

Accelerating Convergence using Rough Sets Theory for Multi-Objective Optimization Problems

Luis V. Santana-Quintero
CINVESTAV-IPN
Computer Science Department
Mexico City, MÉXICO
lvspenny@hotmail.com

Carlos A. Coello Coello
CINVESTAV-IPN
Computer Science Department
Mexico City, MÉXICO
ccoello@cs.cinvestav.mx

ABSTRACT

We propose the use of rough sets theory to improve the first approximation provided by a multi-objective evolutionary algorithm and retain the nondominated solutions using a new adaptive grid based on the ϵ -dominance concept that tries to overcome the main limitation of ϵ -dominance: the loss of several nondominated solutions from the hypergrid adopted in the archive because of the way in which solutions are selected within each box. We decided to use a multi-objective version of differential evolution to build a first approximation of the Pareto front and in a second stage, we use the rough sets theory in order to improve the spread of the solutions found so far. To assess our proposed hybrid approach, we adopt a set of standard test functions and metrics taken from the specialized literature. Our results are compared with respect to the NSGA-II, which is an approach representative of the state-of-the-art in the area.

Categories and Subject Descriptors: G.1.6 [Optimization]: Unconstrained optimization; I.2.8 [Problem Solving, Control Methods and Search]: Heuristic Methods

General Terms: Algorithms

Keywords: Multi-objective optimization, rough sets theory, local search

1. INTRODUCTION

Multi-Objective Optimization (MOO) is a research field that has raised great interest over the last thirty years, mainly because of the many real-world problems which naturally have several (often conflicting) criteria to be simultaneously optimized [6].

In recent years, a wide variety of multi-objective evolutionary algorithms (MOEAs) have been proposed in the specialized literature [2, 3]. Our main motivation for proposing a hybrid approach is to reduce the overall number of fitness function evaluations performed to approximate the true Pareto front of a problem. Our proposed hybrid is able to produce reasonably good approximations of the Pareto

front of a variety of problems of different complexity with only 3000 fitness function evaluations.

The organization of the rest of the paper is the following. Some basic definitions related to MOO are in Section 2. An introduction to rough sets theory is provided in Section 3. In Section 4, we introduce differential evolution, which is the approach adopted as our search engine. Section 5 describes the new adaptive grid that we adopted to retain nondominated solutions (called Pareto-adaptive ϵ -dominance). Our proposed hybrid is described in Section 6. The experimental setup adopted to validate our approach and the corresponding discussion of results are provided in Section 7. Finally, our conclusions and some possible paths for future research are provided in Section 8.

2. MULTI - OBJECTIVE OPTIMIZATION

We are interested in solving problems of the type¹:

Minimize $\vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]$

subject to:

$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \dots, m$ and $h_j(\vec{x}) = 0 \quad j = 1, 2, \dots, p$

where $\vec{x} = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, $j = 1, \dots, p$ are the constraint functions of the problem. To describe the concept of optimality in which we are interested, we will introduce next a few definitions.

Definition 1. Given two vectors $\vec{x}, \vec{y} \in \mathbb{R}^k$, we say that $\vec{x} \leq \vec{y}$ if $x_i \leq y_i$ for $i = 1, \dots, k$, and that \vec{x} **dominates** \vec{y} (denoted by $\vec{x} \prec \vec{y}$) if $\vec{x} \leq \vec{y}$ and $\vec{x} \neq \vec{y}$.

Definition 2. We say that a vector of decision variables $\vec{x} \in \mathcal{X} \subset \mathbb{R}^n$ is **nondominated** with respect to \mathcal{X} , if there does not exist another $\vec{x}' \in \mathcal{X}$ such that $\vec{f}(\vec{x}') \prec \vec{f}(\vec{x})$.

Definition 3. We say that a vector of decision variables $\vec{x}^* \in \mathcal{F} \subset \mathbb{R}^n$ (\mathcal{F} is the feasible region) is **Pareto-optimal** if it is nondominated with respect to \mathcal{F} .

Definition 4. The **Pareto Optimal Set** \mathcal{P}^* is defined by:

$$\mathcal{P}^* = \{\vec{x} \in \mathcal{F} | \vec{x} \text{ is Pareto-optimal}\}$$

Definition 5. The **Pareto Front** \mathcal{PF}^* is defined by:

$$\mathcal{PF}^* = \{\vec{f}(\vec{x}) \in \mathbb{R}^k | \vec{x} \in \mathcal{P}^*\}$$

We thus wish to determine the Pareto optimal set from the set \mathcal{F} of all the decision variable vectors that satisfy the constraint functions of the problem.

¹Without loss of generality, we will assume only minimization problems.

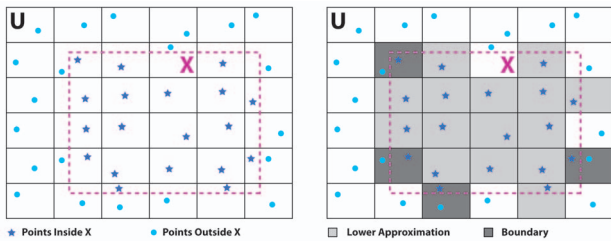


Figure 1: Rough sets approximation

3. ROUGH SETS THEORY

Rough Sets theory is a new mathematical approach to deal with imperfect knowledge. Rough sets theory was proposed by Pawlak [9]. Let's assume that we are given a set of objects U called the *universe* and an indiscernibility relation $R \subseteq U \times U$. Let X be a subset of U . The problem is that we want to characterize the set X with respect to R . The way rough sets theory expresses vagueness is employing a boundary region of the set X built once we know points both inside X and outside X . If the boundary region of a set is empty it means that the set is *crisp*; otherwise, the set is *rough* (inexact).

4. DIFFERENTIAL EVOLUTION (DE)

Differential Evolution [12, 13] is a relatively recent evolutionary algorithm designed to optimize problems over continuous domains. In DE, each decision variable is represented 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 by adding the weighted difference vector between two parents to a third parent. More formally, the process is described as follows:

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

$$\vec{v} = \vec{x}_{r_1,G} + F \cdot (\vec{x}_{r_2,G} - \vec{x}_{r_3,G})$$

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 randomly chosen 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 $(\vec{x}_{r_2,G} - \vec{x}_{r_3,G})$. DE has been extended to solve multi-objective problems by several researchers (see for example [1, 8, 10]). However, in such extensions, DE has been found to be very good at converging close to the true Pareto front (i.e., for coarse-grained optimization), but not so efficient to actually reaching the front (i.e., for fine-grained optimization).

5. PARETO-ADAPTIVE ϵ -DOMINANCE

ϵ -dominance has been mainly used as an archiving strategy in which one can regulate the resolution at which our approximation of the Pareto front will be generated. This allows us to accelerate convergence (if a very coarse resolution is sufficient) or to improve the quality of our approximation (if we can afford the extra computational cost). However,

ϵ -dominance has certain drawbacks and limitations. For example: (1) we can lose a high number of nondominated solutions if the decision maker does not take into account (or does not know) the geometrical characteristics of the true Pareto front, (2) the extrema of the Pareto front are normally lost and (3) the upper bound for the number of points allowed by a grid is not easy to achieve in practice. In order to overcome some of these limitations, the concept of *pa* ϵ -dominance was proposed in [7]. Briefly, the main difference is that in *pa* ϵ -dominance the hyper-grid generated adapts the sizes of the boxes to certain geometrical characteristics of the Pareto front (e.g., almost horizontal or vertical portions of the Pareto front) to increase the number of solutions retained in the grid. This scheme maintains the good properties of ϵ -dominance but improves on its main weaknesses. In [7], it is empirically shown that the advantages of *pa* ϵ -dominance over ϵ -dominance make it a more suitable choice to be incorporated into a MOEA and therefore our decision of adopting this scheme for the work reported in this paper.

6. THE HYBRID METHOD: DEMORS

Our proposed approach, called DEMORS (Differential Evolution for Multi-objective Optimization with Rough Sets) is divided in two different phases, and each of them consumes a fixed number of fitness function evaluations. During Phase I, our DE-based MOEA is applied for 2000 fitness function evaluations. During Phase II, a local search procedure based on rough sets theory is applied for 1000 fitness function evaluations, in order to improve the solutions produced at the previous phase. Each of these two phases is described next.

Phase I : Use of Differential Evolution

The pseudo-code of our proposed DE-based MOEA is shown in Algorithm 1 [11]. Our approach keeps three populations: the main population (which is used to select the parents), a secondary (external) population, which is used to retain the nondominated solutions found and a third population that retains dominated solutions removed from the second population.

Algorithm 1 Phase I pseudo-code

```

1: Initialize vectors of the population  $P$ 
2: Evaluate the cost of each vector
3: for  $i = 0$  to  $G$  do
4:   repeat
5:     Select (randomly) three different vectors
6:     Perform crossover using DE scheme
7:     Perform mutation
8:     Evaluate objective values
9:     if offspring is better than main parent then
10:      replace it on population
11:   end if
12: until population is completed
13: Identify nondominated solutions in population
14: Add nondominated solutions into secondary population
15: Add dominated solutions into third population
16: end for

```

First, we randomly generate 25 individuals, and use them to generate 25 offspring. Phase I has two selection mechanisms that are activated based on the total number of generations and a parameter called $sel_2 \in [0, 1]$, which regulates the selection pressure. Both types of selection and recombination operators are described in [11]. In both selections (random and elitist), a single parent is selected as reference. This reference parent is used to compare the offspring generated by the three different parents. Differential evolution

does not use a specific mutation operator, since such operator is somehow embedded within its recombination operator. However, in multi-objective optimization problems, we found it necessary to provide an additional mutation operator in order to allow a better exploration of the search space. We adopted uniform mutation for that sake.

Phase II : Local Search using Rough Sets

Upon termination of Phase I, we start Phase II, which departs from the nondominated set generated in Phase I (ES). This set is contained within the secondary population. We also have the dominated set (DS), which is contained within the third population. We want to create a grid to describe the set ES in order to intensify the search on it. This is, we want to describe the Pareto front in decision variable space because then we could easily use this information to generate more efficient points and improve this initial approximation.

Algorithm 2 Phase II pseudo-code

```

1:  $ES \leftarrow$  nondominated set generated by Phase I
2:  $DS \leftarrow$  dominated set generated by Phase I
3:  $eval \leftarrow 0$ 
4: repeat
5:    $Items \leftarrow NumEff$  points  $\in ES$  &  $NumDom$  points  $\in DS$ 
6:   Range Initialization
7:   Compute Atoms
8:   for  $i \leftarrow 0$ , Offspring do
9:      $eval \leftarrow eval + 1$ 
10:     $ES \leftarrow Offspring$  generated
11:    Add Offspring into  $ES$  set
12:  end for
13: until  $1000 < eval$ 

```

From the set ES we choose $NumEff$ points. Next, we choose from the set DS $NumDom$ points previously unselected. These points will be used to approximate the boundary between the Pareto front and the rest of the feasible set in decision variable space. What we want to do now is to intensify the search in the area where the nondominated points reside, and refuse finding more points in the area where the dominated points reside. For this purpose, we store these points in the set $Items$ and perform a rough sets iteration: (1) **Range Initialization:** We build a (non-uniform) grid in decision variable space. (2) **Compute Atoms:** We compute $NumEff$ rectangular atoms centered in the $NumEff$ efficient points. (3) **Generate Offspring:** Inside each atom we randomly generate $Offspring$ new points. Each of these points are sent to the set ES (with a pac -dominance grid) to check if it must be included as a new nondominated point.

7. COMPUTATIONAL EXPERIMENTS

In order to validate our proposed approach, our results are compared with respect to those generated by the NSGA-II [4], which is a MOEA representative of the state-of-the-art in the area. Our approach was validated using 6 test problems: three from the **ZDT** set [14] and other three from **DTLZ** set [5]. In all cases, the parameters of our approach were set as follows: $Pc = 0.3$, $sel_2 = 0.3$, $Pop = 25$, $Offspring = 1$, $NumEff = 2$ and $NumDom = 10$. The NSGA-II was used with the following parameters: crossover rate = 0.9, mutation rate = $1/num_variables$, $\eta_c = 15$, $\eta_m = 20$, population size = 100 and maximum number of generations = 30. The population size of the NSGA-II is the same as the size of the grid of our approach, in order to allow a fair comparison of results, and both approaches adopted real-numbers encoding and performed 3000 fitness function evaluations.

Funct	SSC				$I_{\varepsilon+}^1$			
	DEMORS		NSGA-II		DEMORS		NSGA-II	
	Mean	σ	Mean	σ	Mean	σ	Mean	σ
ZDT1	0.852	0.001	0.635	0.021	0.006	0.001	0.193	0.022
ZDT2	0.794	0.014	0.555	0.032	0.031	0.036	0.342	0.053
ZDT3	0.788	0.002	0.647	0.025	0.017	0.006	0.154	0.020
DTLZ1	0.997	0.0007	0.996	0.002	0.023	0.007	0.046	0.009
DTLZ2	0.941	0.0017	0.930	0.004	0.067	0.008	0.079	0.015
DTLZ3	0.996	0.0006	0.996	0.004	0.042	0.018	0.060	0.014

Function	SDC			
	DEMORS		NSGA-II	
	Mean	σ	Mean	σ
ZDT1	0.008	0.004	0.051	0.010
ZDT2	0.033	0.026	0.159	0.041
ZDT3	0.091	0.016	0.073	0.005
DTLZ1	0.096	0.013	0.040	0.018
DTLZ2	0.026	0.011	0.007	0.007
DTLZ3	0.110	0.036	0.043	0.016

Table 1: Comparison of results between our DEMORS and the NSGA-II for the ZDT and DTLZ problems adopted.

Three performance measures were adopted in order to allow a quantitative assessment of our results: (1) Size of the space covered (SSC), proposed by Zitzler and Thiele [15]; (2) Unary additive epsilon indicator ($I_{\varepsilon+}^1$), introduced by Zitzler et al. [16]; and (3) Standard Deviation of Crowding Distances (SDC) which measures the distribution of vectors in the approximation set. A perfect distribution, that is $SDC = 0$, means that the solutions are at the same crowding distance (see [3] for more details on this distance).

Discussion of Results.

Table 1 shows a summary of our results. For each test problem, we performed 30 independent runs per algorithm. The results reported in Table 1 are the mean values for each of the three performance measures and the standard deviation of the 30 runs performed. The best mean values in each case are shown in **boldface**.

It can be seen in Table 1 that our DEMORS produced the best mean values in all cases. The graphical results shown in Figure 2 serve to reinforce our argument of the superiority of the results obtained by our DEMORS. These plots correspond to the run in the mean value with respect to the unary additive epsilon indicator. The spread of solutions of our DEMORS is evidently not the best possible, but the quality of the spread of solutions is sacrificed at the expense of reducing the computational cost required to obtain a good approximation of the Pareto front. Our results indicate that the NSGA-II, despite being a highly competitive MOEA is not able to converge to the true Pareto front in most of the test problems adopted when performing only 3000 fitness function evaluations. If allowed a higher number of evaluations, the NSGA-II would certainly produce a very good (and well-distributed) approximation of the Pareto front.

8. CONCLUSIONS AND FUTURE WORK

We have presented a new local search technique inspired on Rough sets theory to improve the results of a MOEA. The proposed approach was found to provide very competitive results in a variety of test problems performing only 3000 fitness function evaluations. Within this number of evaluations, NSGA-II, a highly competitive MOEA, is not able to converge to the true Pareto front in most of the test problems adopted. This led us to conclude that Rough Sets is a suitable tool to be hybridized with a MOEA in order to improve the local exploration around the nondominated solutions found so far. It is important to mention

that the search engine needs to produce a coarse-grained approximation of the Pareto front with only a few number of function evaluations (as DE did in our case) to allow the Rough sets to explore those solutions efficiently and achieved a good approximation of the true Pareto front with a low computational cost. As part of our future work, we intend to improve the performance of the differential evolution algorithm adopted. For that sake, choosing the parents in a proper manner is of major importance in order to perform a good search. Thus, we are interested in exploring other DE schemes [13] in order to see if we can accelerate the convergence of our search engine. We are also interested in studying schemes that might reduce or automate (at least partially) the parameters adopted by our approach.

9. REFERENCES

- [1] H. A. Abbass. The Self-Adaptive Pareto Differential Evolution Algorithm. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 831–836, Piscataway, New Jersey, May 2002. IEEE Service Center.
- [2] C. A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2nd edition, 2007. ISBN: 978-0-387-33254-3.
- [3] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
- [5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.
- [6] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2005. ISBN 3-540-21398-8.
- [7] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. Coello Coello, and J. Molina. Pareto-adaptive ϵ -dominance. *Evolutionary Computation*, 15(4):493–517, Winter 2007.
- [8] S. Kukkonen and J. Lampinen. An Extension of Generalized Differential Evolution for Multi-objective Optimization with Constraints. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 752–761, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.
- [9] Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11(1):341–356, Summer 1982.
- [10] T. Robič and B. Filipič. DEMO: Differential Evolution for Multiobjective Optimization. In C. A. C. Coello, A. Hernández, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 520–533, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.
- [11] L. V. Santana-Quintero and C. A. C. Coello. An Algorithm Based on Differential Evolution for Multi-objective Problems. *International Journal of Computational Intelligence Research*, 1(2):151–169, 2005.
- [12] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-12, International Computer Science, Berkeley, California, March 1995.
- [13] R. Storn and K. Price. Differential evolution - a fast and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.
- [14] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.
- [15] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.
- [16] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, Summer 2003.

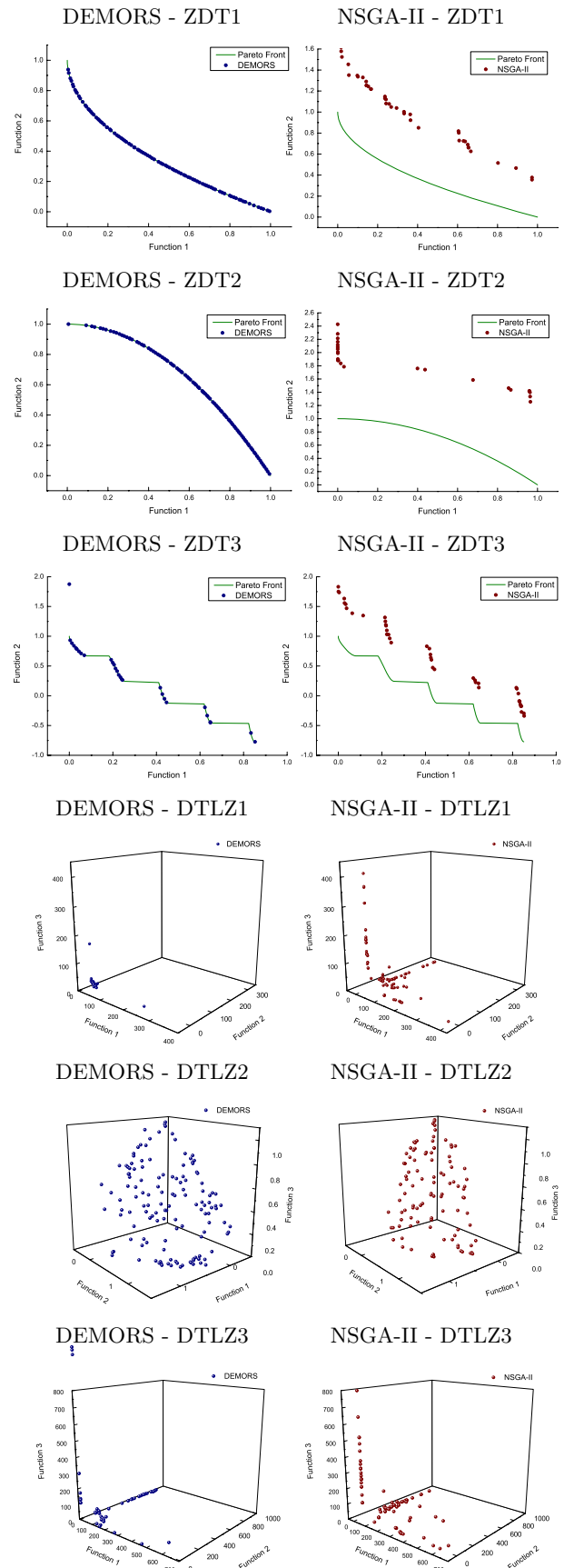


Figure 2: Pareto fronts generated by DEMORS (left) and NSGA-II (right) for ZDT1, ZDT2, ZDT3, DTLZ1, DTLZ2, and DTLZ3