Particle Swarm Optimizers for Pareto Optimization with Enhanced Archiving Techniques

Thomas Bartz–Beielstein, Philipp Limbourg, Jörn Mehnen and Karlheinz Schmitt
Universtität Dortmund
D-44221 Dortmund, Germany
{Thomas.Beielstein,Philipp.Limbourg, Karlheinz.Schmitt}@udo.edu
mehnen@isf.mb.uni-dortmund.de

Konstantinos E. Parsopoulos, and Michael N. Vrahatis
University of Patras
Greece
{kostasp, vrahatis}@math.upatras.gr

Abstract- During the last decades, numerous heuristic search methods for solving multi-objective optimization problems have been developed. Population oriented approaches such as evolutionary algorithms and particle swarm optimization can be distinguished into the class of archive–based algorithms and algorithms without archive. While the latter may lose the best solutions found so far, archive based algorithms keep track of these solutions. In this article a new particle swarm optimization technique, called DOPS, for multi-objective optimization problems is proposed. DOPS integrates well–known archiving techniques from evolutionary algorithms into particle swarm optimization. Modifications and extensions of the archiving techniques are empirically analyzed and several test functions are used to illustrate the usability of the proposed approach. A statistical analysis of the obtained results is presented. The article concludes with a discussion of the obtained results as well as ideas for further research.

1 Introduction

Evolutionary algorithms (EA) have proved very efficient in solving multi-objective optimization (MO) problems. In [Coe99, Deb01], surveys of the existing variants and techniques for EAs are provided. Nearly all the considered algorithms are able to find a good approximations of the true Pareto front but the distribution of the obtained solutions differs among different approaches [DAPM00, KC99].

In the last years, several new nature–inspired algorithms have been developed [BDT99, CDG99, KE95]. Particle swarm optimization (PSO) mimics the social behavior of a flocks of birds in order to guide swarms of particles towards the most promising regions of the search space. PSO has proved very efficient in solving a plethora of problems in science and engineering [Abi02, CH01, LPV02, PV02a]. However, the MO case has not been studied thoroughly, and only a few developments are reported in the literature [CS02, Hu02, PV02b].

This article introduces an extension of PSO for MO problems, called DOPS. Starting from the idea of introducing elitism (archiving) into PSO, different methods of selecting and deleting particles from the archive are analyzed to generate a satisfactory approximation of the Pareto front with evenly distributed solutions along it.

The rest of this article is organized as follows: in Section 2, an overview over existing archiving techniques is given, followed by a description of the basic principles of the proposed technique. In Section 3, the implementation details are described briefly, while in Section 4 the different comparison methods are sketched. In Section 5, preliminary results are reported, and the paper closes with some concluding remarks and future research directions in Section 6.

2 Archiving Techniques

Archiving is the process of maintaining the best (non–dominated) solutions found during the course of a MO algorithm. Several archiving schemes are proposed in the literature, exhibiting different behavior and possessing different theoretical properties [JOS01, KC03, LTZD02, RA00]. A crucial characteristic which is often used for the classification of the archiving schemes in categories is the size of the archive which is used to maintain the solutions.

In simple MO problems, unbounded archives seem an acceptable choice. In such schemes, a new vector is accepted into the archive if it is non–dominated by all solutions already stored in it. All members of the archive that are dominated by the new vector are removed. This procedure is proved to converge towards the true Pareto front under certain assumptions [KC03].

As EA are applied in more difficult problems which consist of a multitude of objective functions, bounding the archive size while maintaining a satisfactory representation of all non–dominated solutions found, becomes very important. Moreover, archiving schemes with guaranteed convergence to a set of solutions that approximates the true Pareto front are highly desirable, although such schemes usually involve sets of parameters whose initial configuration is critical for the algorithm's efficiency [LTZD02]. A common bounded archiving scheme consists of an archive with a

bounding size. As soon as the number of solutions in the archive reaches this bounding size, new vectors are accepted only if they dominate some of the already stored vectors. Thus, the archive size cannot exceed its bound. Archiving algorithms which are similar to the aforementioned procedure have been proposed and their convergence properties have been analyzed [KC03, RA00]. However, such schemes may not provide satisfactorily distributed solutions, leaving some parts of the Pareto front unreachable.

The deficiencies of the bounded archiving techniques described above may be overcome by allowing non–dominated vectors to be replaced by new vectors which may improve the distribution of the archive. Unfortunately, the required computational burden is usually very high for problems with more than a few objective functions [KC03]. Several different techniques, such as the adaptive grid archiving algorithm [KC03], have been recently proposed with promising results. It is a common belief that the development of efficient archiving techniques is still an appealing topic and the increasing use of parallel MO evolutionary algorithms (MOEAs) in real life applications is expected to stimulate research towards this direction [VVZL03].

In the following we will demonstrate how archiving techniques for multi-objective evolutionary algorithms (MOEAs) can be transferred to PSO.

3 Implementation Details

3.1 Basic Concepts of KEA and particle swarm optimization

The software package KEA (Kit for Evolutionary Algorithms), a freely available software package for MOEAs and related MO algorithms, was used to develop a new multi–objective PSO algorithm, called DOPS [BBSM$^+$03]. A sample screen–shot is depicted in Fig. 1. KEA was developed during a one–year student project course at the Department of Computer Science at the University of Dortmund. This software package is under further development and has been successfully applied in research and in complex real–world optimization problems [MMBS03].

DOPS belongs to the class of multi–objective PSO algorithms as presented in [CS02]. PSO was initially proposed by Kennedy and Eberhart [KE95]. It is a population based optimization method inspired by the behavior of bird flocks, which employs a swarm of particles to probe the search space. A particle of the swarm is completely described by three main concepts: its current position in the search space, a memory of its best previous position and an information regarding the best position ever attained by a topological neighborhood of it. In the *global* PSO variant, the whole swarm is considered as the neighborhood of each

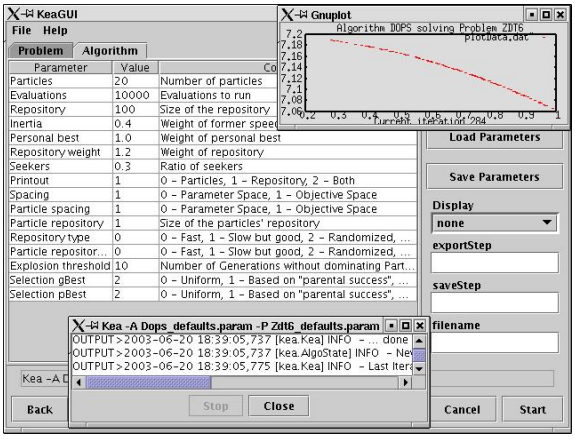

Figure 1: KEA - Kit for Evolutionary Algorithms

particle, while in the *local* variant the neighborhood usually consists of just a few particles. Let $x_i(t)$ be the current position of the i–th particle at time–step t, $v_i(t)$ be its velocity, pBest_i be its own best previous position, and gBest be the best position ever attained by the swarm. Then, the swarm is manipulated according to the equations [KE01]:

$$\begin{aligned} v_i(t+1) &= w\, v_i(t) + c_1\, r_1\, (\text{pBest}_i - x_i(t)) + \\ &\quad c_2\, r_2\, (\text{gBest} - x_i(t)), \\ x_i(t+1) &= x_i(t) + v_i(t+1), \end{aligned} \tag{1}$$

where r_1, r_2 are realizations of uniformly distributed random variables in $[0, 1]$. The parameters c_1 and c_2 are called *cognitive* and *social* parameters, respectively, and they are used to bias the particle's search towards its own best previous position and towards the best experience of the swarm. The parameter w is called *inertia weight* and it is used to control the trade–off between the global and the local exploration ability of the swarm.

It is not intuitively clear how gBest and pBest can be defined in the case of a MO problem. DOPS uses a Pareto based approach, where for the gBest and for each pBest, an archive of fixed size containing non–dominated positions is maintained. The procedure is described in the next section.

3.2 A New Approach

Several new techniques for the selection of gBest and the update of an overfilled archive have been employed. Deletion methods include an adaptive grid as well as a function based technique that uses a relative distance metric, and a randomized approach. Besides uniform and anti–clustering selection, a new technique, which is based on the success of a particle, is proposed.

Each particle is assigned a selection fitness value and a deletion fitness value according to two functions, f_{sel} and

f_{del}, respectively. The selection fitness function, f_{sel}, is defined as a measure of influence of each particle on the diversity of the Pareto front and it has to be distinguished from the fitness value of the objective function. The selection fitness of a particle increases with its distance to its nearest neighbor.

Each time, a gBest or pBest position is needed, an archive member is selected through a roulette wheel selection over f_{sel}. If the number of non–dominated positions found surpasses the archive size, then a member of the archive is chosen for deletion according to the deletion fitness, f_{del}.

DOPS enables the user to choose between three different selection functions and three different deletion functions, that are defined in the following section.

3.3 Deletion Functions (Factor C)

Adaptive Grid (Type 0): This function uses an adaptive grid that is resized in every generation. This extends the grid variant presented in [CS02]. The grid separates the objective space in so called hypercubes. The edge length of a cube is calculated as follows:

$$\text{edgelength}_j = \frac{c\,(\max_j - \min_j)}{\text{swarm size}}, \tag{2}$$

where $\max_j$ and $\min_j$ are the maximal and minimal values reached by an archive member, evaluated with the j–th objective function.

The real–valued constant $c \in [0, 1]$ represents a selection pressure. The probability that two particles share the same hyper–cube is inversely proportional to this value (a value of $c = 0.5$ was used in the following experiments).

The deletion fitness is defined as

$$f_{\text{del}} = |H|^2, \tag{3}$$

where $|H|$ is the number of particles in the same hypercube. f_{del} punishes particles that form huge particle clusters.

Distance Metric (Type 1): This function uses a distance metric that is based on the relative distances in the archive:

$$f_{\text{del}} = \sum_{\forall j \neq i} \frac{1}{D_{ij}^c}, \tag{4}$$

where,

$$D_{ij} = \sqrt{\sum_{k=1}^{\#\text{ objectives}} \left(\frac{x_i - x_j}{\max_i - \min_i} \right)^2}. \tag{5}$$

A problem may occur since the determination of all pairwise distances has time complexity of $O(n^2)$.

Table 1: DOPS Design Factors.

Factor	Symbol	Description	Level
A	p	number of particles	$\{20; 50\}$
B	w	inertia	$\{0.3; 0.5\}$
C	f_{del}	deletion method	$\{0; 1; 2\}$
D	n	repository size	$\{1; 20; 50; 100; 200\}$
E	f_{sel}	selection method	$\{0; 1; 2\}$

Randomized Distance Metric (Type 2): is a randomized variation of Type 1, that chooses only $\sqrt{n}$ particles for a pairwise fitness calculation.

In the following section, three different strategies to select particles from the archive are proposed.

3.4 Selection Functions (Factor E)

Uniform Selection (Type 0): Every particle has the same selection probability:

$$f_{\text{sel}} = 1/n, \tag{6}$$

where n denotes the number of particles in the archive (at time–step t).

Anti–Clustering Selection (Type 1): This technique is based on the deletion probability f_{del}:

$$f_{\text{sel}} = 1/f_{\text{del}}. \tag{7}$$

The inverse of f_{del} is used to prevent particles in huge clusters to be chosen as gBest or pBest.

Success–Based Selection (Type 2): For each particle, the algorithm keeps track of how often it is chosen as gBest or pBest, and how often this choice leads to another non–dominated particle (parameters ch and suc, respectively):

$$f_{\text{sel}} = \frac{\text{suc}}{1 + \text{ch}} + \frac{\sum_{\forall i} \text{suc}_i}{1 + \sum_{\forall i} \text{ch}_i}. \tag{8}$$

The idea behind this function is to reward archive members that often lead to new non–dominated particles by favoring them in the selection process.

4 Comparisons

4.1 Methodology

To analyze the effects of the new archiving technique described in Section 3, on the PSO's performance, different parameter configurations have been tested. The parameters in Table 1 have been varied, whereas other PSO parameters, such as c_1 and c_2, were held constant. Ten independent

simulation runs were performed for each parameter setting, using the test functions described in Section 4.2. In MO problems, there are two main goals to achieve: convergence to the true Pareto front and maintenance of the solutions' diversity in the Pareto front. Both goals have to be taken into consideration when investigating the performance of an algorithm. Several metrics have been proposed for the estimation of the closeness to the Pareto front, the degree of diversity, or both. A set of different metrics of the first two kinds are usually considered to investigate the performance of an algorithm. Each of these metrics is intended to reveal specific aspects of the algorithm's performance properties. Yet, for the first study of an algorithm, the first question that needs to be addressed is whether or not the algorithm can address both aims, and subsequently to evaluate its performance on these aims. For this purpose, the *hyper–volume metric*, which is described later in Section 4.3, was used, because the resulting value of the simulation runs can be easily compared.

4.2 Test Functions

Many of test functions have been proposed for testing MO heuristic algorithms [FF96, KDZ01, Kur90]. Most of them were developed having in mind problem features that may pose difficulties on detecting the Pareto optimal front and maintaining the population's diversity in the current non–dominated front. In this study three well–known test functions have been used: τ_6 [ZDT00], DTLZ-9 [KDZ01] and Kursawe's function [Kur90]. All functions are restricted on a problem that consists of two, 10–dimensional objective functions.

Test Function 1 (ZDT-6): This test function is a composition of three functions f_1, g and h:

$$\text{Minimize } \tau_6(\vec{x}) = (f_1(x_1), f_2(x)), \tag{9}$$

where

$$\begin{aligned} f_1(\vec{x}) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1), \\ f_2(\vec{x}) &= g(x_2, \ldots, x_m)\cdot \\ &\quad h(f_1(x_1), g(x_2, \ldots, x_m)), \\ g(x_2, \ldots, x_m) &= 1 + 9\sqrt[4]{\frac{(\sum_{i=2}^{m} x_i)}{9}}, \\ h(f_1, g) &= 1 - (f_1/g)^2, \end{aligned}$$

and $\vec{x} = (x_1, \ldots, x_m)$, $m = 10$, and $x_i \in [0, 1]$. The density of solutions increases as we move away from the Pareto optimal front and vice–versa. Even more, the solutions are non–uniformly distributed along the non–convex Pareto front (biased for solutions for which $f_1(x)$ goes up to one). Therefore, it is hard to obtain a well–distributed non–dominated set.

Test Function 2 (DTLZ-9): This is a scalable test function with constraints. The Pareto optimal front is a curve with $f_1 = f_2 = \cdots = f_{M-1}$, while the density of solutions gets thinner towards the Pareto optimal region. In [KDZ01] the parameter settings $n = 10M$ has been proposed, so the number of variables, n, has to be larger than the number of the objectives, M. In our experiments, the setting $n = 5M$ was selected. The true Pareto optimal front lies on the intersection of all $(M-1)$ constraints, indicating the difficulty of solving this problem.

$$\begin{aligned} \text{Minimize} \quad f_j(\vec{x}) &= \sum_{i=\lfloor (j-1)n/M \rfloor}^{\lfloor jn/M \rfloor} x_i^{0.1}, \qquad (10) \\ & \quad j = 1, 2, \ldots, M; \\ \text{subject to} \quad g_j(\vec{x}) &= f_M^2(\vec{x}) + f_j^2(\vec{x}) - 1 \geq 0, \\ & \quad \text{for} \quad j = 1, 2, \ldots, (M-1); \\ & \quad 0 \leq x_i \leq 1, \\ & \quad \text{for} \quad i = 1, 2, \ldots, n. \end{aligned}$$

Test Function 3 (Kursawe): In this test function, the Pareto optimal set is non–convex as well as discontinuous, and, thus, the values of the decision variables that correspond to the true Pareto optimal solutions are difficult to obtain:

$$\begin{aligned} \text{Minimize} \quad & F(f_1(\vec{x}), f_2(\vec{x})), \qquad (11) \\ \text{where} \quad f_1(\vec{x}) &= \sum_{i=1}^{n-1} -10 \ \exp^{-0.2\sqrt{x_i^2 + x_{i+1}^2}}, \\ f_2(\vec{x}) &= \sum_{i=1}^{n} (|x_i|^{0.8} + 5 \ \sin^3(x_i)). \end{aligned}$$

4.3 Performance Metric

A number of performance metrics have been proposed in [Deb01, Kno02] due to the fact, that a proper comparison of the results of a multi–objective optimizer is a complex issue. For this purpose, KEA includes several R–metrics [HJ98], attainment surfaces [FF96] and the hyper–volume metric (S–metric) [ZT99]. The R–metrics compare two non–dominated sets on the basis of some utility functions and determine the expected number of occasions where the solutions of one set are better than the other. Attainment surfaces use a set of non–dominated solutions to define a surface that delineates the objective space into a dominated and a non-dominated region. The S–metric calculates a hyper-volume of a multi-dimensional region enclosed by the non-dominated set to be assessed and a reference point to measure the diversity and the convergence of the obtained non-dominated set. Depending on the chosen

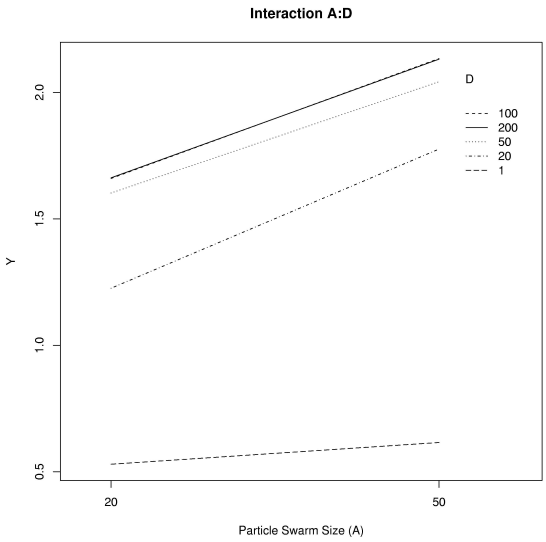

Figure 2: Interaction between particle swarm size and repository size. D denotes the factor repository size. Higher Y values represent better results.

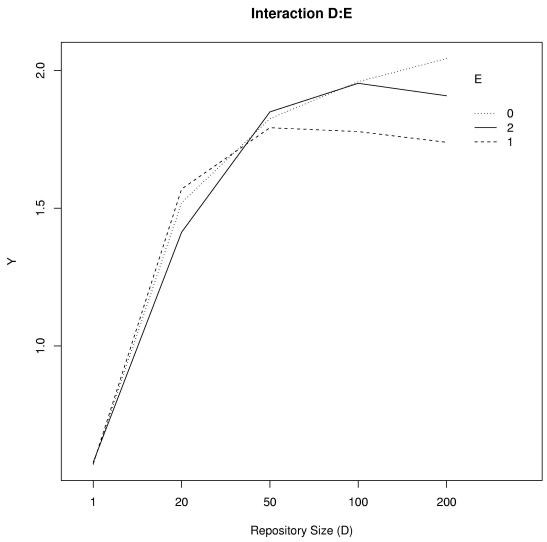

Figure 3: ZDT-6: Interaction between repository size (D) and selection method (E).

reference point, two non–dominated sets can have different relative S–metric values. In this study we use the S-metric, according to which the closeness of the Pareto optimal front as well as the diversity of the obtained solutions can be both evaluated.

We give here a formulation of the S–metrics used in minimization problems: for a non–dominated set A of vectors z^i, $i = \{1, \ldots, |A|\}$, and a reference vector z^{ref} which is dominated by all members of A, the region which is dominated by A and bounded by z^{ref} is defined as the set:

$$R(A, z^{\text{ref}}) = \bigcup_{i \in \{1,\ldots,|A|\}} R(z^i, z^{\text{ref}}), \tag{12}$$

where,

$$R(z^i, z^{\text{ref}}) = y < z^{\text{ref}} \quad \text{and} \quad z^i < y, \quad y \in \mathbb{R}^k.$$

Then, the S-metric is the area defined by the set $R(A, z^{\text{ref}})$. In minimization problem, negative measures shall be avoided for all possible non–dominated sets. In [ZT99], Zitzler proposed a reference vector whose components are the maximum value in each objective. In our experiments, the reference vector $z^{\text{ref}} = (1.4, 8.6)$ was selected.

5 Evaluation

5.1 Questions

Our objective is to develop a robust archiving method that improves the expected performance of a PSO. A robust archiving method is affected minimally by other sources of variability such as swarm size, inertia weights or the random seed that is used to set up the random generator. Thus, interactions have to be considered. Therefore, statistical design of experiments techniques were used to set up an experimental design as shown in Tab. 1 [BPV02]. The particle swarm size is varied as well as the values of the inertia weight, the repository size, and the selection and deletion methods. This design enables the estimation of a good parameter setting of the repository size.

A classical PSO without any explicit archive keeps track of the best value gBest only. It is considered in our context as a PSO with an archive size 1. Archive based PSO maintain an archive with an archive size greater than 1.

5.2 The statistical model

Since the analysis techniques used in the following rely heavily on the assumption of normality, several tests were performed to verify the normality assumption of the PSO performance results. The performance of a PSO algorithm was measured as the fitness value $Y \in \mathbb{R}$ of the S-metric as introduced in Sec. 4.3. These tests show no evidence to reject the assumption of a Gaussian distribution.

Thus, we are able to model a functional relationship between the expected performance $E(Y)$ of the PSO algorithm and its parameter values shown in Tab. 1 through a regression model. In the following the model algebra introduced by Wilkinson and Rogers will be used to describe the regression model [WR73]. Since the S-metric described in Sec. 4.3 was used to generate the result, a maximization

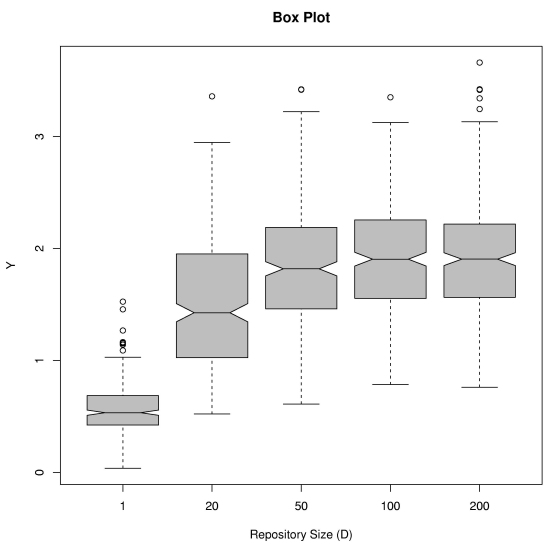

Figure 4: ZDT-6: Boxplots visualizing the influence of different repository sizes on the response Y. A classical PSO has a repository size of 1, since it stores the best solution found so far.

task is analyzed in the following.

The statistical software package R was used to analyze the experimental results [IG96]. A stepwise selection function `stepAIC`, that computes Akaike's information criterion (AIC) for a fitted parametric model, was used to select a linear model automatically. AIC is defined as twice the difference between the number of parameters and the maximized log-likelihood.

5.3 Results

In the following the analysis of the ZDT-6 objective function is described. The test functions DTLZ-9 and Kursawe were analyzed in a similar manner. Starting with the initial model `Y` $\sim$ `A*B*C*D*E`, the suggested final model reads: `Y` $\sim$ `A+B+C+D+E+A:B+A:C+A:D+C:E+D:E` . Since the selection on the basis of AIC is conservative, the `dropterm` function was used to determine the final model `Y` $\sim$ `A+C+D+E+A:D+D:E` manually. The ANOVA table for this model is shown in Tab.2. Considering the effect of the main factors, we can conclude that factor A (swarm size) has a significant effect. The PSO with 50 particles performs better than the PSO with 20 particles. The effect of factor B (inertia weight) is not significant. The same results for factor A and B were obtained from the analysis of the other two test functions, DTLZ-9 and Kursawe.

Table 2: ZDT-6: ANOVA

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	63.13	63.13	197.86	0.0000
C	2	10.01	5.01	15.69	0.0000
D	1	180.51	180.51	565.70	0.0000
E	2	2.33	1.16	3.65	0.0263
A:D	1	2.25	2.25	7.04	0.0080
D:E	2	4.58	2.29	7.18	0.0008
Residuals	1602	511.17	0.32		

Important for the analysis of the archiving techniques are factors that are directly related to the archive: C, D, and E. Factor C (deletion method) has a significant effect, methods 1 and 2 are superior compared to method 0. Consider factor D (repository size) next: Increasing the repository size from 1 to 20 particles and from 20 to 50 particles results in an improved performance (a t-test was performed to compare the average fitness values) of the PSO, cf. Fig. 4. Further augmentations of the repository size lead to non–significant improvements. Thus, a repository size of 50 particles seems to be a good compromise between archiving cost and archiving benefit. Factor E (selection method) has no significant effect and is considered in the model due to its interaction with factor D only. This model includes interactions between the number of particles and the repository size (interaction `A:D`) and interactions between the repository size and the selection method (interaction `D:E`). Interaction plots reveal no conflicting interactions between the particle swarm size and the repository size (Fig. 2), whereas the interactions between repository size and selection method need some further analysis: Fig. 3 leads to the conclusion that selection method 0 can be recommended since: a) it shows the same improvement as the other methods if repository sizes of 1, 20, and 50 particles are compared and b) it is superior to the other methods if the repository size is increased.

It is important to mention that the results discussed so far are not specific for the ZDT-6 test function only. The analysis of the other two test functions reveals similar results. This can be interpreted as a strong hint that archiving can improve the PSO performance significantly and will be subject of further research.

6 Summary and Outlook

A new particle swarm optimization algorithm for multiple criteria optimization problems was introduced in the first part of this paper. The swarm method solved well–known test problems covering difficult cases of multi–criteria optimization. Experimental design techniques have been used to set up the experiments and to perform the data analysis. The obtained results gave first hints that the use of archives in PSO for MCO problems improves their performance. First experimental results demonstrate that moderate archive sizes represent a good compromise between the

costs and benefits of archiving. Experiments analyzing the different selection and deletion strategies will be subject of further research.

For future research, it might be useful to transfer the results to other test functions specially real–world applications.

Acknowledgments. Thomas Bartz–Beielstein's and Karlheinz Schmitt's research was supported by the DFG as a part of the collaborative research center 'Computational Intelligence' (SFB 531) in Dortmund.

Bibliography

[Abi02] M.A. Abido. Optimal design of power system stabilizers using particle swarm optimization. *IEEE Trans. Energy Conversion*, 17(3):406–413, 2002.

[BBSM+03] T. Bartz-Beielstein, K. Schmitt, J. Mehnen, B. Naujoks, and D. Zibold. KEA – a software package for development, analysis and application of multiple objective evolutionary algorithms. Technical report, University of Dortmund, 2003.

[BDT99] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Berlin, springer edition, 1999.

[BPV02] T. Beielstein, K.E. Parsopoulos, and M.N. Vrahatis. Tuning PSO parameters through sensitivity analysis. Technical Report of the Collaborative Research Center 531 *Computational Intelligence* CI–124/02, University of Dortmund, January 2002.

[CDG99] D. Corne, M. Dorigo, and F. Glover. *New Ideas in Optimization*. McGraw-Hill, 1999.

[CH01] A.R. Cockshott and B.E. Hartman. Improving the fermentation medium for *Echinocandin B* production part ii: Particle swarm optimization. *Process Biochemistry*, 36:661–669, 2001.

[Coe99] C. A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization. *Knowledge and Information Systems*, 1(3):269–308, 1999.

[CS02] C. A. Coello Coello and M. Salazar Lechuga. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In *Congr. on Evolutionary Computation (CEC'2002)*, volume 2, pages 1051–1056, Piscataway, New Jersey, May 2002. IEEE Service Center.

[DAPM00] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proc. of the Parallel Problem Solving from Nature VI*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[Deb01] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Ltd., 1 edition, 2001.

[FF96] C. M. Fonseca and P. J. Fleming. On the performace assessment and comparison of stochastic multiobjective optimizers. In I. Rechenberg H.-M. Voigt, W. Ebeling and H.-P. Schwefel, editors, *Parellel Problem Solving from Nature IV (PPSN-IV)*, Lecture Notes in Comupter Science, pages 584–593, Berlin, Germany, September 1996.

[HJ98] M. P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations to the nondominated set. Technical Report IMM-REP-1998-7, Technical University of Denmark, March 1998.

[Hu02] X. Hu. Multiobjective optimization using dynamic neighborhood particle swarm optimization. In *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, Honolulu, HI, USA, 2002.

[IG96] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[JOS01] Y. Jin, M. Olhofer, and B. Sendhoff. Evolutionary dynamic weighted aggregation for multiobjective optimization: Why does it work and how? In *Proc. GECCO 2001 Conf.*, pages 1042–1049, San Francisco, CA, 2001.

[KC99] J. D. Knowles and D. W. Corne. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization. In *Congress on Evolutionary Computation (CEC 2000)*, pages 325–332, Piscataway, NJ, 1999. IEEE Press.

[KC03] J.D. Knowles and D.W. Corne. Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Trans. on Evolutionary Computation*, 7(2):100–116, 2003.

[KDZ01] M. Laumanns K. Deb, L. Thiele and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich, Ch-8092, Zürich, July 2001.

[KE95] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *Proc. IEEE Int.'l Conf. on Neural Networks*, volume IV, pages 1942–1948, Piscataway, NJ, 1995. IEEE Service Center.

[KE01] J. Kennedy and R.C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.

[Kno02] J. D. Knowles. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, The University of Reading, Departement of Computer Science, Reading, UK, January 2002.

[Kur90] F. Kursawe. A variant of evolution strategies for vector optimization. In H.-P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature – Proc. First Workshop PPSN*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197, University of Dortmund, October 1–3, 1990. Springer, Berlin, 1991.

[LPV02] E.C. Laskari, K.E. Parsopoulos, and M.N. Vrahatis. Particle swarm optimization for minimax problems. In *Proc. of the IEEE 2002 Congress on Evolutionary Computation*, pages 1582–1587, Hawaii (HI), USA, 2002. IEEE Press.

[LTZD02] M. Laumanns, L. Thiele, E. Zitzler, and K. Deb. Archiving with guaranteed convergence and diversity in multi–objective optimization. In *Proc. of the GECCO 2002 Conf.*, pages 439–447, San Francisco (CA), USA, 2002.

[MMBS03] J. Mehnen, Th. Michelitsch, T. Beielstein, and K. Schmitt. Evolutionary optimization of mold temperature control strategies – encoding and solving the multi-objective problem with standard ES and KEA. *Journal of Engineering Manufacture*, 2003. (Submitted).

[PV02a] K.E. Parsopoulos and M.N. Vrahatis. Particle swarm optimization method for constrained optimization problems. In P. Sincak, J. Vascak, V. Kvasnicka, and J. Pospichal, editors, *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies*, volume 76 of *Frontiers in Artificial Intelligence and Applications*, pages 214–220. IOS Press, 2002.

[PV02b] K.E. Parsopoulos and M.N. Vrahatis. Particle swarm optimization method in multiobjective problems. In *Proc. of the 2002 ACM Symposium on Applied Computing (SAC 2002)*, pages 603–607, Madrid, Spain, 2002. ACM Press.

[RA00] G. Rudolph and A. Agapie. Convergence properties of some multi–objective evolutionary algorithms. In *Proc. of the IEEE 2000 Conf. on Evolutionary Computation*, pages 1010–1016. Piscataway, NJ: IEEE Press, 2000.

[VVZL03] D. A. Van Veldhuizen, J. B. Zydallis, and G. B. Lamont. Considerations in engineering parallel multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):144–173, 2003.

[WR73] G. N. Wilkinson and C. E. Rogers. Symbolic description of factorial models for analysis of variance. 22(3):392–399, 1973.

[ZDT00] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolution algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[ZT99] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparativce case study and the strength Pareto approach. *IEEE Transaction on Evolutionary Computation*, 3(4):257–271, November 1999.