AN ALGORITHM BASED ON DIFFERENTIAL EVOLUTION FOR MULTIOBJECTIVE PROBLEMS

LUIS VICENTE SANTANA-QUINTERO AND CARLOS A. COELLO COELLO
CINVESTAV-IPN (Evolutionary Computation Group)
Electrical Engineering Department/Computer Science Section
Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07360, MÉXICO
lvspenny@hotmail.com, ccoello@cs.cinvestav.mx

ABSTRACT

This paper presents a new multi-objective evolutionary algorithm based on differential evolution. The proposed approach adopts a secondary population in order to retain the non-dominated solutions found during the evolutionary process. Additionally, the approach also incorporates the concept of ε-dominance to get a good distribution of the solutions retained. The main goal of this work was to keep the fast convergence exhibited by Differential Evolution in global optimization when extending this heuristic to multi-objective optimization. We adopted standard test functions and performance measures reported in the specialized literature to validate our proposal. Our results are compared with respect to another multi-objective evolutionary algorithm based on differential evolution (Pareto Differential Evolution) and with respect to two approaches that are representative of the state-of-the-art in the area: the NSGA-II and ε-MOEA.

1 INTRODUCTION

Most real-world problems involve the simultaneous optimization of two or more (often conflicting) objectives. The solution of such problems (called "multi-objective") is different from that of a single-objective optimization problem. The main difference is that multi-objective optimization problems normally have not one but a set of solutions which are all equally good. In the past, a wide variety of evolutionary algorithms (EAs) have been used to solve multi-objective optimization problems [4]. However, from the several types of EAs available, very few researchers have attempted to extend Differential Evolution (DE) [18] to solve multi-objective optimization problems. DE has been very successful in the solution of a variety of continuous (single-objective) optimization problems in which it has shown a great robustness and a very fast convergence. These are precisely the characteristics of DE that make it attractive to extend it to solve multi-objective optimization problems.

2 DIFFERENTIAL EVOLUTION

Differential Evolution [17, 18] is a relatively recent heuristic designed to optimize problems over continuous domains. In DE, each decision variable is represented in the chromosome by a real number. As in any other evolutionary algorithm, the initial population of DE is randomly generated, and then evaluated. After that, the selection process takes place. During the selection stage, three parents are chosen and they generate a single offspring which competes with a parent to determine who passes to the following generation. DE generates a single offspring (instead of two as the genetic algorithm) by adding the weighted difference vector between two parents to a third parent. In the context of single-objective optimization, if the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with respect to which it was compared. In addition, the best parameter vector $X_{best,G}$ is evaluated for every generation G in order to keep track of the progress that is made during the minimization process. More formally, the process is described as follows:

For each vector $\vec{x}_{i,G}$; i = 0, 1, 2,…, N-1 , a trial vector $\vec{v}$ generated according to:

$$\vec{v} = \vec{x}_{r1,G} + F \cdot (\vec{x}_{r2,G} - \vec{x}_{r3,G})$$

with r_1, r_2, $r_3 \in [0, N-1]$, integer and mutually different, and F > 0.

The integers r_1, r_2 and r_3 are chosen randomly from the interval [0, N-1] and are different from *i*. *F* is a real and constant factor which controls the amplification of the differential variation $(\vec{x}_{r2,G} - \vec{x}_{r3,G})$.

3 RELATED WORK

There have been several recent proposals to extend Differential Evolution to multi-objective optimization.

We will review next the most important of them:

- **PDE by Abbass in 2002 [1]:** It handles only one population, reproduction is undertaken only among non-dominated solutions, offspring are placed into the population if they dominate the main parent and a distance metric relation is used to maintain diversity.
- **PDEA by Madavan in 2002 [12]:** It combines the robust and effective DE strategy with key elements of the NSGA-II algorithm.
- **MODE by Xue in 2003 [22]:** in order to apply multi-objective optimization, the Pareto-based assignment and selection of the NSGA-II [5] is incorporated.
- **DE for Multi-Objective Optimization by Babu in 2003 [3]:** uses a penalty function and applies a weighting factor method to maintain diversity.
- **VEDE by Parsopoulus in 2004 [13]:** it is a parallel, multipopulation Differential Evolution algorithm, which is inspired by the Vector Evaluated Genetic Algorithm (VEGA) [15] approach.
- **GDE by Kukkonnen in 2004 [9]:** is an extension of Generalized Differential Evolution for constrained multi-objective optimization.

- **NSDE by Iorio in 2004 [7]:** it is a simple modification to the NSGA-II [5] where the real-coded crossover and mutation rates have been replaced with a Differential Evolution scheme.
- **DEMO by Robic in 2005 [14]:** Combines the advantages of DE with the mechanisms of Pareto-based ranking and crowding distances sorting. In DEMO, the newly created candidates take part immediately in the creation of the subsequent candidates.

Algorithm 1 Proposed Algorithm: ε- MyDE

```
Initialize vector of the population P
Evaluate the cost of each vector
for $i$ = 0 to G do
    repeat
        Select three distinct vectors randomly
        Perform crossover using DE scheme
        Perform mutation
        Evaluate objective values
        if offspring is better than main parent then
            replace it on population
        end if
    until population is completed
    Identify nondominated solutions in population
    Add nondominated solutions into secondary population
end for
```

4 OUR PROPOSED APPROACH

The pseudo-code of our proposed approach (called ε-MyDE) is shown in Algorithm 1. Our approach keeps two populations: the main population (which is used to select the parents) and a secondary (external) population, in which we adopt the concept of ε-dominance to retain the non-dominated solutions found and to distribute them in a uniform way.

ε-MyDE uses real numbers representation, where each chromosome is a vector of real numbers (each number corresponds to a decision variable of the problem). We also incorporate a constraint-handling mechanism that allows infeasible solutions to intervene during recombination. This helps to solve in a more efficient way highly constrained multiobjective optimization problems.

Our approach has two selection mechanisms that are activated based on the total number of generations and a parameter called $sel_2 \in (0.2 - 1)$, which regulates the selection pressure. For example, if $sel_2 = 0.6$ and the total number of generation is $G_{max} = 200$, this means that during the first 120 generations (60% of G_{max}), a random selection will be adopted, and during the last 80 generations an elitist selection will be adopted. In both selections (random and elitist), a single parent is selected as reference. This parent is used to compare the offspring generated by three different parents. This mechanism guarantees that all the parents of the main population will be reference parents for only one time during the generating process. Both types of selection are described next:

1 Random Selection. - 3 different parents are randomly selected from the main population.
2 Elitist Selection. - 3 different parents are selected from the secondary population. It is required that these 3 parents are close from each other. If no set of 3 individuals exists that fulfills this requirement, then a set of 3 individuals is randomly selected from the secondary population. The expression adopted to determine closeness is the following:

$$f_{close} = \frac{\sqrt{\sum_{i=0}^{FUN} \left(X_{i,max} - X_{i,min}\right)^2}}{2^{FUN}}$$

where:
FUN = number of objective functions
$X_{i,max}$ = upper bound of the i-th objective function in the secondary population
$X_{i,min}$ = lower bound of the i-th objective function in the secondary population

Recombination in our approach is performed using the following procedure. For each parent vector $\vec{p}_i$; $i = 0, 1, 2, \ldots, P - 1$ (P = population), the offspring vector $\vec{h}$ is generated as:

$$(x \in U(0,1)) \begin{cases} h_j = p_{r1,j} + F \cdot (p_{r2,j} - p_{r3,j}), & \text{if } x < p_{crossover}; \\ h_j = p_{ref,j}, & \text{otherwise.} \end{cases} \qquad (1)$$

where: $j = 0, 1, 2, \ldots, var - 1$. (var = number of variables for each solution vector) $p_{r1}, p_{r2}, p_{r3} \in [0, P - 1]$, are integers and mutually different, $F > 0$. The integers r_1, r_2 and r_3 are the indices of the selected parents randomly chosen from the interval [0, N – 1] and *ref* is the index of the reference parent. F is a real and constant factor which controls the amplification of the differential variation $(p_{r2,j} - p_{r3,j})$.

Differential evolution doesn't use a specific mutation operator, since such operator is somehow embedded in its recombination. However, in multi-objective optimization problems, we found it necessary to provide an additional mutation operator to allow a better exploration of the search space (mainly in constrained problems). We adopted uniform mutation for that sake.

Once a child has been generated, it is compared with respect to the reference parent, against which it competes to determine who passes to the following generation. The rules of comparison between a child and its parent are the following:

Version for unconstrained problems

- *If parent dominates child*, the parent is chosen
- *If child dominates parent*, the parent is chosen
- *If both are nondominated with respect to each other*, perform a *flip*(0.5) to determine who passes to the following generation.

Version for constrained problems

- *If parent is infeasible and child is infeasible*, the solution that is closest to the feasible region is selected.

- *If parent is feasible and child is infeasible*, the child is chosen if and only if the child is at least at a distance of 0.1 of the feasible region and a *flip* (0.5) returns true. Otherwise, the father is chosen.

- *If parent is infeasible and child is feasible*, the parent is chosen if and only if the parent is at least at a distance of 0.1 of the feasible region and a *flip* (0.5) returns true. Otherwise, the child is chosen.

- *If parent is feasible and child is feasible*, Pareto dominance is verified between them.

Note that the scheme previously described allows some infeasible solutions to intervene during recombination. We found that this sort of scheme is particularly useful when dealing with highly constrained problems.

For producing a well-distributed set of non-dominated solutions, we adopted a relaxed form of dominance called *ε-dominance*, which is defined as follows [11]:

Let $f, g \in R^m$, then $\vec{f}$ is said to ε-dominate $\vec{g}$ for some $\varepsilon > 0$, denoted as $f \prec_{\varepsilon} g$, if and only if for all $i \in \{1, \ldots, m\}$.

$$(1-\varepsilon) \cdot f_i \leq g_i$$

As indicated before, our proposed approach uses an external archive (also called secondary population). In order to include a solution into this archive, it is compared with respect to each member already contained in the archive using ε-dominance. The procedure is described next.

Every solution in the archive is assigned an identification array ($\mathbf{B} = (B_1, B_2, \ldots, B_M)^T$, where M is the total number of objectives) as follows:

$$B_j(f) = \begin{cases} (\lfloor (f_j - f_j^{min})/\epsilon_j \rfloor), & \text{for minimizing} f_j; \\ (\lceil (f_j - f_j^{min})/\epsilon_j \rceil), & \text{for maximizing } f_j. \end{cases}$$

where: $f_j^{\min}$ is the minimum possible value of the j-th objective and ε_j is the allowable tolerance in the j-th objective [11]. The identification array divides the whole objective space into hyper-boxes, each having ε_j size in the j-th objective. With the identification arrays calculated for the offspring c_1 and each archive member a, we use the procedure illustrated in Figure 1 and described next:

1.- If the identification array $\mathbf{B_a}$ of any archive member a dominates that of the offspring c_i, then it means that the offspring is ε-dominated by this archive member and so the offspring is *not accepted*. Case (a) in Figure 1.

2.- If $\mathbf{B_{C\mathit{i}}}$ of the offspring dominates the $\mathbf{B_a}$ of any archive member a, the archive member is deleted and the offspring *accepted*. Case (b) in Figure 1.

If neither of the above two cases occur, then it means that the offspring is ε-non-dominated with respect to the archive contents. There are two further possibilities in this case:

(a) If the offspring shares the same **B** vector with an archive member (meaning that they belong to the same hyper-box), then they are first checked for the usual non-domination. If the offspring dominates the archive member or the offspring is non-dominated with respect to the archive member but is closer to the **B** vector (in terms of the Euclidian distance) than the archive member, then the offspring *is retained*. This is case (c) in Figure 1.

(b) In the event of an offspring not sharing the same B vector with any archive member, the offspring *is accepted*. Case (d) in Figure 1.

Using the above procedure, we can guarantee the generation of a well-distributed set of non-dominated solutions. Also, the value of ε regulates the size of the external archive. Thus, there is no need to pre-fix an upper limit on the size of the archive as done in most traditional multi-objective evolutionary algorithms.

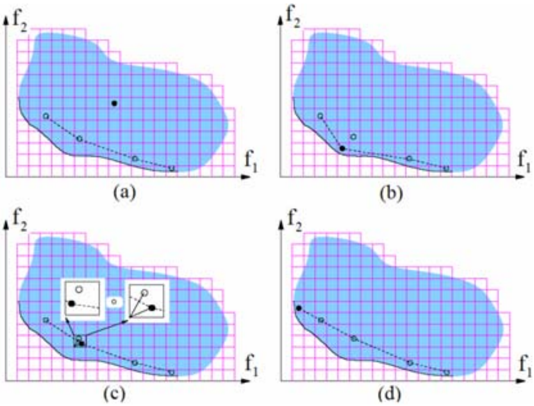

Figure 1: Four cases of inserting a child into the external archive

5 COMPARISON OF RESULTS

In order to validate our proposed approach, our results are compared with respect to those generated by another multi-objective evolutionary algorithm based on Differential Evolution: PDE [2]. Additionally, we also compared results with respect to two approaches representative of the state-of-the-art in the area: the NSGA-II [5] and ε-MOEA [6]. The parameters adopted for all algorithms compared are presented in Table 1. Also, we adopted 4 metrics reported in the specialized literature: Error Ratio (ER) [19], Generational Distance (GD) [20, 21], Spacing (S) [16] and Set Coverage (SC) [23].

Parameter	ε-MyDE	NSGA-II	PDE	ε-MOEA
Population	100	100	100	100
External Pop.	100 (aprox.)	100	100	100 (aprox.)
G_{max}	50 or 250	50 or 250	50 or 250	50 or 250
P_c	0.95	0.8	Nr	1.0
P_m	1 / N	1 / N	Nr	1/N
F	0.5	Nr	N(0, 1)	Nr

where: N = variable number , d = objective number, nr = not required
G_{max} = 50 for unconstrained problems, G_{max} = 250 for constrained problems

Table 1: Parameters adopted for all algorithms compared

[MOP1] Proposed by Kursawe [10]. Minimize:

$$f_1(\vec{x}) = \sum_{i=1}^{n-1} -10e^{(-0.2)\sqrt{x_i^2+x_{i+1}^2}} \;\; ; \;\& \; f_2(\vec{x}) = \sum_{i=1}^{n}(|x_i|^a + 5\sin(x_i^b))$$

with $a = 0.8$, $b = 3$; $n = 3$ and: $-5 \leq x_1, x_2, x_3 \leq 5$

Graphically, in MOP1 we can observe that ε-MyDE and ε-MOEA are the only approaches able to reach the true Pareto front of the problem (the NSGA-II cannot generate the extremes of the front). Regarding spacing, all the approaches seem to have a good distribution of solutions. When comparing the values of the performance metrics, it is clear that our ε-MyDE obtained the best results with respect to both Error Ratio and Generational Distance. Regarding Spacing, the NSGA-II was slightly better than the other approaches. With respect to Set Coverage, our ε-MyDE had the best results, which indicates that it is the approach that dominates more solutions from the other algorithms and it is (at the same time) less dominated by the others.

[MOP2] Proposed by Kita [8]. Maximize:

$$f_1(x, y) = -x^2 + y; \;\& \; f_2(x, y) = \frac{1}{2}x + y + 1$$

subject to:

$$0 \geq \frac{1}{6}x + y - \frac{13}{2}; \; 0 \geq \frac{1}{2}x + y - \frac{15}{2}; 0 \geq 5x + y - 30 \text{ and} : 0 \leq x, y$$

Metric – Algorithm		average	Best	worst	St. Dev
ER	ε – MyDE	**0.157920**	**0.06542**	**0.23301**	**0.0428870**
	NSGA-II	0.4575	0.31	0.64	0.090728
	PDE	0.5085	0.25	0.64	0.095821
	ε – MOEA	0.2820835	0.16326	1	0.177830
GD	ε – MyDE	**0.003270**	**0.002888**	**0.003744**	**0.000258**
	NSGA-II	0.004166	0.003080	0.006163	0.000861
	PDE	0.004675	0.003668	0.006670	0.000966
	ε – MOEA	0.09699	0.003047	1.87613	0.418764
S	ε – MyDE	0.094489	0.054854	**0.109133**	**0.012435**
	NSGA-II	**0.061808**	**0.047978**	0.11722	0.016637
	PDE	0.105224	0.068795	0.145438	0.224935
	ε – MOEA	0.101133	0.0564897	0.126934	0.013587

SC	ε – MyDE	NSGA-II	PDE	ε – MOEA	Dominates
ε – MyDE		0.824752	0.825743	0.615924	**0.755473**
NSGA-II	0.234375		0.633663	0.375063	0.414367
PDE	0.238511	0.676733		0.408112	0.4411186
ε – MOEA	0.323529	0.712871	0.729703		0.588701
They are dominated	**0.26547167**	0.73811867	0.729703	0.46636633	

Table 2: Statistical results corresponding to the metrics adopted for MOP1: Error Ratio(ER), Generational Distance (GD), Spacing (S) and Set Coverage (SC).

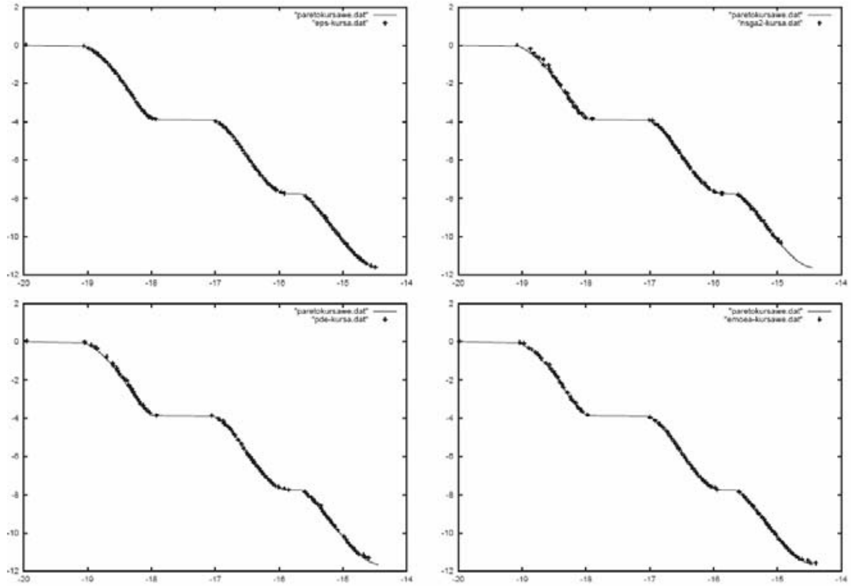

Figure 2: Pareto fronts generated by ε-MyDE (top-left), NSGA-II (top-right), PDE (bottom-left) and ε-MOEA (bottom-right) for MOP1.

Graphically, in MOP2 we can observe that all the algorithms get a reasonably good approximation of the true Pareto front. However, our ε-MyDE was the only approach able to cover the entire Pareto front and of obtaining a good distribution of solutions. When comparing the values of the performance metrics, our ε-MyDE obtained the best results with respect to Error Ratio and Generational Distance. Regarding Spacing, ε-MOEA obtained the best results. However, regarding Set Coverage, our ε-MyDE obtained again the best results. This means that our approach dominated the larger number of solutions produced by the other algorithms and, at the same time, its solutions were the least dominated by the other algorithms.

Metric – Algorithm		average	best	worst	St. Dev
ER	ε – MyDE	**0.016406**	**0**	**0.055045**	**0.016353**
	NSGA-II	0.316	0.19	0.53	0.093942
	PDE	0.8395	0.35	1	0.212119
	ε – MOEA	0.250085	0.15625	0.333333	0.044541
GD	ε – MyDE	**0.004912**	0.001457	0.046416	0.009799
	NSGA-II	0.014703	0.001367	0.090251	0.027298
	PDE	0.005768	**0.000387**	**0.032941**	**0.0076951**
	ε – MOEA	0.006311	0.002205	0.047624	0.010411
S	ε – MyDE	0.073633	0.039430	0.492726	0.099053
	NSGA-II	0.092523	0.007056	0.56773	0.168849
	PDE	**0.017289**	**3.606E-05**	**0.2736**	**0.061153**
	ε – MOEA	0.077128	0.045864	0.492715	0.099079

SC	ε – MyDE	NSGA-II	PDE	ε – MOEA	Dominates
ε – MyDE		0.806436	0.928498	0.879053	**0.871329**
NSGA-II	0.042775		0.666667	0.62069	0.443377
PDE	0.005228	0.096534		0.076170	0.059311
ε – MOEA	0.039924	0.55099	0.951132		0.514015
They are dominated	**0.029309**	0.484653	0.848765	0.525304	

Table 3: Statistical results corresponding to the metrics adopted for MOP2: Error Ratio(ER), Generational Distance (GD), Spacing (S) and Set Coverage (SC)

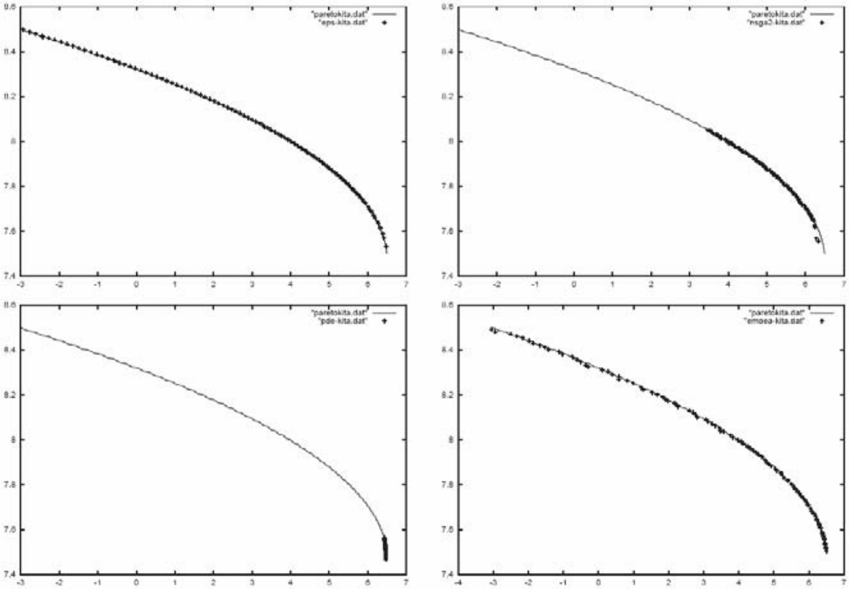

Figure 3: Pareto fronts generated by ε-MyDE (top-left), NSGA-II (top-right), PDE (bottom-left) and ε-MOEA (bottom-right) for MOP2.

6 CONCLUSIONS AND FUTURE WORK

We have introduced an approach that uses Differential Evolution to solve both unconstrained and constrained multi-objective optimization problems. The high convergence rate (which can produce premature convergence) that characterizes the Differential Evolution algorithm was controlled using two forms of elitist selection. Our approach also adopts the concept of ε-dominance to produce well-distributed sets of non-dominated solutions. Additionally, our approach incorporates a novel constraint-handling mechanism that allows some infeasible solutions to intervene in the selection process. Our proposed approach was validated using standard test functions and performance metrics reported in the specialized literature. In our study, we compared results with respect to the NSGA-II, to Pareto Differential Evolution (PDE) and with respect to ε-MOEA.

As part of our future work, we intend to develop a self-adaptation scheme that makes unnecessary the manual fine-tuning of parameters for our approach and we also intend to devise a self-adaptive mechanism to calculate the value of the vector ε used to generate a specific number of nondominated solutions.

ACKNOWLEDGMENTS

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at CINVESTAV-IPN. The second author acknowledges support from CONACyT project number 45683.

7 REFERENCES

[1] Hussein A. Abbass. The Self-Adaptive Pareto Differential Evolution Algorithm. In *Congress on Evolutionary Computation (CEC'2002)*, vol. 1, pp. 831-836, May 2002.

[2] Hussein A. Abbass and Ruhul Sarker. The Pareto Differential Evolution Algorithm. *International Journal on Artificial Intelligence Tools*, 11(4):531-552, 2002.

[3] B.V. Babu and M. Mathew Leenus Jehan. Differential Evolution for Multi-Objective Optimization. In *CEC'2003*, vol. 4, pp. 2696-2703, Canberra, Australia, December 2003.

[4] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.

[5] Kalyanmoy Deb, Amrit Pratab, Samir Agrawal and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, April 2002.

[6] Kalyanmoy Deb, Manikanth Mohan, and Shikhar Mishra. Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. In Carlos M. Fonseca et al., editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pp. 222-236, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[7] Antony W. Iorio and Xiaodong Li. Solving rotated multi-objective optimization problems using differential evolution. In *AI 2004: Advances in Artificial Intelligence,* pp. 861-872. Springer-Verlag, Lecture Notes in Artificial Intelligence Vol. 3339, 2004.

[8] Hajime Kita, Yasuyuki Yabumoto, Naoki Mori, and Yoshikazu Nishikawa. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In Hans-Michael Voigt et al., editors, *PPSN IV,* LNCS Vol. 1141, pp. 504-512, 1996. Springer- Verlag.

[9] Saku Kukkonen and Jouni Lampinen. An Extension of Generalized Differential Evolution for Multi-objective Optimization with Constraints. In *Parallel Problem Solving from Nature - PPSN VIII*, pp. 752-761, September 2004. Springer-Verlag. LNCS Vol. 3242.

[10] Frank Kursawe. A Variant of Evolution Strategies for Vector Optimization. In H. P. Schwefel and R. Mäanner, editors, *Parallel Problem Solving from Nature, PPSN I,* volume 496 of *LNCS Vol. 496*, pp. 193-197, October 1991. Springer-Verlag.

[11] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263-282, Fall 2002.

[12] Nateri K. Madavan. Multiobjective Optimization Using a Pareto Differential Evolution Approach. In *CEC'2002*, vol. 2, pp. 1145-1150, May 2002. IEEE Service Center.

[13] K.E. Parsopoulos, D.K. Taoulis, N.G. Pavlidis, V.P. Plagianakos, and M.N. Vrahatis. Vector Evaluated Differential Evolution for Multiobjective Optimization. In *CEC'2004*, vol. 1, pp. 204-211, Portland, Oregon, USA, June 2004. IEEE Service Center.

[14] Tea Robič and Bodgan Filipič. DEMO: Differential Evolution for Multiobjective Optimization. In C.A. Coello Coello et al. editors, *EMO 2005*, pp. 520-533, March 2005.

[15] J. David Schaffer *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.* PhD thesis, Vanderbilt University, 1984.

[16] Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, MIT, Cambridge, Massachusetts, May 1995.

[17] R. Storn and K. Price. Differential evolution - a simple and efficient adaptative scheme for global optimization over continuous spaces. Technical Report TR-95-12, March 1995.

[18] R. Storn and Kenneth Price. Differential Evolution - a fast and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, (11):341-359.

[19] David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[20] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report TR-98-03, Ohio, 1998.

[21] David A. Van Veldhuizen and Gary B. Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. In *CEC'2000*, vol. 1, pp. 204-211, 2000. IEEE Service Center.

[22] Feng Xue, Arthur C. Sanderson, and Robert J. Graves. Pareto-based Multi-Objective Differential Evolution. In *CEC'2003*, vol. 2, pp. 862-869, December 2003. IEEE Press.

[23] Eckart Zitzler, K. Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation,* 8(2): 173-195, Summer 2000.