Mutual Information for Performance Assessment of Multi Objective Optimisers: Preliminary Results

Noura Al Moubayed, Andrei Petrovski, and John McCall

School of Computing
The Robert Gordon University
Aberdeen, UK
n.al-moubayed@rgu.ac.uk

Abstract. Solving multi-objective problems usually results in a set of Perto-optimal solutions, or a Pareto front. Assessing the quality of these solutions, however, and comparing the performance of different multi-objective optimisers is still not very well understood. Current trends either model the outcome of the optimiser as a probability density function in the objective space, or defines an indicator that quantify the overall performance of the optimiser. Here an approach based on the concept of mutual information is proposed. The approach models the probability density function of the optimisers' output and use that to define an indicator, namely the amount of shared information among the compared Pareto fronts. The strength of the new approach is not only in better assessment of performance but also the interpretability of the results it provides.

1 Introduction

Many real-world applications often involve optimisation of multiple, competing objectives in large search spaces [1]. It is therefore an important task to effectively and simultaneously address multiple optimisation objectives by identifying a set of well-distributed Pareto optimal solutions that yield good values for each objective. Population-based metaheuristics (e.g. Genetic Algorithms) have been developed to facilitate an efficient search in multi-dimensional solution spaces, the feasible regions within which are determined by a set of (often non-linear) constraints. However, instead of obtaining infinite number of Pareto optimal solutions, which is a time consuming and resource demanding task, it is often preferable to search for a set of representative solutions that closely approximate the true Pareto front being uniformly distributed along its length [2,3].

As with single objective optimisation, two factors are important when assessing a multiobjective optimiser: the quality of the found solutions, and the time spent to find them. However the stochastic nature of evolutionary algorithms results in the relation between time and quality not fixed, but rather represented by a probability distribution function. In addition having a set of

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 537–544, 2013.

solutions (Pareto front) instead of a single outcome of the multiobjective optimisation process makes quantifying the quality of these solutions much harder. This is added to having multiple runs and the necessity to statistically quantify the behaviour of the optimiser over these runs increases the difficult of quality assessment [5].

This paper introduces a novel approach for performance assessment of mutliobjective optimizers. First a simple mutual-information based method is presented and discussed. A more robust approach is then proposed which looks at PFs as images. To validate and test this method, five problems are tested using three popular multi-objective optimizers and the results are tested using three popular indicators.

2 Background and Related Work

2.1 Multi-objective Optimisation Problems

Solving a multi-objective optimisation problem is challenging because an improvement in one objective often happens at the expense of deterioration in other objective(s). The optimisation challenge therefore is to find the entire set of trade-off solutions that satisfy all conflicting objectives.

Let $F(X) \in \Delta \subset R^m$ be a vector of objectives: $F(X) = \{f_1(X), \ldots, f_m(X)\}$, where $X = \{x_1, x_2, \ldots, x_n\} \in \Omega \subset R^n$ is the vector of decision variables, n is the dimension of solution space, and $m \geq 2$ is the number of objectives. The search space (also called the solution space) refers to the space of decision variables, whereas the objective space is the space where the objective vectors lie. When minimizing $F(X)$, for example, a domination relationship is defined between the solutions as follows: let $X, Y \in \Omega$, $X \prec Y$ if and only if $f_i(X) \leq f_i(Y)$ for all $i = \{1, 2, ..., m\}$, and there is at least one j for which $f_j(X) < f_j(Y)$. X^* is a Pareto optimal solution if there is no other solution $S \in \Omega$ such that $S \prec X^*$. Therefore the Pareto optimality of a solution guarantees that any enhancement of one objective would results in the worsening of at least one other objective. The Pareto optimal set is the set of all non-dominated solutions. The image of the Pareto optimal set in the objective space (i.e. $F(X^*)$) is called the Pareto Front (PF).

2.2 Measuring Quality of Multi-objective Optimizers

Multiobjective evolutionary algorithms are stochastic in nature, due to the random element in the algorithms, i.e. running the algorithm twice would most likely produce a different set of results. For this reason the optimizer should be run several times and the probability density function is then empirically estimated. Comparing two optimizers would then mean comparing their probability density functions which then implicate the issue of statistical hypothesis testing [6].

In the literature, there are two main approaches to assess the quality of produced PFs. The most common one is the indicator approach where a PF is mapped, using a defined function, to a real number then a standard statistical hypothesis test is applied on the indicator values. The second approach is usually referred to as the attainment function method in which for each objective vector there is a probability p that the produced approximation set contains an objective vector that dominates z. The attainment function then gives a probability estimate of z to be attained in one optimization run with a statistical test procedure to count for all the runs [7].

A comprehensive survey on quality indicators can be found in [5]. However, here we list three recommended indicators in [5] for their theoretical advantage. The indicators are based on different aspects of the data and therefore using them all will provide more ,hopefully complementary, information than using just one.

The inverted generational distance The inverted generational distance, I_{IGD}, [8] measures the uniformity of distribution of the obtained solutions in terms of dispersion and extension. The average distance is calculated for each point of the actual PF (PF_{True}), denoted as A, and the nearest point of the approximated PF (PF_{approx}), denoted as B: $I_{IGD(A,B)} = \frac{(\sum_{a \in A} (\min_{b \in B} \|F(a) - F(b)\|^2))^{1/2}}{|A|}$ The indicator usually gives a fair evaluation of the produced PF, however it can be prune to outliers due to the use of the distance measure.

Hypervolume The hypervolume indicator, I_{hv}, [9] measures the volume of the objective space that is dominated by a PF approximation (A). I_{hv} uses a reference point v^* which denotes an upper bound over all objectives. v^* is defined as the worst objective values found in A (i.e. v^* is dominated by all solutions in A). Using the Lebesgue measure (Λ), I_{hv} is defined as: $I_{hv}(A) = \Lambda\Big(\bigcup_{a \in A} \{x \,|a \prec x \prec v^* |\}\Big)$. The advantage of hypervolume is that it is conceptually intuitive, but it can be computationally costly and it requires a reference point which may affect the ordering of pairs in incomparable sets.

ϵ Indicator The ϵ indicator, I_ϵ, [10] measures the minimum distance which a PF approximation (A) has to be translated in the objective space to dominate the actual PF B. The ϵ-Indicator is defined as:$I_\epsilon(A,B) = \min_{\epsilon \in \mathbb{R}} \{\forall b \in B, \exists b'_i - \epsilon \leq b_i, \forall 1 \leq i \leq n\}$. This is a fast indicator to compute and has an intuitive meaning: how much do I need to translate/scale the set A so that it covers the reference set?. However, the choice of reference set can dramatically affect the results.

Using quality indicators is an attractive approach of quality assessment due to its simplicity. It has , however, some shortcomings: 1) each indicator looks at the performance from only one perspective, e.g. spread, diversity, or dominance, which may skew the conclusions drawn. 2) In the case of incomparable PFs an indicator will actually give an inaccurate result. 3) For indicators that use distance functions, outliers can cause a real problem in disturbing the calculation of the indicator. 4) Quality indicators do not take the statistics of the data in the objective space into account.

The approach we take in this paper models the output of the optimizer directly as an empirical probability density function and then calculates the mutual information between the approximated PF and the theoretical one.

3 Methods

3.1 Mutual Information

Intuitively speaking, mutual information measures how much information are shared between two random variables X, Y. In other words how much knowledge of one variable reduces the uncertainly about the other. Formally, mutual information is defined as follows: $I(X,Y) = \sum_Y \sum_X p(x,y) log \frac{p(x,y)}{p(x)p(y)}$, where $p(x,y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively with summation replaced by an integral in the case of continuous random variables. If the two variables are dependent then $I(X,Y)$ measures the shared information between the two variables and it would be positive. If on the other hand, they are independent then $I(X,Y) = 0$.

To complete the calculation of MI, the joint and marginal probabilities should be calculated. To estimate the joint probability distribution/density function a two dimensional histogram is used [11].

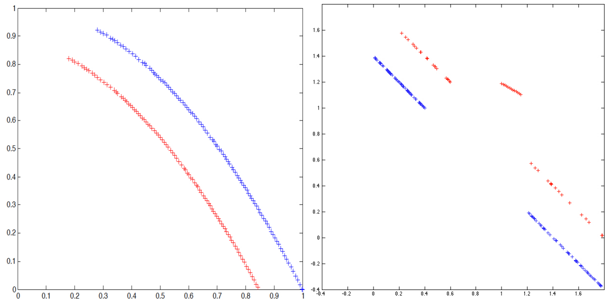
The mutual information as defined so far is not normalized, i.e. it can take any positive value. Here we use a normalized version as defined in [12]:

$$U(X,Y) = 2\frac{I(X,Y)}{H(X) + H(Y)} \tag{1}$$

where $H(X)$, $H(Y)$ are the marginal entropies of X and Y respectively.

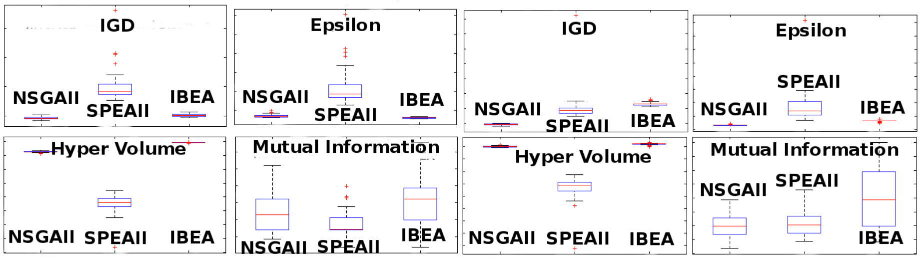
The way mutual information is applied would differ depending on the independence assumptions among the objectives. The definition in Sec3.1 is only valid for univariate random variables, so if we work on the assumption that the objectives are all independent then mutual information can be measured separately between the approximated PF and the true PF, one objective at a time and then the mutual information indicator is defined as: $MI(A,B) = \sum_{i=1}^{n} \alpha_i U(A_i, B_i)$, where A is the approximated PF, B is the true PF, A_i is the values of objective i from the PF A, n is the number of objectives, and α is the weight vector where $\alpha_i \leq 0$ and $\sum_{i=1}^{n} \alpha = 1$. The result of this indicator quantifies how much the approximated PF reduces uncertainty about the true PF. The higher MI then the better approximation A is to the true PF.

This is a simple and reliable measure that can be applied to any MOP regardless of the shape of the PF. It does not require a reference point and does not depend on one aspect of the data like spread or diversity. Another major drawback of this indicator is that the histogram estimation would allow two unequal PFs to produce the same MI if they have similar histograms, as demonstrated in Figure 1a. To circumvent this problem a novel method for mutual-information based indicator is proposed in the next section.

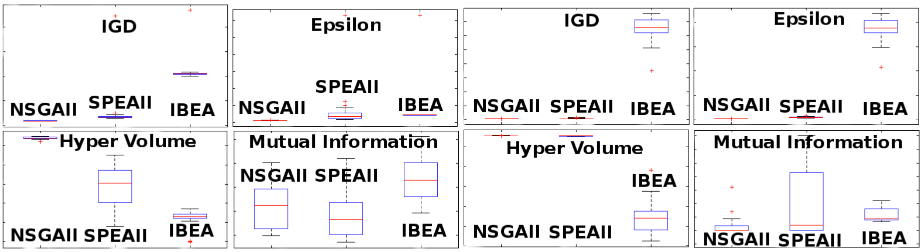


(a) (b)

Fig. 1. (a) An example of two unequal PFs with the same histogram. The blue PF (A) is the true PF for ZDT1 and the red PF (B) is a shifted copy of the blue PF. MI(A,A) = MI(A,B) = 1. (b) The blue dots belong to a hypothetical true PF. The red dots belong to a hypothetical approximated PF. The dots within the circle belong to the approximated PF and are considered outliers.



(a) ZDT1 (b) ZDT2



(c) ZDT3 (d) ZDT4

Fig. 2. Results for ZDT1-4 using the three algorithm and the four quality indicators.

3.2 Processing Pareto Fronts as Images

The main idea here is to create a pixel-based image for each PF, and then calculate mutual information among these newly created images. In order for these images to be comparable they should all have the same resolution per dimension, have the same origin and the same image size.

Given iA: the image of the approximated PF A and iB: the image of the true PF B then a new indicator is defined as: $I_{MI(A,B)} = U(iA, iB)$.

The highest resolution possible for the images is the minimum difference between two adjacent points of all the PF of interest which might be a very high resolution that the generated images would be extremely large to be practical for processing, so the resolution can be reduced in order to generate images of reasonable size. There must be a careful balance here as low resolution images could lead to a large lose of information affecting the quality of the measure itself. Following are the steps to create the PF images. 1) rescale all PFs so they all have the same origin 2) calculate the ratio between the required resolution and the maximum objective value among all PFs which is necessary to project from objective space to the image pixel space. 3) each point in PF is transated to a pixel of value 1 in the resulted image and the rest are zeros.

Going back to the problem of unequal PFs with equal histograms. If we used the new measure as an indicator in Fig.1a then $I_{MI}(A, A) = 1$ and $I_{MI}(A, B) = 0.0018$ which clearly shows that the problem is resolved.

3.3 Handling Outliers

One of the main features of any quality indicator is its ability to handle outliers in the approximated PF. Outliers can cause a bias in the quality measure calculation especially if a distance measure is used. Some measures try to reduce this effect, for example I_{IGD} is calculated by starting from the true PF and then trying to find the closest points in the approximated PF. Although this minimizes the effect of outliers, it still does not provide a fair comparison as it will give the same value for an approximated PF with or without the outliers.

Figure 1b demonstrates this effect . In this example we created a hypothetical true PF and approximated PF that contains some outliers. To check the robustness of the quality indicators we calculate the value of the indicator using the approximated PF with and without the outliers. The results show that I_{MI} and I_{hv} are robust to the outliers and are able to distinguish them (I_{hv}=0.2284 and 0.2332 without and with the indicators respectively and I_{MI} =0.0008 and 0.0007) while the other two indicators could not (I_{IGD}=0.0145 for both cases and I_{ϵ}=0.3911 for both cases as well).

4 Experiments

It is usually tricky to evaluate the evaluation metric. Here we tackle this issue indirectly by comparing the results of three other quality measures and discuss how each, and I_{MI}, handle different cases of evaluation. To test the newly developed measure 5 standard two-dimensional MOPs are used: ZDT1-ZDT4, and ZDT6 [13]. The selected test problems cover diverse MOPs with convex, concave, connected and disconnected PFs. These problems were frequently used to verify the performance of several algorithms in the field of multi-objective optimisation.

The test MOPs are solved using three popular multi-objective optimisation methods: NSGAII [14], SPEAII [15], IBEA [16].

For each MOP the three methods were run 30 times using a population of 300 individual and lasted for 250 generations. The quality of the approximated PFs from the five MOPs is measured using four indicators: I_{IGD}, I_{ϵ}, I_{hv}, and I_{MI}. For I_{MI} calculation a resolution is set to $1000X1000$.

5 Results and Discussion

To demonstrate the results of the approximated PF using the three methods and tested by the four indicators, for each MOP four plots are generated each for each indicator. Each of these plots contains a box plot representation of the values of one indicator applied on the 30 runs for each of the three methods, Figures 2a - 2d with results of ZDT6 not presented due to limited space.

Looking carefully at the figures, the different indicators draw a rather vague, and somehow confusing, picture about the performance of the different methods. Although most of the differences among the different methods are statistically significant they do not always go in the same direction for different indicators. This is a known issue in the quality assessment of mutliobjective optimizers via indicators [5]. This is interpreted as different indicators provide different information regarding the approximated PF so one can chose the optimizer based on what is more relevant to the application.

The I_{MI} indicator seems to be giving a slightly different view from the rest of the indicators. For instance, it is the only indicator to show no significant difference in some cases which actually reflects more what we see from visual inspection of the approximated PFs. It also usually shows less variance among the different runs and fewer outliers which can be credited to its probabilistic nature.

By definition mutual information is only applied on univariate random variables and hence an independence assumption is imposed among the objectives. If the objectives are dependent, which would be the case in most problems, then the previous definition of mutual information is not accurate. However, by transforming the PF to an image all the dependency information are preserved and hence the indicator I_{MI} can be seen as a multivariant approach.

Because the mutual information function uses estimated probability density functions/ distributions, it is less affected by outliers and can produce a much more robust results than the indicators that use distance functions.

References

1. Talbi, E.: Metaheuristics - From Design to Implementation. Wiley (2009)
2. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Genetic and Evolutionary Computation. Springer, New York (2007)
3. Deb, K.: Multi-objective optimization. In: Multi-Objective Optimization using Evolutionary Algorithms, pp. 13–46 (2001)
4. Al Moubayed, N., Hasan, B.A.S., Gan, J.Q., Petrovski, A., McCall, J.: Continuous presentation for multi-objective channel selection in brain-computer interfaces. In: 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1–7. IEEE (2012)
5. Fonseca, C.M., Knowles, J.D., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. In: Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), vol. 216 (2005)
6. Fonseca, C.M., da Fonseca, V.G., Paquete, L.: Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In: Coello, C.A.C., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 250–264. Springer, Heidelberg (2005)
7. Grunert da Fonseca, V., Fonseca, C.M., Hall, A.O.: Inferential performance assessment of stochastic optimisers and the attainment function. In: Zitzler, E., Deb, K., Thiele, L., Coello, C.A.C., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 213–225. Springer, Heidelberg (2001)
8. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective evolutionary algorithm research: A history and analysis. Air Force Inst. Technol., Dayton, OH. Tech. Rep. TR-98-03 (1998)
9. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
10. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)
11. Scott, D.W.: On optimal and data-based histograms. Biometrika 66(3), 605–610 (1979)
12. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2005)
13. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary Multiobjective Optimization, pp. 105–145 (2005)
14. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: Nsga-ii. IEEE Trans. on Evolutionary Computation 6(2), 181–197 (2002)
15. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm (2001)
16. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)