi-Objective Optimisation Algorithms (EMO) try to find the whole Pareto set or at least a good presentation of it. For further reading in MOOP and EMO we suggest the comprehensive book from Deb [Deb, 2001].

Since the first real EMO algorithm by Schaffer [Schaffer, 1984], called VEGA, and the inspiring lines in Goldberg’s book [Goldberg, 1989], a number of seminal approaches have shown the capability of EMOs to demonstrate that Pareto domination-based EMOs can be reliably used to find and maintain multiple trade-off solutions of the Pareto set. In the last years elitist EMOs have shown best performance in order to find global Pareto-optimal solutions and good diversity in presenting the real global Pareto set: *Non-dominated Sorting Genetic Algorithm 2* (NSGA2) [Deb et al., 2000], *Strength Pareto Evolutionary Algorithm 2* (SPEA2) [Zitzler et al., 2001] and *Pareto Envelope based Selection* (PESA) [Corne et al., 2000] and PESA2 [Corne et al., 2001]. All these algorithms use the framework of conventional EAs and differ in fitness assignment, selection operator and an optional external archive, storing the actual Pareto set.

In the following of the paper we show a new Multi-Objective Clustering Selection operator (section 2), compare it to the mentioned algorithms on commonly used test problems (section 3), discuss the results (sec-

tion 4), and conclude (section 5).

## 2 Multi-Objective Clustering Selection (MOCS)

The Multi-Objective Clustering Selection (MOCS) Evolutionary Algorithm works as follows:

**Algorithm 1 (MOCS EA)** Population  $P_t$  of  $\lambda \leq N$  individuals is evolved for  $T$  generations. An additional archive population  $P_t^*$  of size  $\mu = N$  is maintained.  $P_T^*$  gives the result of non-dominated solutions.

1. *Initialisation*: Set  $t = 0$ , generate initial offspring population  $P_t$ , set population  $P_t^* = \emptyset$ .
2. *Evaluate and Assign Fitness*: Evaluate fitness values of individuals of  $P_t$ .
3. *Environmental Selection*: Use a truncation operator in order to reduce the size of  $P_t^* = P_t^* \cup P_t$  to  $N$  (cf. section 2.1): We use “non-dominated Pareto sorting” and “crowding distance measure” from NSGA2.
4. *Termination*: If  $t \geq T$  then remove all dominated individuals from  $P_T$  and stop.
5. *Mating Selection*: **Cluster**  $P_t^*$  into  $k$  clusters using  $(c_1, c_2, \dots, c_k) = x\text{-means}(P_t^*, 1, N)$  with  $1 \leq k \leq N$  [Pelleg and Moore, 2000] (cf. section 2.2).
  - (a) *Region-based selection*: Perform binary tournament selection on the  $k$  clusters found where clusters with lower cardinality win (like PESA does)  $\Rightarrow \lambda$  selected clusters.
  - (b) *Local individual-based selection*: For any of the  $\lambda$  selected clusters perform a binary tournament selection on its individuals where the crowding-distance measure from NSGA2 orders individuals  $\Rightarrow \lambda$  selected individuals.

The mating pool  $P_t$  contains now  $\lambda$  individuals.

6. *Variation*: Apply recombination and mutation to  $P_t$ .
7. *Increment*: Set  $P_{t+1}^* = P_t^*$  and  $P_{t+1} = P_t$ . Set  $t = t + 1$  and go to 2.

As alternatives step 5a is replaced with random selection of clusters and step 5b is replaced with random selection of individuals (PESA-like). Table 1 gives an overview of our implemented combinations.

MOCS uses the basic Genetic Algorithm extended by *Environmental Selection* and *Mating Selection* like

other elitist EMOs do. The archive is limited by  $N$  entries and has to be reduced in the *Environmental Selection* step, because the offspring population has to be merged into the archive, which yields a maximum size of  $2N$ . It is important to keep as many non-dominated solutions as possible and a good diversity among them in the archive. In the *Mating Selection* step individuals are chosen for the mating pool. Again the best and most diverse solutions should be selected with higher probability in order to increase selection pressure for finding better solutions in the variation step.

Besides its own clustering technique, MOCS uses ordering techniques from other EMOs (cf. table 1): PESA2 [Corne et al., 2001] has a “squeeze”-factor, which simply counts the individuals of each hyperbox (see also section 2.2). We use this in a similar way: In a binary tournament the cluster with less members wins. From NSGA2 [Deb et al., 2000] we borrow the “non-dominated Pareto sorting” and the “crowding distance measure” which orders individuals firstly by its Pareto-rank (lower is better) and secondly by the volume enclosed by its next neighbours (larger is better). In NSGA2 it is used for environmental and mating selection.

Algorithm	Mating Selection (after clustering with X-means)
MOCS-1	randomly select clusters, then perform binary tournament with crowding-distance measure
MOCS-2	perform binary tournament on clusters with “squeeze”-factor, then perform binary tournament with crowding-distance measure
MOCS-3	perform binary tournament on clusters with “squeeze”-factor, then randomly select individuals from winning clusters

Table 1: Overview of implemented versions: MOCS uses the crowding-distance measure from NSGA2 and the “squeeze”-factor from PESA2 in order to rank individuals resp. clusters.

### 2.1 Environmental Selection

In a first step we remove all multiple points and keep just one of each. This may reduce the population (archive) size below  $N$  but gives all points the same chance in the later steps of the algorithm. MOCS uses Environmental Selection from NSGA2 in order to reduce the population (archive) size. We also implemented and tested a clustering technique for En-

environmental Selection similar to the NSGA2, but we did not include it in this paper: First do a non-dominated Pareto ranking, which assigns all individuals to  $r$  Pareto fronts; then successively add the ranked fronts of individuals until the actual front does not fit into the remaining  $s$  of  $N$  slots. For this actual front first keep all boundary points  $b$  if possible then perform a clustering into  $s - b$  clusters and take the best individual from each cluster into the population. Best individuals are those who have the maximum average distance to all others. Remaining fronts are discarded.

## 2.2 Mating Selection

Mating selection in MOCS is done region-based. Corne and Knowles introduced this technique in EMOs with PESA2 [Corne et al., 2001]. They showed that region-based selection has advantages over individual-based selection like all other elitist EMOs do: the probability of selecting highly isolated individuals in contrast to crowded individuals rises because the unit of selection is now the “region” and no longer the individual. Thus this technique removes selective attention from crowded regions and assigns this attention more equally over the whole population.

The drawback on PESA2 is its ‘hypergrid’. The user has to provide a parameter called grid-size  $g$  in order to build  $g^m$  hyperboxes in the  $m$ -dimensional objective hyperspace. The difficulty is to choose the best dimensions of the hyperboxes so that the resulting hyperboxes or regions are neither too fine-grained nor too large. The first extreme leads back to individual-based selection and the second extreme could lead to one hyperbox containing all individuals. Big advantages of PESA2 and the hypergrid strategy are its low performance complexity of  $O(mN)$  to find the hyperbox for every individual per generation and the easy implementation of the algorithm.

MOCS uses a more flexible but also more expensive technique: we cluster the population into  $k$  clusters, where  $k$  is determined by the algorithm itself.  $k$  may be in the range from 1 to population size  $N$ . The used clustering algorithm *x-means* was introduced by [Pelleg and Moore, 2000]. X-means extends and improves *k-means* [Duda and Hart, 1973], which clusters a dataset in  $k$  clusters. For detailed explanations of x-means and k-means we refer to the cited papers, figure 1 shows an example.

Related work has been done by [Molyneaux et al., 2001]: They introduced the Clustering Pareto Evolutionary Algorithm (CPEA), which finds and retains many local Pareto-optimal fronts in contrast to our global Pareto-optimal algorithm. Ma-

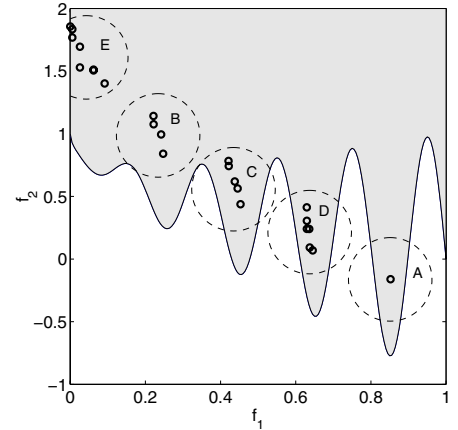


Figure 1: Population of 25 individuals in an EA run for test function T3 (cf. table 2, minimise both,  $f_1$  and  $f_2$ ): 5 clusters (A-E) have been calculated. In the Mating Selection of this population the cluster A down right has the best chances in the tournament selection, because it has the lowest number of elements.

major drawbacks of CPEA are the unlimited number of non-dominated (local) individuals, the fixed number  $k$  of clusters, the local non-dominated sorting (which obviously may be an advantage in the sense of searching for local Pareto-optimal sets) and its poor computational performance due to the unlimited size of the archive.

## 2.3 Complexity Issues

The most complex task in multi-objective optimisation is to find an adequate process to calculate the selective fitness in the sense of crowding in objective space. Isolated individuals must have a higher selective fitness than crowded ones. This selective fitness is used for environment and mating selection to increase selective pressure.

PESA2 has a complexity of  $O(mN)$  to calculate the box of  $N$  individuals and the “squeezes” of the boxes in an  $m$ -dimensional problem per generation. NSGA2 and SPEA2 require  $O(mN^2)$  time.

Simple *k-means* has a performance complexity of  $O(mNk)$  to calculate all distances between the  $N$  individuals and the  $k$  chosen centroids per internal iteration, where the number of internal iterations is not known and may be infinite. X-means is approximately equivalent to running k-means with  $k = 1, 2, 3, \dots, N$  and uses kd-trees to store data which are much more efficient than the naive algorithm (for our low dimensionality). But anyway this yields a complexity of  $O(mN^2)$

Name	Domain	Chromosome Length $L$	Functions
[Zitzler et al., 2000] [Deb, 2001]			
T1-T6 common frame			$f_1(x)$ $f_2(x) = g(x)h(f_1(x), g(x))$
T1	$[0, 1]^n$ $n = 30$	900	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - \sqrt{f_1/g}$
T2	$[0, 1]^n$ $n = 30$	900	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - (f_1/g)^2$
T3	$[0, 1]^n$ $n = 30$	900	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1)$
T4	$x_1 \in [0, 1]$ $x_{i \neq 1} \in [-5, 5]$ $n = 30$	300	$f_1(x) = x_1$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$ $h(f_1, g) = 1 - \sqrt{f_1/g}$
T5	$x_1 \in \{0, 1\}^{30}$ $x_{i \neq 1} \in \{0, 1\}^5$ $n = 11$	80	$u(x_i)$ denotes the number of 1s in $x_i$ $f_1(x) = 1 + u(x_1)$ , $g(x) = \sum_{i=2}^n v(u(x_i))$ , $h(f_1, g) = 1/f_1$ $v(u(x_i)) = 2 + u(x_i)$ if $u(x_i) < 5$ else 1
T6	$[0, 1]^n$ $n = 10$	300	$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $g(x) = 1 + (n-1) [(\sum_{i=2}^n x_i)/(n-1)]^{0.25}$ $h(f_1, g) = 1 - (f_1/g)^2$
[Schaffer, 1985]			
SPH- $m$ $m = 2, 3, 4$	$[-10^3, 10^3]^n$ $n = 20$	400	$f_j(x) = (x_j - 1)^2 + \sum_{1 \leq i \leq n, i \neq j} x_i^2$ $1 \leq j \leq m$
[Quagliarella and Vicini, 1997]			
QV	$[-5, 5]^n$ $n = 20$	400	$f_1(x) = (\frac{1}{n} \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10))^{0.25}$ $f_2(x) = (\frac{1}{n} \sum_{i=1}^n ((x_i - 1.5)^2 - 10 \cos(2\pi(x_i - 1.5)) + 10))^{0.25}$
[Kursawe, 1991]			
KUR	$[-10^3, 10^3]^n$ $n = 20$	400	$f_1(x) = \sum_{i=1}^{n-1} (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2}))$ $f_2(x) = \sum_{i=1}^n ( x_i ^{0.8} + 5 \sin^3(x_i))$

Table 2: Test functions for performance comparisons. Deb’s T test suite, QV and KUR are 2-dimensional, SPH- $m$  is evaluated in 2, 3 and 4 dimensions.

in best case.

### 3 Experiments

The MOCS approaches have been tested against NSGA2, SPEA2 and PESA2<sup>1</sup>. Table 2 shows the test functions with domains of decision variables, dimension of variable space  $n$ , dimension of objective space  $m$  and chromosome length  $L$ . All of them are minimisation problems. We used the same functions T1-T6 like [Corne et al., 2001] did for testing PESA and PESA2, like [Zitzler et al., 1999] did for testing SPEA

<sup>1</sup>We also tested against PESA and SPEA, but their performance was too poor, which has also been proven by the evaluation of its successors.

and 8 other EMOs and KUR, QV and SPH- $m$  like [Zitzler et al., 2001] did for testing SPEA2. But also many other researchers have used these functions as well.

Deb [Zitzler et al., 2000, Deb, 2001] provides a procedure of constructing two-dimensional objective problems with a range of characteristics of varying degrees. These include convexity (T1), concavity (T2), discontinuity (T3,T5), multi-modality (T4), deception (T5) and non-uniformity (T6) at the Pareto front. All these problems have to be managed in multi-objective real world problems by optimisers. T1 and T2 are the baseline tests. T1 is convex and thus a simple hill-climber would do the best job. T2 is concave, which yields first difficulties to overcome. T3 has a number (in

this case 5) of disconnected Pareto-optimal fronts. T4 has a convex Pareto-optimal global front but furthermore there exist  $8 \cdot (10^{11})$  local fronts which produce a large number of hurdles. T5 is a boolean function over bit strings, has again many local fronts and attempts to deceive in order to lead the algorithm to a local instead to the global front. T6 is non-convex, non-uniform and the density to the global front is thin. We ran 2000 generations, which yields a total of 200K fitness evaluations in place of 5K fitness evaluations in other studies. We think looking at the whole optimisation process gives more insight into the behaviour of the algorithms. We tested all the algorithms with the same parameter settings [Corne et al., 2001](PESA2) and [Zitzler et al., 1999](SPEA) did for testing their algorithms.

SPH-m is a multi-objective generalisation of the sphere model [Schaffer, 1985], which is a symmetric unimodal function where the isosurfaces are given by hyperspheres. We used versions with two, three and four objectives. QV [Quagliarella and Vicini, 1997] consists of two multi-modal functions, an extreme concave Pareto-optimal front and a diminishing density of solutions towards the Pareto front. KUR [Kursawe, 1991] consists of a multi-modal function and function with pair-wise interactions among the variables, the Pareto front is disconnected consisting of concave and convex parts and an isolated point. Again we used the same parameter settings [Zitzler et al., 2001] did for testing SPEA2.

All algorithms are implemented in Matlab, embedded in a binary-coded Genetic Algorithm framework, but Evolution Strategies or real-coded Genetic Algorithms may be used as well. Table 3 shows the parameter setting for the Genetic Algorithm. For each algorithm and each problem, 20 runs with different random seeds have been evaluated and per run 200 intermediate results over time have been measured.

For measuring the quality of the results we have employed the hypervolume approach. The hypervolume approach by [Zitzler et al., 1999] (modified in [Zitzler et al., 2001]) calculates the portion of the normalised non-dominated hyperspace in a constructed hyperspace. Zitzler e.a. state it as the most appropriate scalar indicator since it combines both the distance of solutions (towards some utopian trade-off surface) and the spread of solutions.

## 4 Results and Discussion

Figures 3 (T1-T6), 2 (KUR,QV) and 4 (SPH-m) show the results of all runs over time. All algorithms reach

Population sizes ( $\mu = \text{archive}$ )	T1-T6: $\mu = 100, \lambda = 10$ KUR, QV, SPH-m: $\mu = \lambda = 100$
#Generations	T1-T3,T6: 20000, T4,T5: 4000 KUR, QV, SPH-m: 10000
Crossover method	uniform
Crossover rate $p_c$	0.7
Mutation rate $p_m$	$1/L$ , where $L$ (bit-wise)
Additional parameters	32x32 hypergrid used in PESA, PESA2

Table 3: Parameter settings for Genetic Algorithm

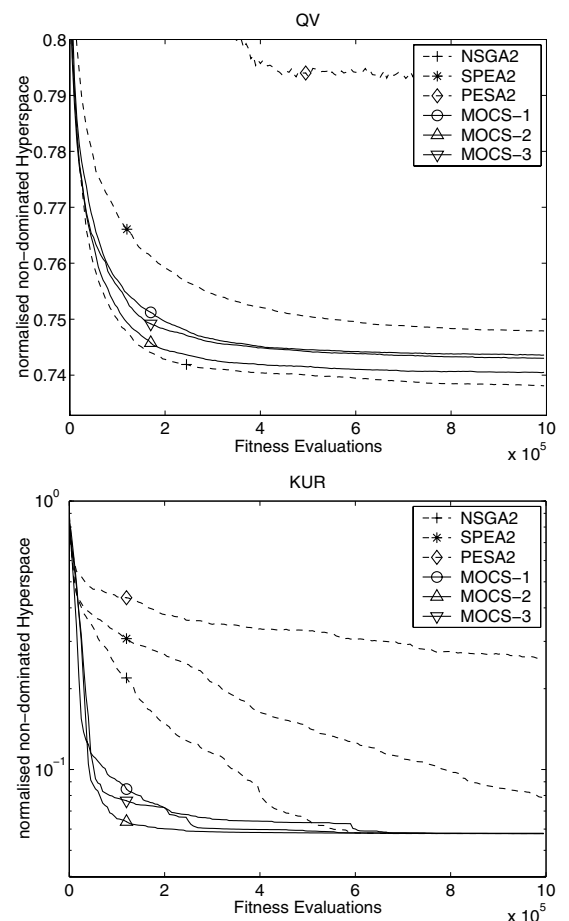


Figure 2: Performance values for QV and KUR. The graphs show the average normalised non-dominated portion of the hyperspace for 20 runs.

the Pareto fronts for the T functions except for T5, where they are deceived and get stuck. On T1-T6 all algorithms except PESA and PESA2 perform quite well and show similar curves. PESA and PESA2 lack the ability of keeping boundary points. They also do badly after reaching the global Pareto-optimal front, where it is crucial to spread uniformly over the front

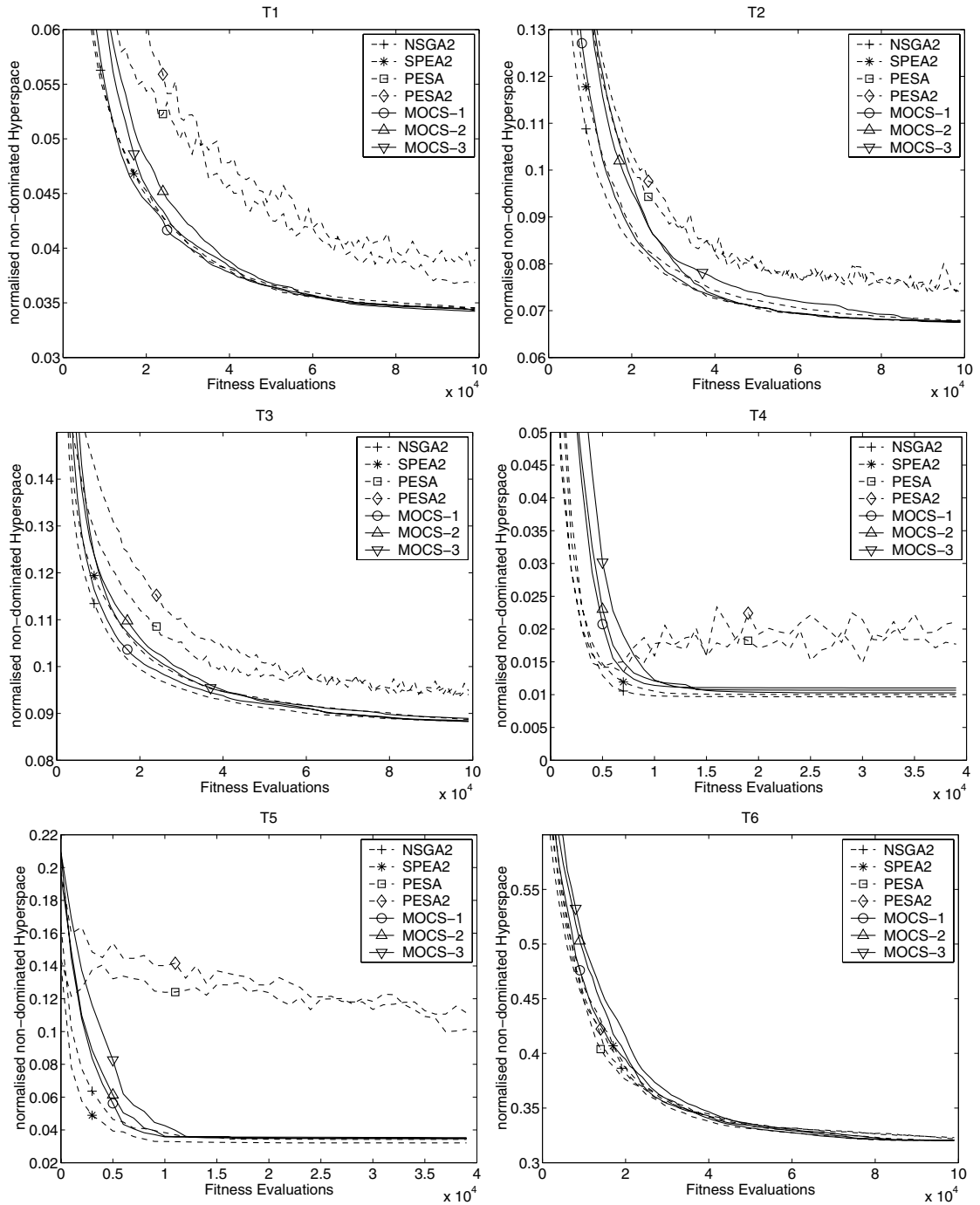


Figure 3: Performance values for T1-T6. The graphs show the average normalised non-dominated portion of the hyperspace for 20 runs.

in order to lessen the non-dominated hyperspace. The results for the T suite also shows that NSGA2, SPEA2 and our approaches are better than PESA and PESA2. On the T suite test functions our approaches keep track with NSGA2 and SPEA2.

On the QV, KUR and SPH-2 functions also all Pareto fronts are reached by MOCS. Again PESA2 performs worst and stagnates because boundary points are not kept. On QV NSGA2 is slightly better than our approaches. On KUR our approaches all perform quite

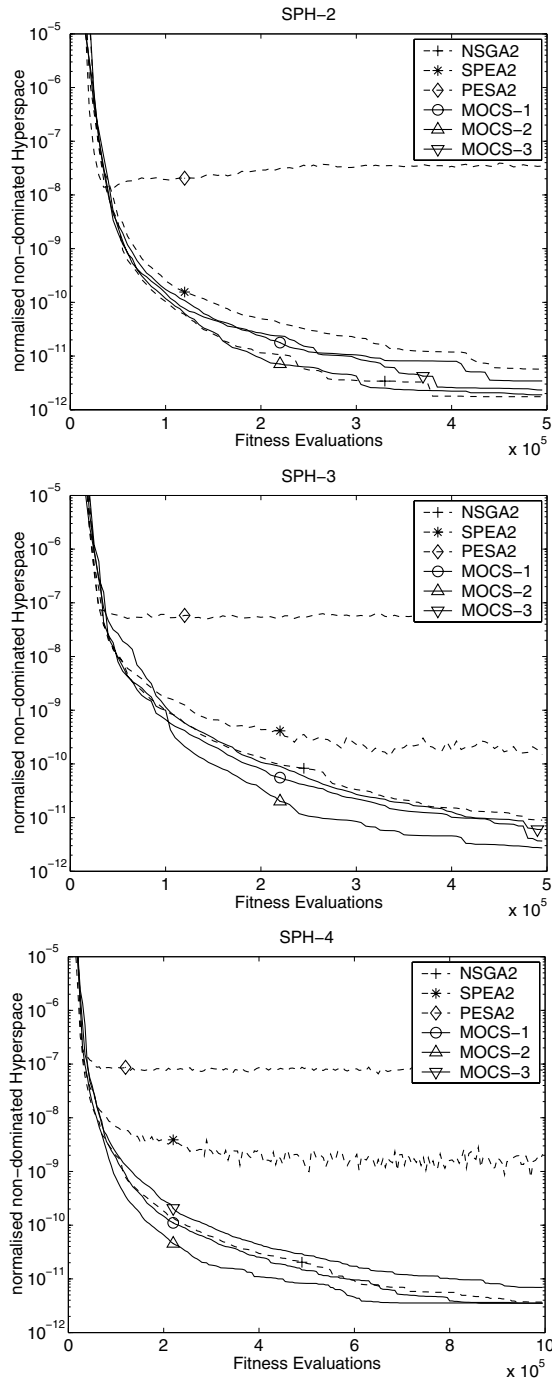


Figure 4: Performance values for SPH-[2,3,4]. The graphs show the average normalised non-dominated portion of the hyperspace for 20 runs.

better than all others. This is due to the fact that our first fronts are always ahead the others before reaching the global Pareto front, which is a quite small region in a huge objective space. On SPH- $m$  our MOCS ap-

	NSGA2	SPEA2	PESA2	MOCS-1,2,3
QV	37	55	51	75 157 45
KUR	37	45	51	74 156 47
SPH-2	44	71	53	74 141 47
SPH-3	45	73	76	47 58 42
SPH-4	46	77	55	87 58 46

Table 4: Average time in minutes on 20 runs on the same processors.

proaches are getting better with increasing dimension  $m$  and outperform all others.

MOCS-2 seems to be the best of our approaches: It uses the NSGA2 Environmental Selection, for Mating Selection it first clusters the population with the  $x$ -means technique into the best quantity of clusters then it uses binary tournament region-based selection with the “squeeze”-factor and lastly for every chosen cluster it performs a binary tournament selection with the NSGA2-crowding distance measure. This last binary tournament is absent in PESA1 and PESA2. They use just random selection to choose individuals from a hyperbox, which may explain their bad performance.

Table 3 shows the average runtime in minutes of the algorithms. Just MOCS-2 needs factor 3 more time compared to NSGA2 on some problems. This shows experimentally that the time complexity of MOCS holds  $O(mN^2)$  as stated in section 2.3.

## 5 Conclusion

We described a region-based selection technique in our Multi-Objective Clustering Selection EA, called MOCS. The advantage of MOCS is its automated clustering which clusters the individuals of the population in a very flexible way. In contrast to any hypergrid strategy this prevents choosing the wrong grid size, which leads to too large or too small hyperboxes. Additionally, after the region-based selection step, MOCS uses another binary tournament inside the clusters to increase the selection pressure. Here we used the technique from NSGA2 to select individuals inside a cluster.

MOCS has the same or better performance compared to NSGA2, SPEA2 and PESA2. Furthermore it shows better performance with higher dimensional problems, which we will evaluate on real world problems like the calibration process of combustion engines or in manufacturing industries [Koch et al., 1999, Koch et al., 2001].

Thus we showed that clustering region-based selection with an additional local individual-based selection is a promising alternative to existing methods. For further investigation we want to incorporate other strategies like SPEA2 into our hybrid framework. Also the behaviour of the clustering technique needs to be investigated to get a deeper knowledge of how MOCS works.

## References

- [Corne et al., 2001] Corne, D. W., Jerram, N. R., Knowles, J. D., and Oates, M. J. (2001). PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In Spector, L., Goodman, E. D., Wu, A., Langdon, W., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. H., and Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 283–290, San Francisco, California. Morgan Kaufmann Publishers.
- [Corne et al., 2000] Corne, D. W., Knowles, J. D., and Oates, M. J. (2000). The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H.-P., editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848, Paris, France. Springer. Lecture Notes in Computer Science No. 1917.
- [Deb, 2001] Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley.
- [Deb et al., 2000] Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H.-P., editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France. Springer. Lecture Notes in Computer Science No. 1917.
- [Duda and Hart, 1973] Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Wiley Sons.
- [Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts.
- [Koch et al., 1999] Koch, T. E., Müller, R., Schneider, F., and Zell, A. (1999). Positionierung von Spannmitteln zur Werkstückfixierung mittels Evolutionärer Algorithmen. *Automatisierungstechnische Praxis atp*, pages 32–39.
- [Koch et al., 2001] Koch, T. E., Schneider, F., and Zell, A. (2001). Optimal positioning of clamps for workpiece adjustment using multi-objective evolutionary computation. In Parmee, I. C. and Hajela, H. C., editors, *Proceedings of Optimisation in Industries III*, page in press. Springer, London.
- [Kursawe, 1991] Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In Schwefel, H. P. and Männer, R., editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197, Berlin, Germany. Springer-Verlag.
- [Molyneaux et al., 2001] Molyneaux, A., Leyland, G., and D.Favrat (2001). A New, Clustering Evolutionary Multi-Objective Optimisation Technique. In *Proceedings of the Third International Symposium on Adaptive Systems—Evolutionary Computation and Probabilistic Graphical Models*, pages 41–47, Havana, Cuba. Institute of Cybernetics, Mathematics and Physics.
- [Pelleg and Moore, 2000] Pelleg, D. and Moore, A. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Seventeenth International Conference on Machine Learning ICML*, Stanford.
- [Quagliarella and Vicini, 1997] Quagliarella, D. and Vicini, A. (1997). Coupling Genetic Algorithms and Gradient Based Optimization Techniques. In Quagliarella, D., Périaux, J., Poloni, C., and Winter, G., editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, chapter 14, pages 289–309. John Wiley and Sons, West Sussex, England.
- [Schaffer, 1984] Schaffer, J. D. (1984). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University.
- [Schaffer, 1985] Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum.
- [Zitzler et al., 1999] Zitzler, E., Deb, K., and Thiele, L. (1999). Comparison of Multiobjective Evolutionary Algorithms on Test Functions of Different Difficulty. In Wu, A. S., editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program*, pages 121–122, Orlando, Florida.
- [Zitzler et al., 2000] Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195.
- [Zitzler et al., 2001] Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.