Resource Allocation by Diversity

Yuri Lavinas

***Abstract*—Multi-objective Evolutionary Algorithm based on Decomposition decomposes a multi-objective problem into simpler, single-objective subproblems by means of scalarizations. In it, all subproblems are uniformly treated, i.e, there is no priority among them. Each subproblem is related to a area of the theoretical Pareto Front, and it is expected that different areas would be more difficult to find approximations that other areas. Therefore, this may lead to an unbalanced exploration of the search space. In this work the goal is to analyze a prioritizing function based on an on-line diversity metric. hence, we perform an experimental study of the proposed algorithm, with NSGA-2, MOEA/D-DE and MOEA/D-GRA on both benchmark group functions, the WFG and the recently proposed Bi-BBOB. The results indicate that XXX.**

I. Introduction

Multi-objective Optimization Problems (MOPs) [1] are problems composed by two or more functions which must be simultaneously optimized by the same set of parameters. This composition is characterized by a set of conflicting objective functions resulting in a set of optimal compromise solutions.

These m multiple objective functions must be optimized simultaneously:

$$\text{minimize} f(x) = (f_1(x), ..., f_m(x)), \text{ with } x \in \mathbb{R}^D, \quad (1)$$

where m is the number of objective functions and $\mathbb{R}^m$ is the objective function space. $x \in \mathbb{R}^D = \{x_1, x_2, ..., x_D\}$ is a D-dimensional vector and it represents a candidate solution with D variables, $f : \mathbb{R}^D \to \mathbb{R}^m$ is a vector of objective functions.

Objectives often conflict with each other therefore no point in Ω minimizes all the objectives at the same time. Consequently, the goal of MOP solvers is to find the best trade-off that balances the different objectives in an optimal way.

Given two feasible solutions vectors u, v in Ω, u Pareto-dominates v, denoted by $f(u) \prec f(v)$, if and only if $f_k(u) \leq f_k(v), \forall_k \in \{1, ..., m\}$ and $f(u) \neq f(v)$.

A solution $x \in \mathbb{R}^D$ is considered Pareto-Optimal if there exists no other solution $y \in \mathbb{R}^D$ such that $f(y) \succ f(x)$, i.e., if x is non-dominated in the feasible decision space. A point is called non-dominated if no other point dominates it. That is, no single solution provides a better trade-off in all objectives.

The set of all Pareto-optimal solutions is known as the Pareto-Optimal Set (PS), while the image of this set is referred to as the Pareto-optimal Front (PF).

$$PS = \{x \in \Omega | \nexists y \in \Omega : f(y) \succ f(x^*)\}, \quad (2)$$

$$PF = \{f(x) | x \in PS\}. \quad (3)$$

Multi-objective evolutionary algorithms (MOEAs) are characterized by their ability to find good approximations the PF of a MOP in a single run [2].Over the last two decades, many different search techniques were proposed for improving the effectiveness of multi-objective algorithms. Among them, three are the major paradigms: Pareto domination-based approaches [3], [4], the indicator-based approaches [5], [6], and the decomposition-based approaches [7], [8].

Multi-objective Evolutionary Algorithm based on Decomposition, MOEA/D [8], represents a class of population-based meta-heuristics for solving Multi Objective Problems [9]. In this search paradigm, the original multi-objective problem is decomposed into simpler, single-objective subproblems by means of scalarizations, see Figure 1. However, it may waste functions evaluations searching in directions that do not present Pareto-optimal solutions [10]. As in the example of Figure 2, some vector represent ”empty regions” of the PF.

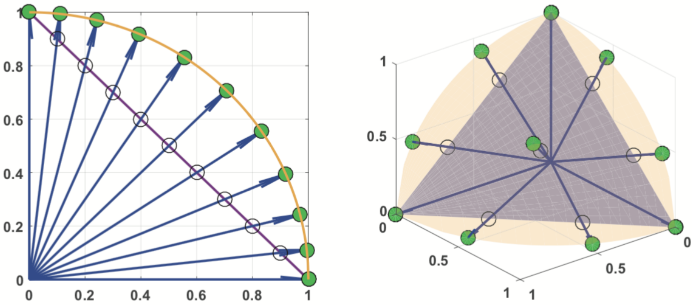


Fig. 1. A decomposition strategy generates weight vectors that defines the subproblems. Figure from [11].

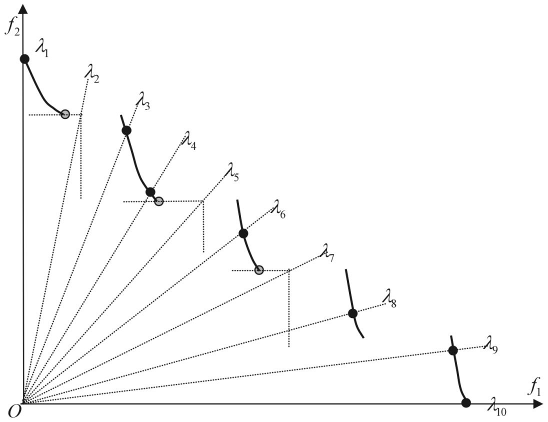
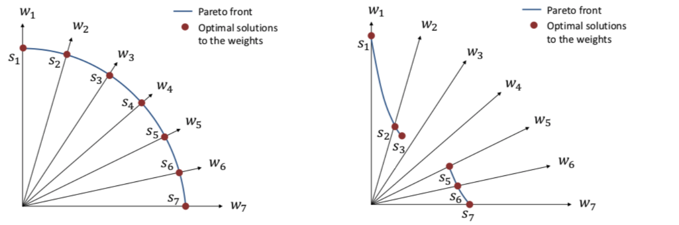


Fig. 2. Distribution of optimal solutions of subproblems with uniform weight vectors on ZDT3. Figure from [12].

In the original MOEA/D [8], all the subproblems have the same amount of computational resource (number of iterac-



(a) On a concave Pareto front (b) On a disconnected, convex Pareto front

Fig. 3. An example that uniformly distributed weights may lead to different distributions of optimal solutions. (a) Solutions s_1 to s_7 are the optimal solutions of weights w_1 to w_7, respectively. (b) Solutions s_1,s_2,s_3,s_6 and s_7 are the optimal solutions of w_1,w_2,w_3,w_6 and w_7, respectively, while solution s_5 is the optimal solution of w_4 and w_5.Figure from [13]

tions). It has been observed, as one can expect that some parts of the theoretical Pareto Front PF, in an MOP can be more difficult to approximate than others, Figure 3, [14] [15] [16].

This difficulties on approximating the PF may lead to a unbalanced exploration of the search space. One of the reasons is that, given to this dissimilar conditions, the number of iteractions needed will possibly not be the same.

To reduce this computational cost, the allocation of resources may be better distributed among different subproblems according to their difficulties. One way to address this problem is to adjust the behavior of the algorithm in an on-line manner to suit the problem in question [17], [18], [19], [14], [20], [21] [22]. All algorithmic components can be tunned adaptively and often feedback information is needed for these adaptation strategies.

Another way, is to allocate different amounts of resources to the subproblems based on some priority function. In a few recent works recently proposed, a utility function is being used to distribute the resources given priorities to subproblems that contribute more to the algorithm's search. In the works of Zhang et al. [14] and Zhou et al. [15] the a utility function was proposed aiming to prioritize solutions based on a historical convergence information during different generations. Another approach was implemented in [16], where the utility function was based on the presence of a solution from the main population on a secondary population.

This work is divided in sections and subsections according to this and that.

II. Related Works

A. Utility functions

Utility functions are one way of defining priorities [23],[24], and may be used as one way of deciding computational resources distributions among subproblems by guiding the distribution over generations [22].

To the best of the authors knowledge most of the works related to utility functions and resource allocation are: MOEA/D-DRA [14], MOEA/D-GRA [15], MOEA/D-CRA [16], MOEA/D-AMS [25]. In these work an utility function, u, is used but without a clear justification.

Zhou and Zhang [15] claim that MOEA/D-GRA may be seen as an extension of MOEA/D-DRA and MOEA/D-AMS. The reason is that in MOEA/D-DRA and in MOEA/D-AMS some unsolved subproblems are chosen for evolution in each generation according to their utility functions. For both MOEA/D-DRA and MOEA/D-GRA, the utility function, $u = \{u_1, u_2, ..., u_N\}$ for every subproblem $i = 1, ..., N$, is as defined in the next equation.

$$u_i = \frac{\text{old fun val} - \text{new fun val}}{\text{old fun val}}, \tag{4}$$

supported by the idea that if a subproblem has been improved over the last ΔT generations (*old function value*), it should have a high probability to be improved over the next few generations. While MOEA/D-CRA is based on a MOEA/D variation with two populations. Their utility function is based on if of an individual from population A is selected to compose the population B.

My proposal is to integrate utility function with a diversity metric based on a geometrical interpretation of convergence and diversity to MOEA/D-GRA, given its straightforward implementation and integration with other MOEA/D variants.

B. Diversity Metric

Here, I use MRDL as a way to assess the diversity of solutions in an on-line manner and use a utility function based on the output of this metric to guide resource allocation at each generation.

The hyper-volume indicator (HV) or the Inverted Generational Distance (IGD) could be used as the diversity metric. However both of them include information about convergence and diversity in a single metric.

There are mainly two groups metrics, the off-line ones, that calculate the diversity after the execution of the algorithm, and the on-line, that calculate the diversity during the execution of the algorithm.

a) Off-line metrics: Chi-square-like deviation [26], Spacing method [27], Uniformly distribution index [28], Entropy approach [29], Grid diversity metric [30], sparsity measure [31], and many others.

They need knowledge of the PF or the ideal vector.

b) On-line metrics: sigma method [32] (PF lies in the positive objective space), entropy of the solutions by using Parzen window density estimation-[33] (sensitive to kernel width), maximum relative diversity loss [34] (expensive $O(N^2)$, with N being the size of the parent population).

In this work the maximum relative diversity loss was chosen to an online diversity assessment to measure the diversity loss caused by any individual in the population, for future works other metrics could be introduced.

III. Proposed Method

MOEA/D with on-line Resource Allocation by Diversity metric, MOEA/D-RAD, is a variant of MOEA/D that used a diversity metric as a way to select values for the utility function. Besides having a utility function based on a diversity metric, the algorithm is the same as MOEA/D-DE [7].

The algorithm 1 describe MOEA/D-RAD. Except from line 4 and 7, the whole procedure in algorithm 1 is as in MOEA/D-DE. Likewise, all reproduction procedures are the same as in

Algorithm 1 MOEA/D-RAD

```
Initialize the weight vectors $\lambda_i$, the neighborhood $B_i$, the
  utility value $u_i$ every subproblem $i = 1, ..., N$.
while Termination criteria do
    for 1 to N do
        if rand() $< u_i$ then            ▷ From MOEA/D-GRA
            Generate a offspring $y$ for subproblem $i$.
            Update the population by $y$.
    Update $\boldsymbol{u}$ using a diversity metric. ▷ Introduced here.
```

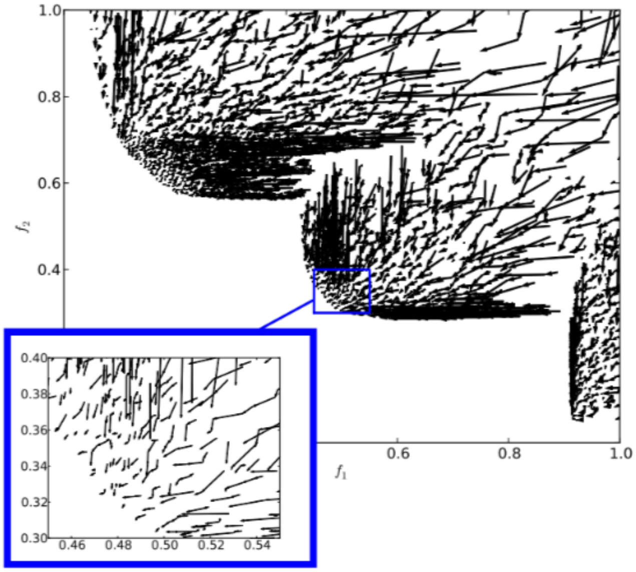


Fig. 4. Estimated convergence directions over 500 generations in CEC-09 UF1 benchmark test problem.Figure from [34]

MOEA/D-DE [7]. We initialize the value of the vector $u = 0.5$ as in [15]. A sensitivity analyzes need to be performed for deciding suitable values for u.

A. Maximum Relative Diversity Loss (MRDL)

Maximum relative diversity loss (MRDL) is used to estimate the diversity loss of a solution to the whole population [34].

Some properties and implications of the MRDL:

- If a new offspring generated is identical to any offspring solution in the convergence archive, $\Gamma^{p\to c}$ will be infinite.
- High values of $\Gamma^{p\to c}$ indicates the existence of similar offspring solution in the convergence archive or the offspring solution is close to the line of estimated convergence direction.

The idea of the MRDL is to use the space movement (convergence directions) of a solution on the objective space towards the PF, as can be seen in Figure 4, to calculate the diversity of the population. The further a objective vector of a solution is from the convergence direction, more it contributes for the diversity of the approximated PF.

The main idea of calculating Maximum Relative Diversity Loss MRDL, $\Gamma^{p\to c}$, is to use it as a utility function as follows.

$$u_i = \Gamma_i^{p\to c} \text{ , with } i = 1, ..., N. \tag{5}$$

$$u_i = (u_i - min(u_i))/(max(u_i) - min(u_i)) \tag{6}$$

a) MRDL: To calculate the MRDL, we need to compute k convergence directions at every generation and then estimate the Relative Diversity Loss (RDL) for each of these k convergence directions. Therefore, to estimate the diversity loss of a solution to the whole population by:

For every incumbent solution related to a subproblem i, from the whole population.

$$\Gamma_i^{p\to c} = \max_{i=1,\ldots,k} \Gamma_{d.conv_y}^{p\to c} \tag{7}$$

b) RDL.: RDL is a diversity measurement quantity that indicates the amount of diversity loss of an individual solution between two consecutive generations. High values of RDL imply a reduction of the solution spread and this equation may indicated the amount of diversity loss.

This quantity is given by a division between the shortest distance of a parent and offspring to the line of convergence direction.

$$\Gamma_{d.conv_y}^{p\to c} = \frac{||p\prime - proj_{d.conv_y} p\prime||}{||c\prime - proj_{d.conv_y} c\prime||} \tag{8}$$

The numerator in 8 is the closest distance between the parent solution (p) to the convergence direction ($c_r - p_r$). While, the denominator in 8 is the closest distance between the offspring solution (c) to the convergence direction ($c_r - p_r$).

with $p\prime$ and $c\prime$ given by:

$$\begin{aligned} p\prime &= p - p_r \\ c\prime &= c - c_s \end{aligned} \tag{9}$$

with p_r and c_s being the parent and offspring objective vectors used to calculate the convergence direction in equation 12. That is the index s is equal to the index j used to calculate $conv_y$. The same principle is valid for the index h.

The vector projection between two vectors a and b that I used is defined as follows.

$$proj_{d.conv_y} p\prime = \frac{d.conv_y \cdot p\prime}{|p\prime|^2} p\prime \tag{10}$$

While the norm of $p\prime - proj_{d.conv_y} p\prime$ is calculated as follows.

$$||p\prime - proj_{d.conv_y} p\prime|| = sqrt(crossprod(proj_{d.conv_y} p\prime)) \tag{11}$$

The norm of $c\prime - proj_{d.conv_y} c\prime$ is calculated similarly.

c) Estimating convergence direction given weak-dominance relationship.: To estimate the convergence direction, $d.conv_y$, we need to have a offspring, c_j, that dominates a parent. Select a parent, p_h, solution that is closest to this offspring in the objective space.

For every weakly dominated parent, one convergence direction is calculated as in the next equation.

$$d.conv_y = c_j - p_h \tag{12}$$

The index j (for indexing offsprings, c_j) is selected from the set D_c.

$$D_c = \{d | \exists c_d \prec p_k, k \in 1, ..., N, d \in [1, ..., |C|]\} \tag{13}$$

N is the parent population size, $|C|$ is the size of the offspring population C. In equation 13, the offspring c_d must weakly dominate at least one parent solution.

As from Zitzler et al. [35] study, weak dominance ($A \succeq B$) means that any solution in B is weakly dominated by a solution in A. However, this does not rule out equality, because $A \succeq A$ for all approximation sets $A \in \Omega$.

The index h (for indexing parents, p_h) comes from the following two equations.

$$h = \underset{k \in D_p}{argmin} ||p_k - c_j|| \tag{14}$$

$$D_p = \{k | \exists c_j \prec p_k, k \in 1, ..., N\} \tag{15}$$

D_p in equation 15 denotes the index set of parent solutions which are weakly dominated by c_j (the j index comes from equation 13).

References

[1] K. Miettinen, "Nonlinear multiobjective optimization, volume 12 of international series in operations research and management science," 1999.

[2] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[5] N. Beume, B. Naujoks, and M. Emmerich, "Sms-emoa: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[6] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 832–842.

[7] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE Transactions on evolutionary computation*, vol. 13, no. 2, pp. 284–302, 2009.

[8] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[9] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.

[10] L. C. Bezerra, M. López-Ibáñez, and T. Stützle, "Comparing decomposition-based and automatically component-wise designed multi-objective evolutionary algorithms," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 396–410.

[11] T. Chugh, "Handling expensive multiobjective optimization problems with evolutionary algorithms," *Jyväskylä studies in computing 263.*, 2017.

[12] H. Li, M. Ding, J. Deng, and Q. Zhang, "On the use of random weights in moea/d," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 978–985.

[13] M. Li and X. Yao, "What weights work for you? adapting weights for any pareto front shape in decomposition-based evolutionary multi-objective optimisation," *arXiv preprint arXiv:1709.02679*, 2017.

[14] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of moea/d on cec09 unconstrained mop test instances," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 2009, pp. 203–208.

[15] A. Zhou and Q. Zhang, "Are all the subproblems equally important? resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 52–64, 2016.

[16] Q. Kang, X. Song, M. Zhou, and L. Li, "A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[17] R. Hinterding, Z. Michalewicz, and A. E. Eiben, "Adaptation in evolutionary computation: A survey," in *Evolutionary Computation, 1997., IEEE International Conference on*. IEEE, 1997, pp. 65–69.

[18] K. De Jong, "Parameter setting in eas: a 30 year perspective," in *Parameter setting in evolutionary algorithms*. Springer, 2007, pp. 1–18.

[19] S. Meyer-Nieberg and H.-G. Beyer, "Self-adaptation in evolutionary algorithms," in *Parameter setting in evolutionary algorithms*. Springer, 2007, pp. 47–75.

[20] O. Kramer, "Evolutionary self-adaptation: a survey of operators and strategy parameters," *Evolutionary Intelligence*, vol. 3, no. 2, pp. 51–65, 2010.

[21] J. Zhang, W.-N. Chen, Z.-H. Zhan, W.-J. Yu, Y.-L. Li, N. Chen, and Q. Zhou, "A survey on algorithm adaptation in evolutionary computation," *Frontiers of Electrical and Electronic Engineering*, vol. 7, no. 1, pp. 16–31, 2012.

[22] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multi-objective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 508–523, 2015.

[23] V. Chankong and Y. Haimes, "Multiobjective decision making: Theory and methodology north holland," *New York*, 1983.

[24] S. O. Hansson, "Decision theory a brief introduction. department of philosophy and the history of technology, royal institute of technology (kth) stockholm," 2005.

[25] T.-C. Chiang and Y.-P. Lai, "Moea/d-ams: Improving moea/d by an adaptive mating selection mechanism," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1473–1480.

[26] K. Deb, "Genetic algorithms in multimodal function optimization," Ph.D. dissertation, Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics, University of Alabama, 1989.

[27] J. Scott, "Fault tolerant design using single and multi-criteria genetic algorithms," *Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology*, 1995.

[28] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons," *Artificial intelligence review*, vol. 17, no. 4, pp. 251–290, 2002.

[29] A. Farhang-Mehr and S. Azarm, "Diversity assessment of pareto optimal solution sets: an entropy approach," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 1. IEEE, 2002, pp. 723–728.

[30] K. Deb and S. Jain, "Running performance metrics for evolutionary multi-objective optimization," 2002.

[31] K. Deb, M. Mohan, and S. Mishra, "A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions," *KanGAL report*, vol. 2003002, pp. 1–18, 2003.

[32] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (mopso)," in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE, 2003, pp. 26–33.

[33] K. C. Tan, C. K. Goh, A. Mamun, and E. Ei, "An evolutionary artificial immune system for multi-objective optimization," *European Journal of Operational Research*, vol. 187, no. 2, pp. 371–392, 2008.

[34] S. B. Gee, K. C. Tan, V. A. Shim, and N. R. Pal, "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 542–559, 2015.

[35] 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*, vol. 7, no. 2, pp. 117–132, 2003.