Differential Evolution for Multiobjective Optimization with Self Adaptation

Aleš Zamuda, *Student Member, IEEE*, Janez Brest, *Member, IEEE*, Borko Bošković, *Student Member, IEEE*, Viljem Žumer, *Member, IEEE*

***Abstract*— This paper presents performance assessment of Differential Evolution for Multiobjective Optimization with Self Adaptation algorithm, which uses the self adaptation mechanism from evolution strategies to adapt F and CR parameters of the candidate creation in DE. Results for several runs on CEC2007 special session test functions are presented and assessed with different performance metrics. Based on these metrics, algorithm strengths and weaknesses are discussed.**

I. Introduction

In the last decade, there has been a growing interest in applying randomized search algorithms such as evolutionary algorithms, simulated annealing, and tabu search to multiobjective optimization problems in order to approximate the set of Pareto-optimal solutions [1]. Various methods have been proposed for this purpose, and their usefulness has been demonstrated in several application domains [2]. Since then a number of performance assessment methods have been suggested. Most of the existing simulation studies comparing different evolutionary multiobjective methodologies are based on specific performance measures.

In this study we assess a new multiobjective evolutionary algorithm, based on the DEMO algorithm [3], [4]. Among the other optimizers using Differential Evolution (DE) [5] in multiobjective optimization [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], the DEMO algorithm is using DE for the candidate solution creation. Environmental selection in DEMO can be chosen to be either the NSGA-II [16], SPEA2 [17], or IBEA [18] algorithm's environmental selection. Our previous experience showed that DE with self-adaptation, which is well known in evolution strategies [19], [20], [21], can lead to faster convergence than the original DE [22], [23]. Therefore we decided to extend the DEMO algorithm by incorporating the self-adaptation mechanism in DE control parameters.

Assessment of the new algorithm is performed on the test functions from the CEC2007 special session [24] with the therein provided performance assessment metrics. The test functions suite used comprises some functions proposed recently such as OKA2 [25] and SYMPART [26]. The test functions suite also consists of rotated or scaled ZDT [27] and DTLZ [28] functions. The last part of the test functions suite consists of WFG functions [29]. Each DTLZ and WFG function used has two variants in the test suite, with different number of search parameters and number of objective functions. The performance metrics [1] used to evaluate the attained approximation sets with our optimizer are taken from [30]. The I_H and I_{R2} indicators are computed (for all functions) and the CS indicator is computed for the SYMPART function. The empirical attainment surfaces are also calculated using [30]. The algorithm complexity is given at the end.

This paper is organized as follows. In the next section, our new algorithm DEMOwSA is described. The third section describes the experiments and the obtained results on the test suite. The results are assessed by accompanying performance metrics. The last section concludes with final remarks and prepositions for future work.

II. Differential Evolution for Multiobjective Optimization with Self Adaptation (DEMOwSA)

As mentioned in the introduction, evolution strategies include a self-adaptive mechanism [19], encoded directly in each individual of the population to control some of the parameters in the evolutionary algorithm. An evolution strategy (ES) has a notation $\mu/\rho, \lambda$-ES, where μ is parent population size, ρ is the number of parents for each new individual, and λ is child population size. An individual is denoted as $\vec{a} = (\vec{x}, \vec{s}, F(\vec{x}))$, where $\vec{x}$ are search parameters, $\vec{s}$ are control parameters, and $F(\vec{x})$ is the evaluation of the individual. Although there are other notations and evolution strategies, we will only need a basic version. We use an idea of self-adaptation mechanism from evolution strategies and apply this idea to the control parameters of the candidate creation in the original DEMO algorithm [3]. We name the new constructed version of DEMO, DEMOwSA algorithm.

Each individual (see Fig. 1) of DEMOwSA algorithm is extended to include self-adaptive F and CR control parameters to adjust them to the appropriate values during the evolutionary process. The F parameter is mutation control parameter and CR is the crossover control parameter.

For each individual in the population, a trial vector is composed by mutation and recombination. The mutation procedure is different in the DEMOwSA algorithm in comparison to the original DEMO. For adaptation of the amplification factor of the difference vector F_i for trial individual i, from parent generation G into child generation $G+1$ for the trial vector, the following formula is used:

$$F_{i,G+1} = \langle F_G \rangle_i \times e^{\tau N(0,1)},$$

The authors are with the Computer Architecture and Languages Laboratory, Institute of Computer Science, Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova ulica 17, SI-2000 Maribor, Slovenia (phone: ++386 2 220 7404; fax: ++386 2 220 7272; email: {ales.zamuda, janez.brest, borko.boskovic, zumer}@uni-mb.si).

1-4244-1340-0/07$25.00 ©2007 IEEE

$x_{1,1,G}$	$x_{1,2,G}$	$x_{1,3,G}$	...	$x_{1,D,G}$	$F_{1,G}$	$CR_{1,G}$
$x_{2,1,G}$	$x_{2,2,G}$	$x_{2,3,G}$	...	$x_{2,D,G}$	$F_{2,G}$	$CR_{2,G}$
$x_{3,1,G}$	$x_{3,2,G}$	$x_{3,3,G}$	...	$x_{3,D,G}$	$F_{3,G}$	$CR_{3,G}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$x_{NP,1,G}$	$x_{NP,2,G}$	$x_{NP,2,G}$	...	$x_{NP,D,G}$	$F_{NP,G}$	$CR_{NP,G}$

Fig. 1. Encoding of the self-adapting control parameters.

where τ denotes learning factor and is usually proportional to $\tau = 1/(\sqrt{2D})$, D being a dimension of the problem. $N(0,1)$ is a random number with a Gauss distribution. The $\langle F_G \rangle_i$ denotes averaging the parameter F of the current individual i and the randomly chosen individuals r_1, r_2, and r_3 from the generation G:

$$\langle F_G \rangle_i = \frac{F_{i,G} + F_{r_1,G} + F_{r_2,G} + F_{r_3,G}}{4},$$

where indices r_1, r_2, and r_3 are the indices of individuals also being used in the chosen DE strategy for the mutation process. The mutation process for i-th candidate $\vec{v}_{i,G+1}$ for generation $G+1$ is defined as:

$$\vec{v}_{i,G+1} = \vec{x}_{r_1,G} + F_{i,G+1} \times (\vec{x}_{r_2,G} - \vec{x}_{r_3,G}),$$

where $\vec{x}_{r_1,G}$, $\vec{x}_{r_2,G}$, and $\vec{x}_{r_3,G}$ are search parameter values of the uniform randomly selected parents from generation G.

An analogous formula is used for the CR parameter:

$$CR_{i,G+1} = \langle CR_G \rangle_i \times e^{\tau N(0,1)},$$

where τ is same as in the adaptation of F parameter and $\langle CR_G \rangle_i$ denotes averaging:

$$\langle CR_G \rangle_i = \frac{CR_{i,G} + CR_{r_1,G} + CR_{r_2,G} + CR_{r_3,G}}{4}.$$

Recombination process is taken from the strategy *'rand/1/bin'* used in the original DEMO [3] or the original DE [5], and the adapted $CR_{i,G+1}$ is used to create a modified candidate $u_{i,j,G+1}$ by binary crossover:

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{if } rand(0,1) \leq CR_{i,G+1} \text{ or} \\ & j = j_{rand} \\ x_{i,j,G} & \text{otherwise,} \end{cases}$$

where $j \in [1, D]$ denotes the j-th search parameter, $rand(0,1) \in [0,1]$ denotes a uniformly distributed random number, and j_{rand} denotes an uniform randomly chosen index of the search parameter, which is always exchanged.

The selection principle also helps in adapting F and CR parameters, because only the individuals adapting good control and search parameter values survive. With the more appropriate values for the self-adaptive control parameters, the search process attains better objective space parameters. Therefore, the search process converges faster to better individuals which, in turn, are more likely to survive and produce offspring and, hence, propagate these better parameter values.

III. Results

Since the approximation set results for DEMO on CEC2007 special session test functions were not freely available on the web, nor were they ever performed for such parameter settings with same functions, we produced approximation set results for performance assessment ourselves. The DEMOwSA implementation did not exist either, so we implemented the algorithm by extending the existing DEMO code.

A. PC Configuration

All tests were performed on the following PC configuration. System: Linux, 2.6.17-1.2142_FC4smp. CPU: Twice 64-bit Dual Core AMD Opteron(tm) Processor 280 (2390.651 MHz, 4787.74 bogomips). RAM: 8GB. Language: C++. Algorithm: DEMOwSA.

B. Parameter Settings

The DEMOwSA algorithm has many parameters, among which the most important are the selection algorithm and candidate creation strategy. We have chosen SPEA2 as the selection algorithm and our candidate creation strategy based on self adaptation mechanism. We have adjusted the F and CR parameters of the candidate creation process with DE. The maximal dynamic range of F parameter is $F \in [0,2]$, and for CR it is $CR \in [0,1]$. With the applied adaptation mechanism, some control parameter constraints were used, as will be described below. The estimated cost of parameter tuning in terms of number of function evaluations is zero – only a few additional random numbers and simple calculations are conducted to express the adapted control parameters. No search is required and one control parameters adaptation is executed before each function evaluation.

In the presented experiments the following parameter settings were used for our algorithm. The global lower and upper bounds for control parameter F were $0.1 \leq F \leq 0.9$, and for control parameter CR they were $1/24 \leq CR \leq 0.3$. Their initialization was $F = 0.5$, $CR = 0.3$. The τ parameter was set to $1/(8\sqrt{2D})$, which conforms to the recommendation from [19]. The population size parameter NP was set equal to approximation set size for each of the functions, i.e. 100, 150, and 800, respectively.

C. Results Achieved

Based on the intermediate and final resulting approximation sets of optimization process, performance assessment was induced based on CEC2007 special session performance assessment metrics. Regarding the DEMOwSA for all the test functions, an approximation set was recorded in each of the evolutionary runs after 5e+3, 5e+4, and 5e+5 function evaluations (FEs). Regarding the performance assessment suite we present:

- best, median, worst, mean, and standard variance for values of the indicators I_{R2} and $I_{\overline{H}}$ of all 25 runs for test functions 1–7 and 8–13 (with $M=3$ and $M=5$) after 5e+3, 5e+4, and 5e+5 FEs: three tables for each of

two indicators result from this requirement (see Tables I, II, III, IV, V, and VI),
- best, median, worst, mean, and standard variance for values of the indicator CS of all 25 runs for test function SYMPART after 5e+3, 5e+4, and 5e+5 FEs: one table results from this requirement (see Table VII),
- 0%, 50%, and 100% empirical attainment surfaces of all 25 runs for test functions 1–7 after 5e+5 FEs: a figure with seven subfigures results from this requirement (see Fig. 2),
- 50% empirical attainment surfaces of all 25 runs for test functions 8-13 with $M = 3$, after 5e+5 FEs: one figure with six subfigures results from this requirement (see Fig. 3),
- Pareto front plots after 5e+5 FEs for test functions 12 and 13 with $M = 5$: one figure with subfigures for each objective function combination results from this requirement (see Fig. 4),
- computing time of 10,000 evaluations (T_1) and of the same with the algorithm (T_2) timing: one table results from this requirement (see Table VIII).

From all tables, it can be observed that, from most indicator values, the algorithm is mostly successful at least to the level of attaining 0.01 close to the ideal performance metric value (which is 0). When the obtained indicator value is negative, we checked various plots and numerical data and determined that, because of the nonevenly distributed and less dense reference set than the approximation set, there are minor improvements compared to the reference set in the indicators for some of our approximations sets.

From Table I, it can be seen that our algorithm performed well with respect to the given reference sets for functions OKA2, SYMPART, S_ZDT1, S_ZDT4, R_ZDT4, and S_ZDT6. The function S_ZDT2 was the most difficult among these functions with regard to the R indicator. Table II shows good algorithm performance for S_DTLZ2, R_DTLZ2, and S_DTLZ3 as also for the WFG8 and WFG9. For the function S_DTLZ2, the indicator values display a small anomaly from 5e+4 FEs to 5e+5 FEs, where they get a bit worse. In this table the best solved function is WFG8 and the worst solved is WFG1, which is also true for Table III. From Tables IV, V, and VI, it can be observed that the hardest function to optimize was WFG1 with $M = 3$, and $M = 5$, whereas indicator values for the functions S_ZDT6, WFG8 and WFG9 with $M = 3$, and the functions WFG8 and WFG9 with $M = 5$ are even better than the corresponding indicator values of their reference sets.

IV. Conclusions

We have presented performance assessment of Differential Evolution for Multiobjective Optimization with Self Adaptation algorithm. The algorithm uses the self adaptation mechanism from evolution strategies to adapt F and CR parameters of the candidate creation in DE. Results for 25 runs on 19 test functions are presented and assessed using 4 performance metrics. Based on these metrics, the algorithm

TABLE VII
The results for covered sets for test function SYMPART.

FES	5e+3	5e+4	5e+5
Best	1.0000e+00	1.0000e+00	1.0000e+00
Median	1.0000e+00	1.0000e+00	1.0000e+00
Worst	1.0000e+00	1.0000e+00	1.0000e+00
Mean	1.0000e+00	1.0000e+00	1.0000e+00
Std	0.0000e+00	0.0000e+00	0.0000e+00

TABLE VIII
Algorithm Complexity.

T_1	T_2	$(T_2 - T_1)/T_1$
1.0600e+00 s	2.7693e+01 s	2.5130e+01

averagely attains good results on the test suite. The most problematic functions that were encountered were S_ZDT4, OKA2, and WFG1. On all the remaining functions the algorithm performed better, attaining at least 0.01 proximity to the Pareto optimal set regarding the performace metrics I_H and I_{R2}.

We have already conducted a preliminary F and CR dynamics study on the algorithm, which cannot be presented here due to paper page limit, and is a good starting point for further research. Hybridizing our algorithm with local search seems to be a good idea as well. Performance assessment of DEMO with IBEA and NSGA-II has also been conducted, but the metrics and results interpretation are left for future research, although it already seems that SPEA is performing better on the test functions regarding the given quality indicators.

Acknowledgment

This work was supported in part by the Slovenian Research Agency under programme P2-0041 — Computer systems, Methodologies, and Intelligent Services. Special thanks goes to Tea Tušar of Department of Intelligent Systems at the Jozef Stefan Institute, Jamova 39, SI - 1000 for providing us with the codes of the original DEMO algorithm.

References

[1] J. Knowles, L. Thiele, and E. Zitzler, "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland," 214, Feb. 2006, revised version.

[2] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: John Wiley & Sons, 2001, ISBN 0-471-87339-X.

[3] T. Robič and B. Filipič, "DEMO: Differential Evolution for Multiobjective Optimization," in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization – EMO 2005*, ser. Lecture Notes in Computer Science, vol. 3410. Springer, 2005, pp. 520–533.

[4] T. Tušar and B. Filipič, "Differential Evolution versus Genetic Algorithms in Multiobjective Optimization," in *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization – EMO 2007*, ser. Lecture Notes in Computer Science, vol. 4403. Springer, 2007, pp. 257–271.

[5] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

TABLE I

THE RESULTS FOR R INDICATOR ON TEST FUNCTIONS 1–7.

FES		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Best	-6.2136e-04	2.1494e-02	5.3022e-02	1.0790e-01	7.5437e-02	1.9364e-02	1.3688e-01
	Median	8.4401e-04	3.2792e-02	5.8049e-02	1.2095e-01	8.4788e-02	2.5630e-02	1.4261e-01
	Worst	4.0591e-03	4.7681e-02	6.7336e-02	1.3360e-01	9.3693e-02	3.3457e-02	1.4606e-01
	Mean	1.0738e-03	3.2867e-02	5.9281e-02	1.2142e-01	8.5307e-02	2.5097e-02	1.4213e-01
	Std	1.1723e-03	6.8335e-03	3.9633e-03	6.4693e-03	5.0711e-03	3.7516e-03	2.1999e-03
5e+4	Best	-9.7859e-04	1.6145e-05	1.4196e-04	3.2265e-04	2.4269e-02	1.6790e-03	2.5129e-02
	Median	-3.0302e-04	2.5157e-05	2.4980e-04	6.9049e-04	3.0425e-02	3.2908e-03	2.9697e-02
	Worst	9.9941e-04	3.8961e-05	3.0445e-04	4.0628e-02	3.5089e-02	5.6485e-03	4.2450e-02
	Mean	-2.6206e-04	2.5823e-05	2.4294e-04	1.5529e-02	3.0420e-02	3.2849e-03	3.0595e-02
	Std	6.0745e-04	5.0663e-06	3.7686e-05	1.9750e-02	3.0321e-03	9.8099e-04	4.0446e-03
5e+5	Best	-1.0581e-03	3.2744e-06	-8.1112e-10	0.0000e+00	1.7241e-03	1.1622e-04	-1.0216e-06
	Median	-1.0044e-03	4.1441e-06	1.8302e-06	3.1582e-08	2.7763e-03	2.5458e-04	-1.0216e-06
	Worst	-1.0624e-04	5.0803e-06	6.9727e-06	4.0053e-02	4.5511e-03	1.4549e-03	-9.3898e-07
	Mean	-8.0507e-04	4.1873e-06	2.2716e-06	1.4419e-02	3.0107e-03	3.6550e-04	-1.0152e-06
	Std	3.0496e-04	5.4693e-07	2.1717e-06	1.9622e-02	7.3453e-04	3.3329e-04	2.2215e-08

TABLE II

THE RESULTS FOR R INDICATOR ON TEST FUNCTIONS 8–13 WHEN $M = 3$.

FES		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Best	1.1818e-04	3.5323e-04	2.7764e-04	5.5449e-02	-1.3805e-02	-8.4837e-03
	Median	1.8319e-04	2.2186e-03	3.3250e-04	5.5973e-02	-1.1809e-02	-6.7991e-03
	Worst	2.7839e-04	4.1487e-03	4.1502e-04	5.6303e-02	-9.7870e-03	-4.9215e-03
	Mean	1.8734e-04	2.2245e-03	3.3716e-04	5.5934e-02	-1.1773e-02	-6.7800e-03
	Std	3.8855e-05	1.0171e-03	3.4022e-05	2.2945e-04	1.0110e-03	8.5961e-04
5e+4	Best	2.6126e-06	2.5312e-04	7.3613e-05	5.2649e-02	-2.5590e-02	-8.7265e-03
	Median	2.0169e-05	1.5043e-03	8.0394e-05	5.3474e-02	-2.4526e-02	-7.5057e-03
	Worst	3.7881e-05	2.5698e-03	8.6294e-05	5.3927e-02	-2.2079e-02	-6.2060e-03
	Mean	2.0590e-05	1.3952e-03	8.0801e-05	5.3434e-02	-2.4488e-02	-7.4822e-03
	Std	1.0259e-05	6.8988e-04	3.5885e-06	3.2994e-04	7.0955e-04	6.0076e-04
5e+5	Best	9.8209e-06	2.4133e-04	1.0461e-05	2.7007e-02	-2.7946e-02	-8.9103e-03
	Median	2.8298e-05	5.3103e-04	1.4378e-05	3.8503e-02	-2.7509e-02	-7.7135e-03
	Worst	6.5781e-05	1.5445e-03	2.1224e-05	4.2837e-02	-2.7140e-02	-6.5550e-03
	Mean	3.2122e-05	6.1748e-04	1.4618e-05	3.7772e-02	-2.7515e-02	-7.8231e-03
	Std	1.6826e-05	3.8772e-04	2.7490e-06	3.5759e-03	1.9937e-04	5.9615e-04

TABLE III

THE RESULTS FOR R INDICATOR ON TEST FUNCTIONS 8–13 WHEN $M = 5$.

FES		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Best	2.5972e-04	2.3065e-04	1.9638e-04	4.7040e-02	7.3327e-03	6.7127e-03
	Median	4.0952e-04	6.3999e-04	2.5457e-04	4.7170e-02	8.9772e-03	1.2295e-02
	Worst	8.9412e-04	1.3679e-03	3.3627e-04	4.7337e-02	1.1911e-02	1.5543e-02
	Mean	4.4348e-04	7.0549e-04	2.5939e-04	4.7182e-02	9.3102e-03	1.2250e-02
	Std	1.5641e-04	3.1821e-04	3.5440e-05	6.5757e-05	1.2508e-03	1.9841e-03
5e+4	Best	2.2147e-05	8.4360e-05	1.7229e-05	4.6510e-02	1.4684e-03	1.3739e-03
	Median	2.6692e-05	1.9724e-04	2.4391e-05	4.6866e-02	2.9816e-03	3.1034e-03
	Worst	3.8250e-05	6.9037e-04	3.3276e-05	4.7134e-02	4.8075e-03	4.3950e-03
	Mean	2.7842e-05	2.5883e-04	2.4993e-05	4.6881e-02	3.0397e-03	3.0019e-03
	Std	4.7033e-06	1.7582e-04	4.1082e-06	1.3306e-04	9.8039e-04	7.0520e-04
5e+5	Best	3.2301e-06	7.2334e-05	5.8196e-06	4.5573e-02	-5.9577e-03	1.2386e-03
	Median	6.7514e-06	1.0759e-04	7.0181e-06	4.5880e-02	-3.9907e-03	2.6170e-03
	Worst	1.3679e-05	3.2339e-04	7.9988e-06	4.6167e-02	-1.7535e-03	3.5564e-03
	Mean	7.8107e-06	1.3375e-04	6.9797e-06	4.5889e-02	-3.9639e-03	2.5642e-03
	Std	3.0363e-06	6.3656e-05	5.4495e-07	1.5295e-04	9.9207e-04	5.9861e-04

TABLE IV

THE RESULTS FOR HYPERVOLUME INDICATOR $I_{\overline{H}}$ ON TEST FUNCTIONS 1–7.

FES		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Best	2.4143e-02	6.1409e-02	1.9323e-01	2.6134e-01	2.2752e-01	5.9711e-02	3.4675e-01
	Median	2.8175e-02	9.3032e-02	2.1597e-01	3.1546e-01	2.5866e-01	7.7408e-02	3.6269e-01
	Worst	3.5897e-02	1.3406e-01	2.3718e-01	3.4924e-01	2.8511e-01	9.9695e-02	3.6999e-01
	Mean	2.8984e-02	9.3160e-02	2.1654e-01	3.1091e-01	2.5908e-01	7.6820e-02	3.6117e-01
	Std	3.3118e-03	1.8985e-02	1.0037e-02	2.1469e-02	1.6146e-02	9.9631e-03	5.4189e-03
5e+4	Best	1.9719e-02	4.8726e-05	7.8927e-04	1.0321e-03	7.1588e-02	5.5318e-03	5.7133e-02
	Median	2.2169e-02	7.3721e-05	1.2176e-03	2.1531e-03	8.9864e-02	1.0047e-02	6.8166e-02
	Worst	3.0472e-02	1.1745e-04	1.4637e-03	4.9544e-02	1.0379e-01	1.7009e-02	9.4732e-02
	Mean	2.3433e-02	7.6615e-05	1.2287e-03	1.9458e-02	8.9844e-02	1.0123e-02	6.9641e-02
	Std	3.0383e-03	1.5276e-05	1.4700e-04	2.3477e-02	9.0167e-03	2.8202e-03	8.7574e-03
5e+5	Best	1.2041e-02	9.7898e-06	1.3739e-04	1.8058e-04	5.5745e-03	3.2075e-04	-2.3087e-04
	Median	1.5692e-02	1.2586e-05	1.4282e-04	1.9367e-04	8.6292e-03	7.4731e-04	-2.3037e-04
	Worst	2.9064e-02	1.5293e-05	1.6001e-04	4.7812e-02	1.3784e-02	4.2179e-03	-2.2917e-04
	Mean	1.7877e-02	1.2558e-05	1.4549e-04	1.7334e-02	9.3111e-03	1.0772e-03	-2.3035e-04
	Std	5.0099e-03	1.6817e-06	6.8624e-06	2.3330e-02	2.1321e-03	9.7863e-04	4.5290e-07

TABLE V

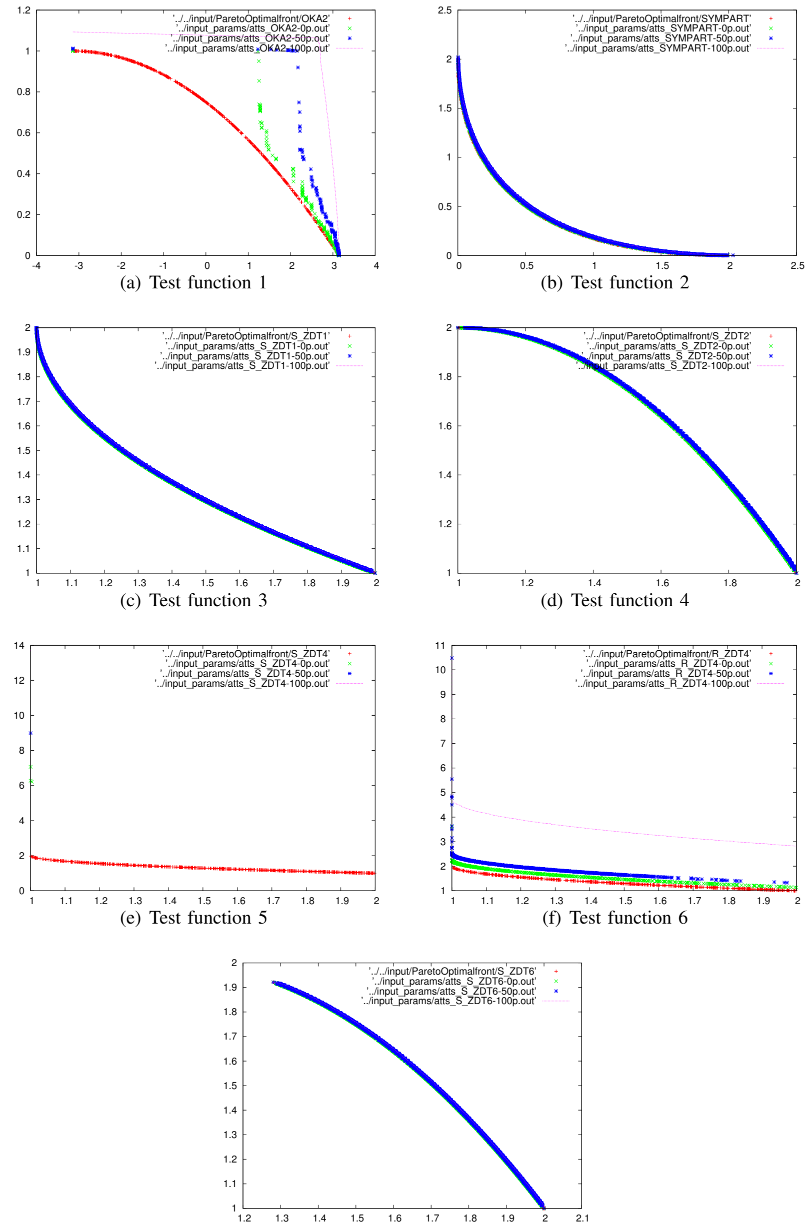
THE RESULTS FOR HYPERVOLUME INDICATOR $I_{\overline{H}}$ ON TEST FUNCTIONS 8–13 WHEN $M = 3$.

FES		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Best	2.6295e-03	1.6055e-02	4.8341e-03	2.8445e-01	-9.5554e-02	-5.0487e-02
	Median	3.3986e-03	2.4112e-02	8.4427e-03	2.8649e-01	-8.6996e-02	-4.2278e-02
	Worst	4.0632e-03	3.3955e-02	9.7431e-03	2.8815e-01	-8.0269e-02	-3.1775e-02
	Mean	3.3835e-03	2.4995e-02	8.2174e-03	2.8646e-01	-8.7677e-02	-4.1769e-02
	Std	3.8775e-04	5.0935e-03	1.2046e-03	9.8989e-04	4.1093e-03	3.9215e-03
5e+4	Best	3.4065e-05	1.2632e-02	2.3325e-04	2.6922e-01	-1.5981e-01	-5.7051e-02
	Median	7.1711e-05	1.9742e-02	3.0622e-04	2.7308e-01	-1.5562e-01	-5.2706e-02
	Worst	2.2160e-04	2.3879e-02	3.8901e-04	2.7518e-01	-1.4652e-01	-4.9750e-02
	Mean	9.1415e-05	1.9169e-02	3.0148e-04	2.7296e-01	-1.5575e-01	-5.2631e-02
	Std	5.8805e-05	3.3016e-03	4.1183e-05	1.5300e-03	2.7123e-03	1.5370e-03
5e+5	Best	1.8285e-05	1.0053e-02	5.5737e-07	1.4021e-01	-1.7271e-01	-5.7978e-02
	Median	8.1342e-05	1.3384e-02	1.8037e-06	1.9913e-01	-1.7071e-01	-5.3687e-02
	Worst	2.7180e-04	1.7965e-02	5.0289e-06	2.2126e-01	-1.6818e-01	-5.0289e-02
	Mean	1.1109e-04	1.3482e-02	1.9128e-06	1.9554e-01	-1.7056e-01	-5.3982e-02
	Std	7.1987e-05	2.2466e-03	1.0168e-06	1.8141e-02	1.0640e-03	1.9586e-03

TABLE VI

THE RESULTS FOR HYPERVOLUME INDICATOR $I_{\overline{H}}$ ON TEST FUNCTIONS 8–13 WHEN $M = 5$.

FES		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Best	1.1614e-02	1.4329e-02	7.7667e-03	5.2944e-01	-4.1803e-02	1.1185e-01
	Median	1.5096e-02	1.9708e-02	9.0675e-03	5.3124e-01	-2.9683e-02	1.3650e-01
	Worst	2.1401e-02	3.0605e-02	1.0691e-02	5.3260e-01	-7.7059e-03	1.6541e-01
	Mean	1.5551e-02	2.1254e-02	9.0716e-03	5.3115e-01	-2.7957e-02	1.3861e-01
	Std	2.5813e-03	4.1541e-03	8.2674e-04	6.7832e-04	9.0027e-03	1.4118e-02
5e+4	Best	2.8815e-05	1.0009e-02	3.2854e-05	5.2599e-01	-1.8195e-01	-1.0449e-01
	Median	4.5833e-05	1.2463e-02	5.1549e-05	5.2911e-01	-1.6911e-01	-9.3887e-02
	Worst	8.8489e-05	1.9405e-02	6.8974e-05	5.3097e-01	-1.5794e-01	-8.3515e-02
	Mean	4.9841e-05	1.3054e-02	5.1144e-05	5.2895e-01	-1.7012e-01	-9.3784e-02
	Std	1.6208e-05	2.5257e-03	8.7546e-06	1.2583e-03	7.4509e-03	5.9951e-03
5e+5	Best	2.5086e-08	8.1098e-03	5.3786e-07	5.1726e-01	-2.7190e-01	-1.2913e-01
	Median	1.5131e-06	9.6464e-03	7.8897e-07	5.2033e-01	-2.5721e-01	-1.1966e-01
	Worst	5.6978e-06	1.3215e-02	1.2490e-06	5.2388e-01	-2.4125e-01	-1.1439e-01
	Mean	2.0109e-06	9.8835e-03	8.0469e-07	5.2053e-01	-2.5771e-01	-1.2020e-01
	Std	1.6375e-06	1.1920e-03	1.7566e-07	1.5214e-03	9.0056e-03	3.9269e-03



(a) Test function 1 (b) Test function 2

(c) Test function 3 (d) Test function 4

(e) Test function 5 (f) Test function 6

(g) Test function 7

Fig. 2. 0%, 50%, and 100% empirical attainment surfaces of all 25 runs for test functions 1–7 after 5e+5 FEs.

[6] H. A. Abbass, “The self-adaptive pareto differential evolution algorithm,” in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002 (CEC ’02)*, vol. 1, May 2002, pp. 831–836.

[7] N. K. Madavan, “Multiobjective Optimization Using a Pareto Differential Evolution Approach,” in *Congress on Evolutionary Computation (CEC’2002)*, vol. 2. Piscataway, New Jersey: IEEE Service Center, May 2002, pp. 1145–1150.

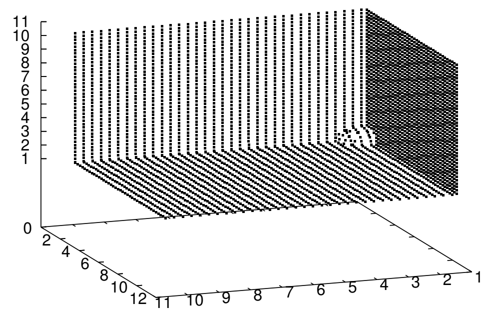
[8] B. Babu and M. M. L. Jehan, “Differential Evolution for Multi-Objective Optimization,” in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC’2003)*, vol. 4. Canberra, Australia: IEEE Press, December 2003, pp. 2696–2703.

[9] B. V. Babu, J. H. S. Mubeen, and P. G. Chakole, “Multiobjective Optimization Using Differential Evolution,” *TechGenesis – The Journal of Information Technology*, vol. 2, no. 2, pp. 4–12, 2005.

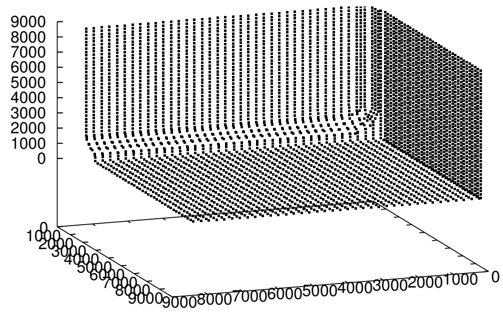
[10] F. Xue, A. C. Sanderson, and R. J. Graves, “Pareto-based Multi-Objective Differential Evolution,” in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC’2003)*, vol. 2. Canberra, Australia: IEEE Press, December 2003, pp. 862–869.

[11] S. Kukkonen and J. Lampinen, “GDE3: The third Evolution Step of Generalized Differential Evolution,” in *2005 IEEE Congress on Evolutionary Computation (CEC’2005)*, vol. 1. Edinburgh, Scotland: IEEE Service Center, September 2005, pp. 443–450.

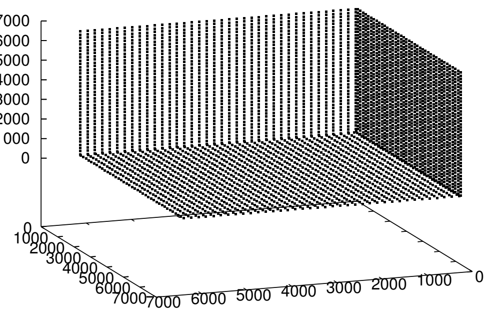
[12] V. L. Huang, P. N. Suganthan, A. K. Qin, and S. Baskar, “Multi-objective Differential Evolution with External Archive and Harmonic



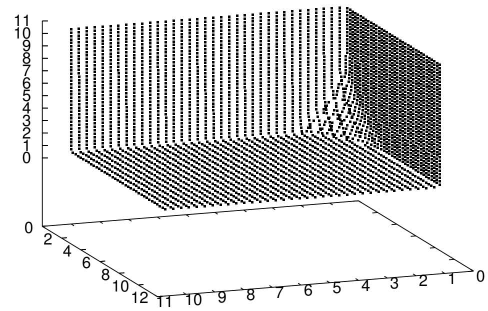
(a) Test function 8 (S_DTLZ2 with $M = 3$).



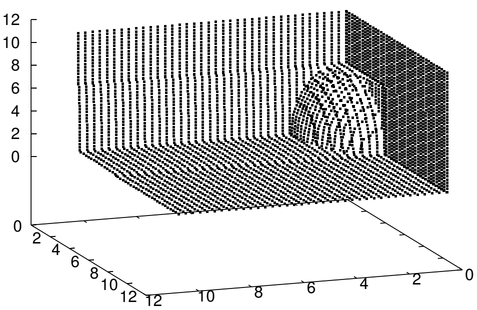
(b) Test function 9 (R_DTLZ2 with $M = 3$).



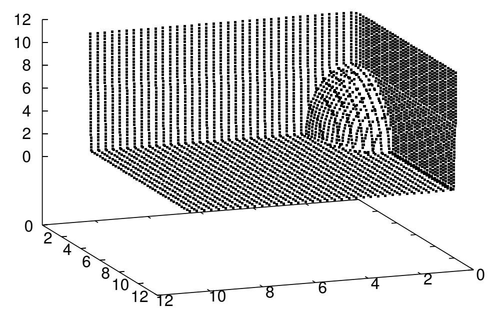
(c) Test function 10 (S_DTLZ3 with $M = 3$).



(d) Test function 11 (WFG1 with $M = 3$).



(e) Test function 12 (WFG8 with $M = 3$).



(f) Test function 13 (WFG9 with $M = 3$).

Fig. 3. 50% empirical attainment surfaces of all 25 runs for test functions 8–13 with $M = 3$ after 5e+5 FEs.

Distance-Based Diversity Measure," Nanyang Technological University, Singapore, Tech. Rep. TR-07-01, 2006.

[13] L. V. Santana-Quintero and C. A. Coello Coello, "An Algorithm Based on Differential Evolution for Multi-Objective Problems," *International Journal of Computational Intelligence Research*, vol. 1, no. 2, pp. 151–169, 2005.

[14] A. W. Iorio and X. Li, "Incorporating Directional Information within a Differential Evolution Algorithm for Multi-objective Optimization," in *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, M. K. et al., Ed., vol. 1. Seattle, Washington, USA: ACM Press. ISBN 1-59593-186-4, July 2006, pp. 691–697.

[15] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. C. Coello, R. Caballero, and J. Molina, "A new proposal for multi-objective optimization using differential evolution and rough sets theory," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation — GECCO 2006*, vol. 1, 2006, pp. 675–682.

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

[17] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds. Athens, Greece: International Center for Numerical Methods in Engineering (Cmine), 2001, pp. 95–100.

[18] E. Zitzler and S. Künzli, "Indicator-Based Selection in Multiobjective Search," in *Parallel Problem Solving from Nature (PPSN VIII)*, X. Yao *et al.*, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 832–842.

[19] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies — A comprehensive introduction," *Natural Computing*, vol. 1, pp. 3–52, 2002.

[20] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York: Oxford University Press, 1996.

[21] H.-P. Schwefel, *Evolution and Optimum Seeking*, ser. Sixth-Generation Computer Technology. New York: Wiley Interscience, 1995.

[22] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006, DOI:

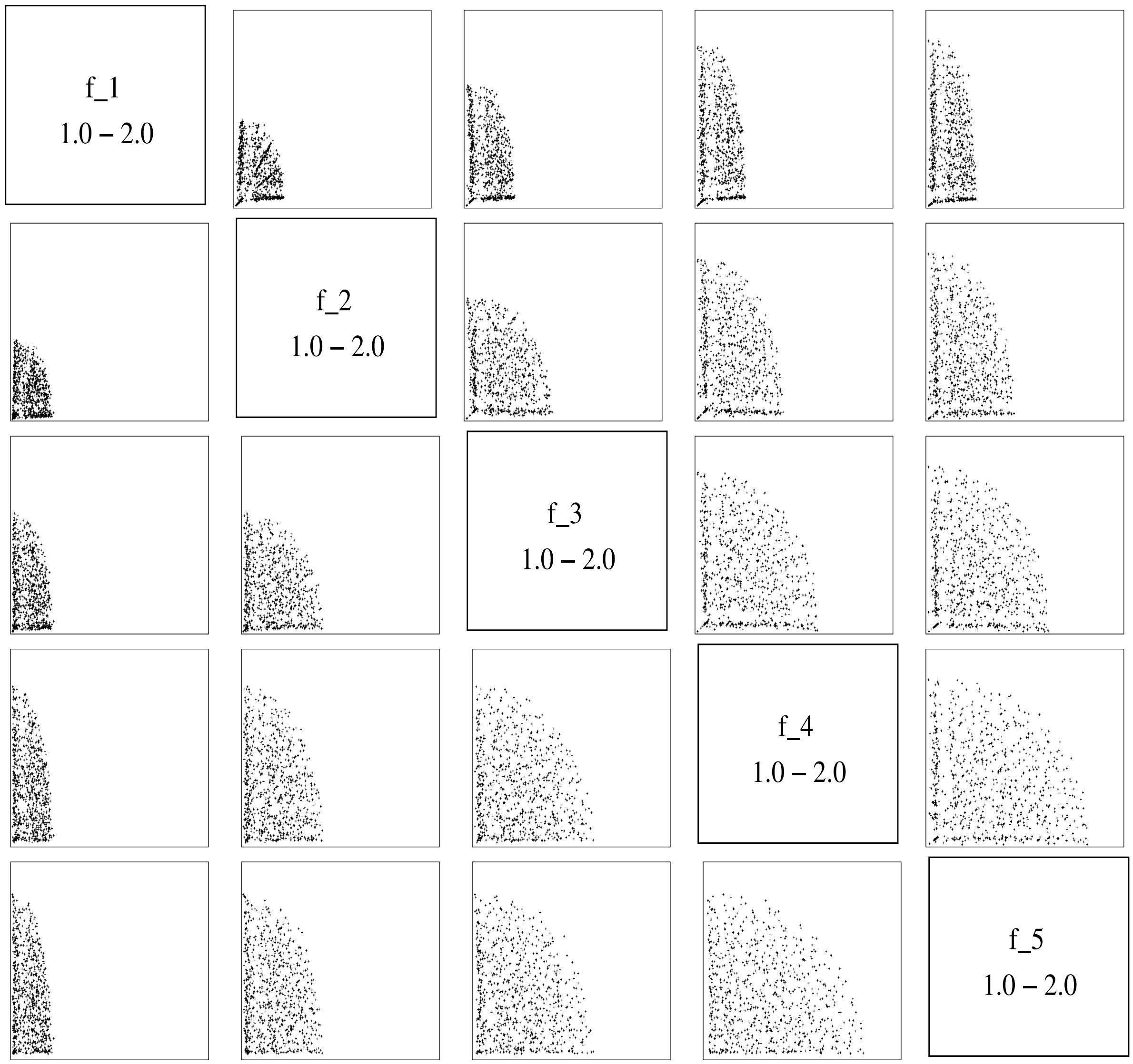


Fig. 4. Pareto front plots of the median approximation set with respect to the R indicator after 5e+5 FEs for test functions 12 and 13 with $M = 5$. Upper diagonal plots are for WFG8 (M=5) and lower diagonal plots are for WFG9 (M=5). The axes of any plot can be obtained by looking at the corresponding diagonal boxes and their ranges. For WFG8, the label in a column denotes the abscissa and the label in a row denotes the ordinate, and contrary for WFG9.

10.1109/TEVC.2006.872133.

[23] J. Brest, V. Žumer, and M. S. Maučec, "Self-adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization," in *The 2006 IEEE Congress on Evolutionary Computation CEC2006*. IEEE Press, 2006, pp. 919–926.

[24] V. L. Huang, A. K. Qin, K. Deb, E. Zitzler, P. N. Suganthan, J. J. Liang, M. Preuss, and S. Huband, "Problem Definitions for Performance Assessment & Competition on Multi-objective Optimization Algorithms," Nanyang Technological University et. al., Singapore, Tech. Rep. TR-07-01, 2007.

[25] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff, "On Test Functions for Evolutionary Multi-objective Optimization," in *PPSN*, 2004, pp. 792–802.

[26] G. Rudolph, B. Naujoks, and M. Preuss, "Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets," in *Evolutionary Multi-Criterion Optimization (EMO 2007)*, vol. 4403. Springer, 2007, pp. 36–50.

[27] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, Summer 2000.

[28] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems," in *Congress on Evolutionary Computation (CEC'2002)*, vol. 1. Piscataway, New Jersey: IEEE Service Center, May 2002, pp. 825–830.

[29] S. Huband, P. Hingston, L. Barone, and L. While, "A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, October 2006.

[30] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, "PISA — A Platform and Programming Language Independent Interface for Search Algorithms," in *Evolutionary Multi-Criterion Optimization (EMO 2003)*, ser. Lecture Notes in Computer Science, C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds. Berlin: Springer, 2003, pp. 494–508.