Population Dynamics Indicators for Evolutionary Many-Objective Optimization

Raunak Sengupta[1], Monalisa Pal*[3], Sriparna Saha[2], and Sanghamitra Bandyopadhyay[3]

[1] Department of Electrical Engineering and
[2] Department of Computer Science and Technology,
Indian Institute of Technology, Patna
[3] Machine Intelligence Unit, Indian Statistical Institute, Kolkata
{raunaksengupta,monalisap90,sriparna.saha,sanghami}@gmail.com

Abstract. Recent research on multi- and many-objective optimization has led to the development of various state-of-the-art algorithms which produce satisfactory results for various kinds of problems. However, in real life, the underlying objective functions or the characteristic landscape formed by the objectives may not be known beforehand. This makes it difficult for a user to choose the correct optimization algorithm. This paper proposes new indicators which attempt to summarize the population dynamics across iterations. The statistics of the population movement can help in identifying various features of the problem at hand and the capacity of an algorithm to deal with the challenges corresponding to the features. The analysis of population movement can enable further modifications of an existing algorithm according to the optimization problem. The indicators can also help in the development of adaptive optimization algorithms by providing feedback during the search for optimality.

Keywords: Many-Objective Optimization, Population movement, Pareto-optimality, Visualization

1 Introduction

In the recent years, several many-objective optimization (MaOO) algorithms have been developed that make use of different kinds of strategies. Algorithms based on reference points such as θ-DEA [9], MOEA/D [10], NSGA-III [3] etc. have been shown to perform well for several benchmark test problems like DTLZ [2] and WFG [6] functions. NSGA-III uses the concept of non-dominated sorting for selecting the members of the new population [3] whereas θ-DEA introduces θ-dominance based on the argument that non-dominated sorting does not produce enough selection pressure [9]. MOEA/D, which has mating constraints, performs very well for problems like DTLZ2, but its performance is not very good for problems such as DTLZ4, which has a biased density of solutions [10]. Another

* Corresponding author.

major class of MaOO algorithms is indicator-based algorithms. Algorithms such as HypE [1], IBEA [11], MOMBI-II [5] come under this class as they make use of indicator values to perform selection. The problems that need to be optimized can provide different kinds of challenges. The possible features of a problem at hand have been classified into five different types in literature [6]:

1. *Geometry*: Shape of Pareto-front can be convex, concave, linear, mixed, degenerate
2. *Parameter Dependencies*: Separable objectives, non-separable objectives (Capability to determine ideal points by considering only one objective at a time)
3. *Bias*: Presence of a bias while mapping solutions from decision space to fitness functions in objective space
4. *Many-to-One mappings*: Pareto one-to-one, Pareto many-to-one, flat regions, isolated optima
5. *Modality*: Uni-modal, multi-modal (presence of local optimal fronts)

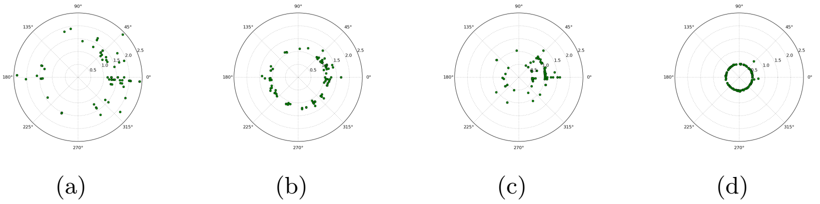
All of the algorithms have their own advantages and disadvantages and give best results for problems with only certain kinds of features. In real life, the problems that need to be optimized often have either a too complicated mathematical expression or is in the form of a black box (simulation, physical device) and thus much is not known about the features. This makes it difficult to choose the correct optimization problem suited for the function. Vigorous work has also not been done on relating the various problem features with the strategies best suited to overcome the challenges corresponding to the problem features.

In this paper, we propose indicators that can be observed throughout the iterations of an algorithm and based on them, the user can understand the features of the problem at hand. Much work has not been done on visualizing the dynamics of the points during search for optima. The indicators attempt to summarize the overall movement of the points in a population for a particular problem and algorithm. This will help in identifying the shortcomings of an algorithm, enhancing an algorithm and even designing better algorithms.

2 Indicators and Visualization of Population Movement

Information about the population movement can give a great deal of insight about both the problem function and the nature of the algorithm being used. Several works in literature use the information obtained from the previous iterations to optimize a problem more efficiently. Neighborhood based cross generation mutation (NCG) [8] produces new solutions by performing vector differencing between points in the same neighborhood, appearing in consecutive generations. This strategy has been justified as an attempt to direct a point towards the optimal front. On the other hand, Fitness Improvement Rate (FIR) [7] measures the improvement in solutions and uses this information as a heuristic to select a reproduction operator for the next generation.

However much work has not been done on explicitly quantifying and visualizing the movement of a population. To achieve this, we need to design quantitative indicators that define different properties of a distribution at a particular



(a) (b) (c) (d)

Fig. 1: Radial plots of NSGA-III run on DTLZ1 with 3 objectives (a) iteration 59 (b) iteration 87 (c) iteration 113 (d) iteration 170

iteration as well as the changes that occur through the iterations. The proposed indicators are inspired from the recently developed radial plot visualization technique [4]. The movement of the population on a radial plot is shown in Fig. 1 for various iterations when NSGA-III is used for optimizing the 3-objective DTLZ1 problem. As seen in Fig. 1(a), the population members gather around a point at a distance corresponding to radius 1 on the polar plot indicating that the algorithm is trying to overcome the local optimal front. The points then start to converge at this local optimum as shown in Fig. 1(b). At this stage, the consecutive generations keep improving the diversity until some new points overcome the local optimal front (Fig. 1(c)). At this point, there is a relative deterioration in the diversity until the points finally converge to the global optimum at a radius of 0.5 (Fig. 1(d)).

Table 1: Mean Hypervolumes on DTLZ4 and WFG6 (3 objectives)

Problem	MOEA/D-PBI	NSGA-III	θ-DEA	HypE
DTLZ4	0.406020	0.744634	0.729265	0.549999
WFG6	0.654956	0.685939	0.690060	0.708633

Table 1 lists the Hypervolume values for four algorithms viz.-MOEA/D-PBI, θ-DEA, HypE and NSGA-III, which are obtained from [9]. The final Pareto-front obtained by MOEA/D-PBI after optimizing on DTLZ4 problem is shown in Fig. 2(a). The plot clearly shows the poor distribution of solutions which results in the relatively poor Hypervolume values as compared to the other algorithms (Table 1). The radial plots in Fig. 2(b) and Fig. 2(c), shows that poor diversity is obtained by NSGA-III and θ-DEA, respectively, for WFG6 problem. However, these plots contradict with the corresponding Hypervolume values in Table 1 which are relatively good and comparable with the other state-of-the-art algorithms. This implies that Hypervolume indicator values may not always be enough to assess the efficacy of an algorithm.

To have an overall understanding of the movement, it is necessary to have several indicators each indicating different properties and statistics of the move-

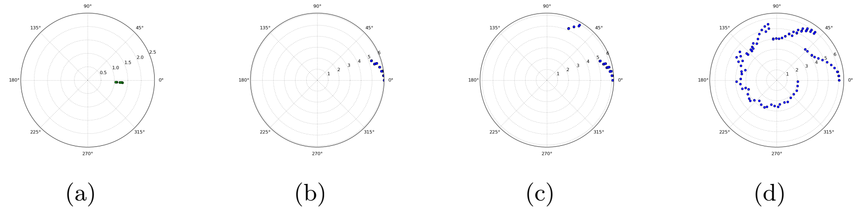


Fig. 2: Radial plots for 3 objective functions (a) MOEA/D on DTLZ4 (b) NSGA-III on WFG6 (c) θ-DEA on WFG6 (d) A good distribution for WFG6

ment. The various properties of the movement that have been deemed necessary to have an overall summary of the response of an algorithm to a problem, are as follows :

1. The tendency of a population distribution to diversify and spread over iterations needs to be captured. This indicator should not be affected by the tendency of the population members to improve their fitnesses.
2. An indicator is required for capturing the overall tendency of the population distribution to shrink in when plotted on radial coordinates, considering a minimization problem. This indicator should be as independent of the diversification nature of the population as possible.
3. An indicator capturing the overall rate of change in the fitnesses of the population would be able to tell about the speed of convergence.
4. An indicator for the consistency among the members could tell whether the tendency to improve is same for population members from different regions.

Based on the above discussions, this paper proposes several indicators to summarize the movement and performance of an algorithm. The indicators defined in this paper are inspired from the radial coordinate plots. At any generation (or iteration), g, each of the members of the population are associated with one of the reference lines closest to it, as done in [4]. Therefore, we have a vector called $Spread$, which is of the same length as that of the number of reference lines. Each member of this array represents the number of population members associated with it. Each member of the population also has an associated distance value, r, which is its Euclidean distance from the origin. Since a reference line might have several associated members, it also might have several corresponding values of r. We define vectors r_inner and r_outer. Each member of these vectors represent the smallest and largest value of r associated to each reference line respectively. In case a reference line has only one value of r associated to it, the corresponding r_inner and r_outer values become equal. If a reference line does not have any associated member, its corresponding value of r is considered undefined and not used for calculations.

The indicators can be classified into two types - Independent and Dependent.

1. *Independent Indicators*: The values of the indicators under this class are independent of factors such as the range of values each objective can take,

the scales of different objectives and other factors which are dependent on the objective functions. Such indicators have a fixed range of values. Indicators under this class are:

(a) *D_metric*: This indicator is a measure of the diversity of the population at a certain iteration, g. To define this indicator, we first define the vector, *Spread* as in Eq. (1), whose i-th element is the number of points coupled with the i-th reference line.

$$Spread^g = (s_1^g, s_2^g, \cdots, s_n^g)^T \tag{1}$$

We also define another vector called $Ideal\_Spread$ such that $s_i^g = \frac{pop\_size}{n}$, where n is the number of reference lines being used, *pop_size* is the cardinality of the population and $i = 1, 2, \cdots, n$. We have chosen all the members of $Ideal\_Spread$ to be equal. However, one is free to define a distribution according to requirements.

Now, *D_metric* (as defined by Eq. (2)) at generation g, is the euclidean distance of the current *Spread* vector from the $Ideal\_Spread$ vector. A value close to zero implies good diversity while a higher value implies poor diversity.

$$D\_metric^g = \frac{n}{pop\_size}\sqrt{\sum_{i=1}^{n}(Spread_i^g - Ideal\_Spread_i)^2} \tag{2}$$

(b) *V_metric*: This indicator is a measure of the tendency of the population in general to move towards the origin of the radial plot i.e. improve the member's fitnesses considering a minimization problem. This indicator is defined in Eq. (3).

$$V\_metric^g = \frac{\sum_{i=1}^{n}(1 - a_i^g)}{n^g} \tag{3}$$

where,

$$a_i^g = \begin{cases} 1 & , (r\_inner_i^{g-1} - r\_inner_i^g) > \epsilon \\ 0 & , -\epsilon < (r\_inner_i^{g-1} - r\_inner_i^g) < \epsilon \\ -1 & , \text{otherwise} \end{cases}$$

such that n^g is the number of points satisfying the conditions: $Spread_i^g \neq 0$ and $Spread_i^{g-1} \neq 0$. ϵ in the equation is just a small constant . Intuitively, *V_metric* is the fraction of reference lines with an associated point that cease to show improvement i.e. move towards origin .

(c) *Consistency*: This indicator (given by Eq. 4) attempts to capture the consistency in the improvement of members associated to a reference line. A poor value of consistency in the initial iterations would imply the presence of certain regions where improvement is difficult for the algorithm.

$$Consistency^g = \sum_{i=1}^{n} \frac{a_i^g}{l_i^g} \tag{4}$$

where, l_i^g is the total number of generations up to g with continued association of at-least one point, for the i-th reference line. If the number of points associated to the i-th reference line becomes zero for any iteration, the corresponding value of l_i^g resets to zero and again starts counting when an associated point appears.

(d) *Velocity*: Velocity tries to measure the amount of change occurring per generation as shown in Eq. (5).

$$Velocity^g = \frac{\sum_{i=1}^{n} \tan^{-1}(r\_inner_i^{g-1} - r\_inner_i^g)}{n^g} \tag{5}$$

It captures the amount of change at each generation by calculating the average slope of the plot of the inner radii associated with each of the reference lines through iterations.

2. *Dependent Indicators*: The values of indicators under this class are dependent on the range of values that the objective function can take. These indicators are relatively simple and naïve. However these are necessary and provide a lot of information. Indicators under this class are:

(a) *Innermost Radius*: As the name suggests, the value of this indicator is given by the minimum value of the radius, r, among the population members at a particular generation. This is also equivalent to $\min_{i=1}^{n} r\_inner_i^g$.

(b) *Outermost Radius*: The value of this indicator is given by the maximum value of the radius, r, among the population members at generation g. This is also equivalent to $\max_{i=1}^{n} r\_outer_i^g$.

(c) *Average Inner Radius*: Average of all the inner radii associated with the reference lines at generation g. This is equivalent to $(\sum_{i=1}^{n} r\_inner_i^g)/n$.

(d) *Average Outer Radius*: Average of all the outer radii associated with the reference lines at generation g. This is equivalent to $(\sum_{i=1}^{n} r\_outer_i)/n$.

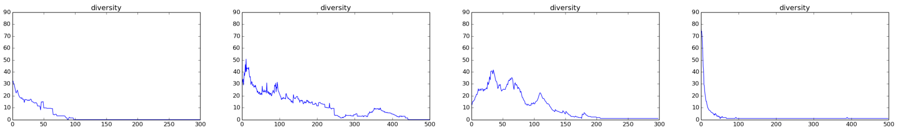
(e) *Inner Band*: Difference between the maximum inner radius and the minimum inner radius at generation g. This measure is an attempt to capture the variance in the inner radii of the members. This is also equivalent to $(\max_{i=1}^{n} r\_inner_i - \min_{i=1}^{n} r\_inner_i)$.

3 Results and Discussions

This section presents a few of the results that were obtained by employing the indicators. All of the experiments have been run using Python 2.7.6 on a system with Intel Core i7 processor @ 2.5 GHz and GTX 860 M GPU.

We have done our analysis on DTLZ1 to DTLZ4 problems with 3 and 8 objectives. The algorithms that have been used for various comparisons are NSGA-II, NSGA-III, MOEA/D and θ-DEA. Parameters of the algorithms have been set according to the recommendations in [9].

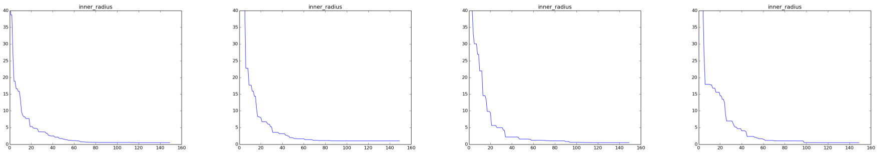
Due to constraint of space, only the plots which give a deeper insight have been shown and discussed over. Following are some of the inferences that have been enabled by observing the indicators :



(a) (b) (c) (d)

Fig. 3: Plots of *D_metric* (a) MOEA/D on DTLZ2 (8-obj) (b) MOEA/D on DTLZ3 (8-obj) (c) NSGA-III on DTLZ1 (3-obj) (d) NSGA-III on DTLZ4 (3-obj)

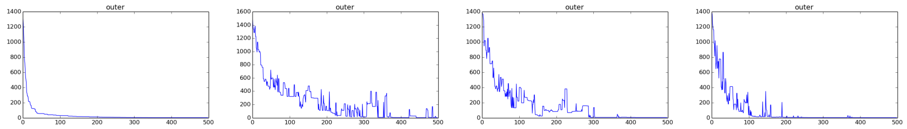
1. As observed from plots in Fig. 3, problems with local optimal fronts like DTLZ1 and DTLZ3 usually tend to have a relatively more bumpy plot of diversity. Further, the plot of the innermost radius for such problems (Fig. 4) often tend to have flat regions. These regions correspond to the local optimal fronts where the points get stuck for some time. At these points, the diversity usually starts to improve until the points eventually overcome the optimal front.



(a) (b) (c) (d)

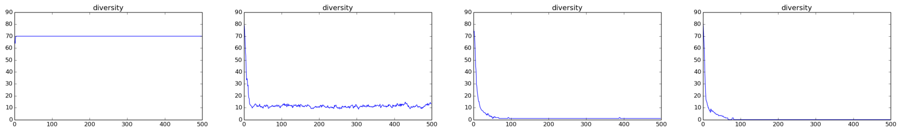
Fig. 4: Plots of innermost radius on 3-objective problems (a) MOEA/D on DTLZ1 (b) NSGA-II on DTLZ1 (c) NSGA-III on DTLZ1 (d) θ-DEA on DTLZ1

2. It has been pointed out in literature that NSGA-III has a lower convergence speed than algorithms using PBI (penalty based boundary intersection) functions due to its lower selection pressure [9]. It can be observed from the plots in Fig. 4 that MOEA/D reaches global optimal front roughly by 70 iterations while NSGA-III takes more than 90 iterations. NSGA-II on the other hand gets stuck at a local optimal front corresponding to radius equal to 1. However, NSGA-III and θ-DEA are comparable.
3. It can be observed from Fig. 5 that NSGA-II and NSGA-III in general have a higher tendency to have outliers, MOEA/D has the least tendency while θ-DEA is in between. This implies that performing non-dominated sorting whether based on Pareto dominance or θ-dominance has a tendency to generate outliers. This behavior can be explained through the argument that non-dominated sorting enables relatively higher exploration and simultaneously reduces the selection pressure. Using mating constraints and PBI,



(a) (b) (c) (d)

Fig. 5: Plots of outermost radius on 3-objective problems (a) MOEA/D on DTLZ1 (b) NSGA-II on DTLZ1 (c) NSGA-III on DTLZ1 (d) θ-DEA on DTLZ1

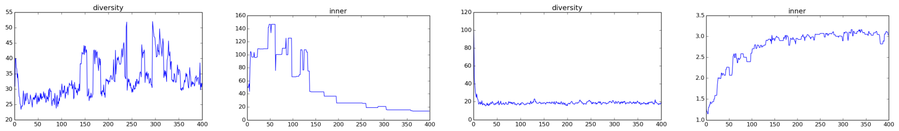


(a) (b) (c) (d)

Fig. 6: Plots of *D_metric* on DTLZ4 (3-objective) functions (a)MOEA/D-PBI (b) NSGA-II (c) NSGA-III (d) θ-DEA

such as done in MOEA/D, leads to an increased selection pressure and more exploitation which ultimately results into faster convergence rates and less outliers.

4. MOEA/D-PBI however suffers from a very poor diversity when run on problems with a biased density of solutions such as DTLZ4 as can be seen from the plots in Fig. 6. This happens because of the presence of mating constraints which decreases the ability of an algorithm to explore, which is required very much when there are regions with biased density of solutions.
5. From the diversity plots (Fig. 6), we also observe the improved capability of reference point based techniques in obtaining a good diversity of solutions when compared to methods such as crowding distance.

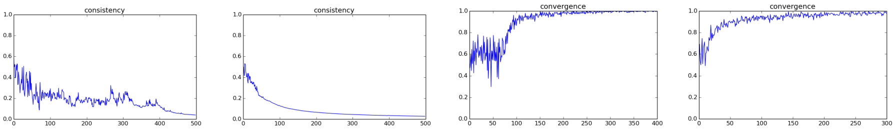


(a) (b) (c) (d)

Fig. 7: NSGA-II plots for 8-objective problems (a) D-metric for DTLZ1 (b) Innermost Radius for DTLZ1 (c) D-metric for DTLZ4 (d) Innermost Radius for DTLZ4

6. The behavior of the reference point based algorithms do not change much with an increase in the number of objectives. Only the required number of

function evaluation increases. However, NSGA-II shows a significant difference in its behavior as can be observed from the plots for the 8-objective problems (Fig 7). The value of the innermost radius does not decrease steadily like it does for the other algorithms. In fact it increases and converges to a high value for all the test problems except DTLZ1. This shows that crowding distance renders useless when the number of objectives are as high as 8. This combined with the low selection pressure of non-dominated sorting makes NSGA-II not very suitable for problems with number of objectives as high as 8.



(a) (b) (c) (d)

Fig. 8: Plots of consistency (a) NSGA-III on DTLZ3 (3-obj) (b) NSGA-III on DTLZ4 (3-obj); and V-metric (c) θ-DEA on DTLZ1 (8-obj) (d) θ-DEA on DTLZ2 (8-obj)

7. It can be observed in general that the consistency plots and the *V-metric* plots (Fig. 8) associated to problems such as DTLZ1 and DTLZ3 which have local optimal fronts in general have more spikes and large variances.

4 Conclusion

This paper provides a direction towards explicitly quantifying the population movement of an evolutionary multi-objective optimization and making various inferences based on their analysis. This work is based on the argument that a single indicator fails to provide sufficient information about the performance of an algorithm. The indicators observed through iterations will be able to tell us about the response of an algorithm while optimizing a particular kind of a problem and in turn tell us about its advantages, disadvantages and how well it is able to overcome various challenges posed by an algorithm. The indicators will also enable identification of various features that might be present in unknown objective functions and act as a heuristic in choosing the right strategies for optimization.

Future work includes developing indicators that take into account the shape and structure that emerges from the iterations. The variance among the innermost solutions at a particular iteration might potentially be informative. However, using the variance at its simplest form will not work if the shape is complicated. One might also consider deriving other indicators from the ones mentioned in this paper by simply performing various signal processing operations on them. A major direction that can be pursued from here is using the information obtained from the indicators as a feedback for the development of an adaptive and robust many-objective optimization algorithm and correlating various features of objective functions to the best suited strategies. Future work also includes performing a much more detailed survey incorporating various classes of algorithms and a large variety of challenging test problems.

Acknowledgment

This work is funded by the project (DST-INRIA/2015-02/BIDEE/0978) of the Indo-French Centre for the Promotion of Advanced Research (IFCPAR).

References

1. Bader, J., Zitzler, E.: Hype: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation 19(1), 45–76 (2011), http://dx.doi.org/10.1162/EVCO_a_00009
2. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. Evolutionary Multiobjective Optimization pp. 105–145 (2005)
3. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. Evolutionary Computation, IEEE Transactions on 18(4), 577–601 (2014)
4. He, Z., Yen, G.G.: Visualization and performance metric in many-objective optimization. IEEE Trans. Evolutionary Computation 20(3), 386–402 (2016), http://dx.doi.org/10.1109/TEVC.2015.2472283
5. Hernández Gómez, R., Coello Coello, C.A.: Improved metaheuristic based on the r2 indicator for many-objective optimization. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. pp. 679–686. GECCO '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2739480.2754776
6. Huband, S., Hingston, P., Barone, L., While, R.L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans. Evolutionary Computation 10(5), 477–506 (2006), http://dx.doi.org/10.1109/TEVC.2005.861417
7. Li, K., Fialho, ., Kwong, S., Zhang, Q.: Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 18(1), 114–130 (Feb 2014)
8. Qiu, X., Xu, J.X., Tan, K.C., Abbass, H.A.: Adaptive cross-generation differential evolution operators for multiobjective optimization. IEEE Transactions on Evolutionary Computation 20(2), 232–244 (April 2016)
9. Yuan, Y., Xu, H., Wang, B., Yao, X.: A new dominance relation-based evolutionary algorithm for many-objective optimization. IEEE Trans. Evolutionary Computation 20(1), 16–37 (2016), http://dx.doi.org/10.1109/TEVC.2015.2420112
10. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evolutionary Computation 11(6), 712–731 (2007), http://dx.doi.org/10.1109/TEVC.2007.892759
11. Zitzler, E., Künzli, S.: Indicator-Based Selection in Multiobjective Search, pp. 832–842. Springer Berlin Heidelberg, Berlin, Heidelberg (2004), http://dx.doi.org/10.1007/978-3-540-30217-9_84