Swarm Intelligence Guided by Multi-Objective Mathematical Programming Techniques

Saúl Zapotecas Martínez
CINVESTAV-IPN (EVOCINV)
Departamento de Computación
México, D.F. 07360, MEXICO
saul.zapotecas@gmail.com

Carlos A. Coello Coello
CINVESTAV-IPN (EVOCINV)
Departamento de Computación
México, D.F. 07360, MEXICO
ccoello@cs.cinvestav.mx

ABSTRACT

Since the early days of multi-objective particle swarm optimizers (MOPSOs), researchers have looked for appropriate mechanisms to define the set of leaders (or *global best* set) from the swarm. At the beginning, leaders were randomly selected from the set of nondominated solutions currently available. However, over the years, researchers realized that random selection schemes were not the best choice, and additional information was incorporated in the leader selection mechanism (namely, information related to density estimation). Here, we study the use of mathematical programming techniques for defining the leader selection mechanism of a MOPSO. The proposed approach decomposes a multi-objective optimization problem (MOP) into several single objective optimization problems by using traditional multi-objective mathematical programming techniques. Our preliminary results indicate that our proposed approach is a viable choice for solving MOPs, since it is able to outperform a state-of-the-art multi-objective evolutionary algorithm (MOEA).

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search.*

General Terms

Algorithms, Theory.

Keywords

Swarm Intelligence, multi-objective optimization.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'11, July 12–16, 2011, Dublin, Ireland.
Copyright 2011 ACM 978-1-4503-0690-4/11/07 ...$10.00.

1. INTRODUCTION

A continuous and unconstrained multi-objective optimization problem (MOP), can be stated as follows [1]:

$$\min_{x \in \Omega} \quad \{F(x)\} \tag{1}$$

where Ω defines the decision space and $F : \Omega \rightarrow \mathbb{R}^k$ is defined as the vector $F(x) = (f_1(x), \ldots, f_k(x))^T$, such that $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous and unconstrained function.

In multi-objective optimization, we aim to produce a set of trade-off solutions representing the best compromises among the objectives (i.e., solutions in which no objective can be improved without worsening another). Therefore, in order to describe the concept of optimality in which we are interested, the following definitions are introduced.

Definition 1. Let $x, y \in \Omega$, we say that x *dominates* y ($x \prec y$) if and only if, $f_i(x) \leq f_i(y)$ and $F(x) \neq F(y)$, for all $i = 1, \ldots, k$.

Definition 2. Let $x^\star \in \Omega$, we say that $x^\star$ is a *Pareto optimal* solution, if there is no other solution $y \in \Omega$ such that $y \prec x^\star$.

Definition 3. The *Pareto Optimal Set* is defined by $\mathcal{PS} = \{x \in \Omega | x \text{ is Pareto optimal solution}\}$ and its image (i.e., $\mathcal{PF} = \{F(x) | x \in \mathcal{PS}\}$) is called *Pareto Optimal Front.*

The main goal in multi-objective optimization is to maximize the number of elements of the Pareto optimal set and to maintain a well-distributed set of solutions along the Pareto front. In the specialized literature, there exist several multi-objective programming techniques to solve MOPs (see for example [6, 11]). However, several researchers have identified the main limitations of these mathematical programming approaches [1, 4, 10], which has motivated a lot of research on multi-objective evolutionary algorithms (MOEAs). The population-based nature of MOEAs and their flexible selection mechanisms have shown to be extremely useful and successful for dealing with MOPs, specially in the engineering field. In particular, particle swarm optimization (PSO) has been found to be a very successful bio-inspired metaheuristic for dealing with continuous and unconstrained optimization problems. This has motivated the interest of researchers in extending PSO to solve MOPs [14].

In this study, we employ a multi-objective programming technique for decomposing a MOP into several scalar functions. Then, a multi-objective particle swarm optimizer (MOPSO) follows the directions defined by each scalar function. The proposed MOPSO updates the position of each

[1]Without loss of generality, we assume minimization

particle using the best solution to each scalar problem. The proposed approach is compared with respect to a well-known decomposition-based MOEA (MOEA/D [15]) which is representative of the state of the art in the area.

2. THE PROPOSED APPROACH

2.1 Decomposing MOPs

In the specialized literature, there are several approaches for transforming a MOP into multiple single-objective optimization subproblems [6, 11]. These approaches use a weighted vector as their search direction. In this way, and under certain assumptions (e.g., the minimum is unique, the weighting coefficients are positive, etc.), a Pareto optimal point is achieved by solving such subproblems. Such decomposition techniques are perhaps the most common basis for currently available multi-objective mathematical programming approaches. From the many techniques available, the two most widely used are the *Tchebycheff* and the *Weighted Sum* approaches, in spite of the fact that approaches based on boundary intersection have certain advantages over them [2, 15]. This motivated the work presented here, in which we adopt a decomposition approach based on boundary intersection coupled to a MOPSO.

2.1.1 Penalty Boundary Intersection Approach

The Penalty Boundary Intersection (PBI) approach[2] introduced by Zhang and Li [15], minimizes both the distance (d_1) from $F(x)$ to the boundary intersection as well as the direction error (d_2) from $F(x)$ to the weighted vector, see Figure 1. The problem is mathematically stated as follows:

Let $w = (w_1, \ldots, w_k)^T$ be a weighted vector, i.e., $w_i \geq 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$. Then, the optimization problem is defined as:

$$\text{minimize:} \quad g(x|w, z^\star) = d_1 + \theta d_2 \tag{2}$$

such that:

$$d_1 = \frac{||(F(x) - z^\star)^T w||}{||w||}$$
$$\text{and} \quad d_2 = \left|\left|(F(x) - z^\star) - d_1 \frac{w}{||w||}\right|\right|$$

where $x \in \mathbb{R}^n$ and $z^\star = (z_1^\star, \ldots, z_k^\star)^T$ is the utopian vector, i.e., $z^\star = \min\{f_i(x)|x \in \Omega\}$ for each $i = 1, \ldots, k$.

In this way, the PBI approach can generate a good approximation along the Pareto optimal front by defining a well-distributed set of weighted vectors.

2.2 Guiding a MOPSO

The proposed MOPSO employs a decomposition-based framework similar to the one adopted by MOEA/D [15]. Therefore, a well-distributed set of weighted vectors W has to be previously defined. As it was mentioned before, PBI is adopted in our proposed approach. Note, however, that Tchebycheff or any other decomposition technique could be adopted as well.

At the beginning of the algorithm, a swarm of N particles $\mathcal{P} = \{x_1, \ldots, x_N\}$ is randomly initialized. Each particle possesses a flight velocity v_i and an age a_i, both of which are initially set to zero. Along the flight circuits, each particle tries to minimize one of the subproblems defined by the weighted vector w_i. Therefore, each particle undertakes its flight towards a better position in order to minimize a single subproblem $g(x_i|w_i, z^\star)$.

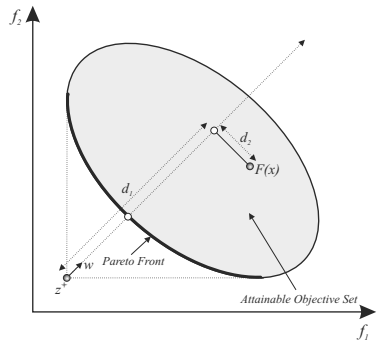

Figure 1: Illustration of the PBI approach.

The *personal best* $x_{pb,i}$ of the i^{th} particle, represents the best solution provided by the particle to the i^{th} subproblem. Since, at the beginning, a particle does not have a previous movement, the best personal position is initialized with the same position as the particle, i.e., $x_{pb,i} = x_i$.

At each cycle, the flight historical record of each particle is used to find the best solutions to each subproblem. Hence, *global best* set ($\mathcal{G}_{best}$) is defined in a natural way. This set contains the solutions that minimize each subproblem and it is updated in each cycle using $(\mu + \lambda)$-selection, where μ and λ represent the current and the next particles in the swarm, respectively. Therefore, the notion of elitism used in evolutionary multi-objective optimization is implicitly employed. However, in this case, a decomposition approach is used instead of the, more traditional, Pareto optimality.

Once the global best set has been defined, the velocity and the position of each particle are updated according to the traditional PSO flight equations:

$$\begin{array}{rcl} v_i^{t+1} & = & wv_i^t + c_1 r_1 (x_{pb,i} - x_t^t) + c_2 r_2 (x_{gb,i} - x_i^t) \\ x_i^{t+1} & = & x_i^t + v_i^{t+1} \end{array} \tag{3}$$

where $w \geq 0$ represents the inertia factor, $c_1, c_2 \geq 0$ are the constraints on the velocity, r_1, r_2 are two random variables having a uniform distribution in the range $(0, 1)$, and $v_i, x_{pb,i}$ and $x_{gb,i}$ represent the velocity, the personal best and the global best position for the i^{th} particle, respectively.

Since the proposed approach tries to minimize a set of subproblems (whose solutions at the end of the flight cycles should be very close to the Pareto optimal set), and the $(\mu + \lambda)$-selection introduces a high selection pressure that should contribute to this, we assume that all solutions in $\mathcal{G}_{best}$ are equally good (i.e., we assume that all the subproblems were satisfactorily solved). Thus, the velocity of each particle is computed using as their global best a solution which is randomly taken from $\mathcal{G}_{best}$.

The age of each particle is used to maintain diversity along the flight circuits. Basically, aging is used to indicate when a particle is not providing good information in its flight experience. When a particle does not improve its personal position in a flight cycle, then the particle increases (by one) its age. On the other hand, if a particle exceeds a certain age threshold, the particle (including, its velocity,

[2]based on the well-known Normal Boundary Intersection (NBI) method [2]

its age and its personal best) is reinitialized. The proposed approach employs a reinitialization mechanism based on a parametric probability density function, which involves the selected global best $x_{gb,i}$ and the personal best $x_{bp,i}$ of the current particle x_i^t. It aims to perform smart reinitialization movements from the personal best towards the global best solutions using a Gaussian distribution. Therefore, the j^{th} component of the new particle is reset according to the following equation:

$$x_i^{t+1}(j) = N\left(\frac{x_{gb,i}(j) - x_{pb,i}(j)}{2}, |x_{gb,i}(j) - x_{pb,i}(j)|\right) \quad (4)$$

where $N(m, \sigma)$ represents a random number using a normal distribution with mean m and sigma σ.

2.3 Our Proposed Approach

Since its very inception, (MOEA/D) [15] has motivated the development of other MOEAs based on decomposition (see for example [9, 12, 13, 16]). This new generation of MOEAs uses the same idea as MOEA/D (i.e., they define a neighborhood to select individuals for recombining and updating solutions from this same neighborhood). However, when it comes to MOPSOs, few of them are based on decomposition [12, 13]. Such decomposition-based MOPSOs use a neighborhood as their global best set. Moreover, they include archiving strategies (based on ϵ-dominance [8]) to store nondominated solutions along their flight cycles.

As indicated before, one of the main issues involved in the design of a good MOPSO is the definition of the global best set. In their early days, MOPSOs relied on simple random selection schemes, which were later refined through the addition of information from a density estimator. Additionally, the use of archiving strategies has also become common practice.

Unlike traditional MOPSOs, the approach presented here does not use any explicit diversity maintenance mechanism to obtain well-distributed solutions along the Pareto front. Instead, it relies on the penalty from the PBI approach, which provides diversity to such solutions along the flight circuits. It is worth noticing, however, that the use of other decomposition approaches could deteriorate the performance of our algorithm, especially when dealing with more complex problems. This is because the PBI approach forces particles to follow a single direction, since moving away from such direction will be penalized. Furthermore, our approach does not require neither the use of turbulence nor the Pareto optimality concept for approximating solutions towards the Pareto optimal set. Instead, we adopt both, a mechanism for selecting global best solutions based on the nature of the decomposition approach, and a mechanism to reinitialize the particles based on their age. Therefore, our proposed approach avoids the use of an external archive, which results in a lower computational time than any of the archive-based MOPSOs currently available.

3. EXPERIMENTAL RESULTS

In order to assess the performance of our proposed approach, we compared its results with respect to those obtained by MOEA/D [15] in the following test problems:

Standard Test Problems

We adopted six test problems whose Pareto fronts have different characteristics including convexity, concavity, disconnections and multi-frontality. For two-objective problems, we adopted four problems taken from the Zitzler-Deb-Thile (ZDT) test suite [17] (ZDT2, ZDT3, ZDT4 and ZDT6). For three-objective problems, we adopted three problems taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [5] (DTLZ6 and DTLZ7). For ZDT2 and ZDT3, 30 decision variables were adopted. ZDT4 and ZDT6 were tested using 10 decision variables. Finally, DTLZ6 and DTLZ7 were adopted with 12 and 22 decision variables, respectively.

Experimental Setup

For each MOP, 30 independent runs were performed with each algorithm. The parameters for MOEA/D were set as in [15], using a population size $N = 100$ for the bi-objective problems and $N = 300$ for the three-objective problems. The maximum number of generations was set to 150. For our proposed MOPSO, the constraints on the velocity (c_1, c_2) and the inertia factor (w) were dynamically defined. As in [12], we used values uniformly distributed, such that: $c_1, c_2 \in (1.2, 2.0)$ and $w \in (0.1, 0.5)$. The parameter θ used in the PBI approach was set to 5 for both our MOPSO and MOEA/D.

Performance Measures

For each MOP, the algorithms were evaluated using the *Hypervolume* performance measure [18]. The results are summarized in Table 1. For the hypervolume metric, the reference vectors adopted were: $r = (1.1, \ldots, 1.1)^T$ for DTLZ6 and the ZDT test problems, while for DTLZ7 the reference vector $r = (1.0, 1.0, 6.1)^T$ was used. Table 1 displays both the *average* and the standard deviation (σ) of the hypervolume performance measure, for each of the test problems adopted. The best results are shown in **boldface** for each test problem adopted.

Preliminary Results

As it is well-known, the hypervolume performance measure is Pareto compliant [19] and quantifies the approximation of nondominated solutions to the Pareto optimal front. From this table, it can be seen that our proposed approach outperformed MOEA/D in all the adopted MOPs. This indicates that our proposed algorithm produced a better approximation and distribution to the Pareto optimal front. However, note also that in problem DTLZ6, for example, there is hardly any difference in the performance between both algorithms. Finally, Figure 2 shows a comparison of results for our MOPSO and MOEA/D in the ZDT4 problem.

Table 1: Results of the $\mathcal{H}v$ performance measure for our proposed MOPSO and MOEA/D

MOP	MOPSO	MOEA/D
	average (σ)	*average* (σ)
ZDT2	**0.536473** (0.000425)	0.316948 (0.091569)
ZDT3	**1.317248** (0.002921)	1.246748 (0.044609)
ZDT4	**0.862245** (0.029966)	0.774609 (0.065285)
ZDT6	**0.504519** (0.000004)	0.457862 (0.009460)
DTLZ6	**0.426532** (0.000038)	0.426153 (0.000098)
DTLZ7	**1.409133** (0.007166)	1.375630 (0.141559)

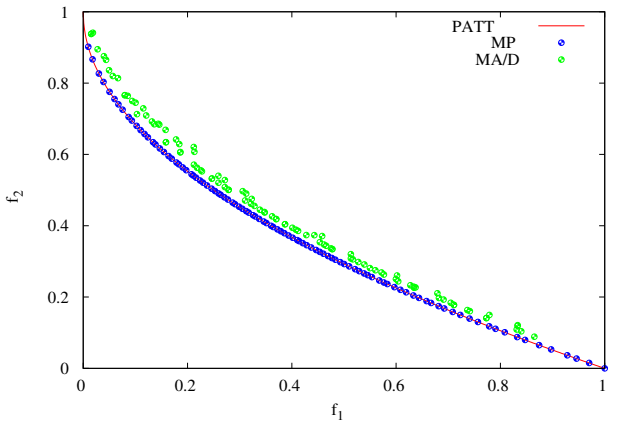

Figure 2: Comparison of results for our MOPSO and MOEA/D in ZDT4.

4. CONCLUSIONS AND FUTURE WORK

In this study, we have presented a multi-objective particle swarm optimizer based on decomposition. Preliminary results showed that our proposed MOPSO outperformed MOEA/D in the test problems adopted, with respect to the hypervolume performance measure. This proposed approach does not use an external archive to store nondominated solutions and adopts instead, a mechanism to select the globally best solutions. The penalty defined by the parameter θ (in the PBI approach) enforces to follow a search direction defined by each weighted vector. However, the study of the variation of this parameter along the flight circuits is a promising path for future research.

As part of our future work, we are interested in providing a local search mechanism using non-gradient mathematical programming techniques to improve solutions in different directions. In this way, a new global best set could be defined using those improved solutions. Additionally, the exploitation of the decomposition approach when coupling it to other metaheuristics (e.g., scatter search [7] or artificial immune systems [3]) is also a promising research area. In general, we are interested in providing to MOEAs, search mechanisms based on mathematical methods.

Acknowledgments

The first author acknowledges support from CONACyT to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 103570.

5. REFERENCES

[1] C. A. Coello Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, Agosto 1999.

[2] I. Das and J. E. Dennis. Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[3] L. N. de Castro and J. Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer, 2002.

[4] K. Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In K. Miettinen, M. M. Mäkelä, P. Neittaanmäki, and J. Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, chapter 8, pages 135–161. John Wiley & Sons, Ltd, Chichester, Reino Unido, 1999.

[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, 2nd edition edition, June 2005.

[7] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39:653–684, 2000.

[8] M. Laumanns, L. Thiele, E. Zitzler, and K. Deb. Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization. In W. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. Potter, A. Schultz, J. Miller, E. Burke, and N. Jonoska, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pages 439–447, San Francisco, California, July 2002. Morgan Kaufmann Publishers.

[9] H. Li and Q. Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, April 2009.

[10] Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Berlin, 2000.

[11] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Dordrecht, 1999.

[12] N. A. Moubayed, A. Petrovski, and J. A. W. McCall. A novel smart multi-objective particle swarm optimisation using decomposition. In *PPSN (2)*, pages 1–10, 2010.

[13] W. Peng and Q. Zhang. A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *IEEE International Conference on Granular Computing, 2008. GrC 2008*, pages 534 –537, 2008.

[14] M. Reyes-Sierra and C. A. Coello Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

[15] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[16] Q. Zhang, W. Liu, and H. Li. The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 203–208, Trondheim, Norway, May 2009. IEEE Press.

[17] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[18] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.

[19] 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, April 2003.