G-INDICATOR: AN M-ARY QUALITY INDICATOR FOR THE EVALUATION OF NON-DOMINATED SETS

Giovanni Lizárraga Lizárraga, Arturo Hernández Aguirre and Salvador Botello Rionda

Comunicación Técnica No I-07-18/14-12-2007
(CC/CIMAT)

G–Indicator: an M–ary Quality Indicator for the Evaluation of Non–dominated Sets.

Giovanni Lizárraga Lizárraga, Arturo Hernández Aguirre, Salvador Botello Rionda
Center for Research in Mathematics (CIMAT)
Department of Computer Science
A.P. 402, Guanajuato, Gto. 36240, México
giovanni@cimat.mx, artha@cimat.mx, botello@cimat.mx

December 13, 2007

Abstract

An open problem in multi–objective optimization using the Pareto optimality criteria, is how to evaluate the performance of different evolutionary algorithms that solve multi–objective problems. As the output of these algorithms is a non–dominated set (NS), this problem can be reduced to evaluate what NS is better than the others based on their projection on the objective space. In this work we propose a new performance measure for the evaluation of NSs, that does not need any information a priori of the multiobjective problem. Neither it needs any parameter tuning. Besides, its evaluations of the NSs agree with intuition. Also, we introduce a benchmark of test cases to evaluate performance measures, that considers several topologies of the Pareto Front.

1 Introduction

Many algorithms based on the Pareto's Optimality Criteria (POC) have been developed [5] [6] [7] in order to solve multi–objective optimization problems. Instead of generating a single solution, these algorithms generate a set X of vector solutions x that approximate the Pareto set (PS). A PS is a *non–dominated set* (defined next), while the True Pareto Set is a PS that can not be improved by any means. A Pareto front is the projection of the PS over the space of decision functions $F(X) = \{f_1(x), f_2(x), \ldots, f_M(x)\}$. The Pareto front, another set itself, is usually described as a surface in the objective space that represents the best tradeoff possible between the objective functions. The True Pareto Front (P^*) is the projection of the True Pareto Set over the space of decision functions. Hereafter in the article we locate points, sets, vectors, and solutions in the space of the objective functions.

The property that defines a *non–dominated set*, (NS) is that none of its elements is dominated by any other element in the set. A vector a dominates a vector b if it is true that $(\forall i \in [1, \ldots, d],\ a_i \leq b_i) \ \wedge \ (\exists j | a_j < b_j)$, where d is the dimension of the vectors. This definition is for minimization problems, in maximization problems the "$\leq$" and "$<$" are substituted by "$\geq$" and "$>$" respectively. Without loss of generality, we consider we are working with minimization problems.

One of the most important difficulties about using the POC is how to compare the performance of different algorithms. Usually, this is done by comparing the NSs that the algorithms generate, so it is necessary to have a criteria (a performance measure or comparison method) to evaluate a NS. In order to create such a criteria we need to decide what we want from a NS. We focus these three characteristics in order to decide how good is a NS:

Convergence: it refers to how close is a NS to P*. It comes directly from the definition of the Pareto's optimality.

Dispersion: we are interested in the different tradeoff for the objective functions in a multiobjective problem. As the output of a multiobjective algorithm is a finite set of solutions, it is desirable to maximize the information these solutions provide. For this reason we expect a good NS to be as uniformly distributed as possible, in order to avoid zones of the Pareto Front with too many solution and zones with too few.

Extension: It refers to the range of values of the objective functions. It is desirable to have information in all ranges of the objective functions. This characteristic is closely related with dispersion. Actually, dispersion and extension can be treated as a single characteristic. From now we refer to dispersion–extension as *DE*

Convergence is related with the information in the direction of improvement of all objective functions, we mean, the direction in which all objective functions decrease their value. DE is related to information orthogonal to the direction just mentioned. In other words, DE is related with the information in all those directions in which if we improve one or more objective functions, we degrade other(s). For these reasons we consider convergence and DE as orthogonal characteristics.

Convergence is considered the most important property, because it is related with "how optimal" are the points in the NS. So, it is not a surprise that a lot of the research in performance measures for NSs is focused on convergence. A classical work is that of Hansen and Jaszkiewicz [8], where they define the three following relationships between two NSs.

Weak outperformance: the NS A weakly outperforms the NS B ($A\ O_W\ B$), if for every point $b \in B$ there exists a point $a \in A$ so that a dominates or is equal to b and there exists at least a point $c \in A$ so that $c \notin B$.

Strong outperformance: the NS A strongly outperforms the NS B ($A\ O_S\ B$), if for every point $b \in B$ there exists a point $a \in A$ so that a dominates or is equal to b and there exists at least a pair of points $r \in A$ and $s \in B$ such that r dominates s.

Complete outperformance: the NS A completly outperforms the NS B ($A\ O_C\ B$), if for every point $b \in B$ there exists at least one point $a \in A$ so that a dominates b.

It is clear that $O_C \subset O_S \subset O_W$. The idea behind these ourperformance relations is to establish a minimum of what we expect from a comparison method. It is easy to understand that $A\ O_C\ B$ implies that A is a better NS than B, because for every vector in B there is a better one in A. So, if we have a comparison method R, and R evaluates B as better than A, than this method is not reliable. We expect the same respect to O_S, but in this case we made a weaker criteria because $A\ O_S\ B$ implies that no all vectors in B are dominated by vectors in A and some of the vector in B are also in A. The weakest criteria comes from O_W, because if $A\ O_W\ B$, no vector in B are dominated by vectors in A, and all vectors in B are in A. With all this in mind, Hansen and Jaszkiewicz [8] also define the property of *compatibility with an outperformance relation* O, where O can be O_W, O_S or O_C, as follows:

Compatibility. A comparison method R is (weakly) compatible with O if for two NSs A and B, $A\ O\ B$ implies that R will evaluate A as (not worse) better than B.

Also, there are two more desirable properties related to performance measures, *relativity* and *monotony.*

Relativity. A comparison method R has the property of (weak) relativity if adding a new non–dominated point to a NS A (do not decrease) improves its value $R(A)$.

Monotony. A comparison method R has the property of (weak) monotony if the evaluation of the true Pareto Front is (not worse) better than any other non dominated set.

The compatibility with the outperformance relations and with the properties described before is desirable because it makes a comparison method more robust to misleading cases, and with a behavior according to intuition. Note that if a comparison method is (weakly) compatible with O_W, then it has the property of (weak) monotony and (weak) relativity. This is because if we add a new non–dominated vector to a NS A and we call this new NS A', than $A'\ O_W\ A$, and the true Pareto front weakly ourperforms any other possible NS.

In this work we propose a performance measure that is weakly compatible with the ourperformance relations, needs no parameter tuning and no extra information of the multiobjective problem. It is an m–ary method. This means that it takes m NSs as argument, and evaluates them to decide what NS is better than the others.

The organization of the rest of the paper is as follows. In Section 2 we review some of the performance measures present in literature. In Section 3 we present our approach with detailed descriptions of its elements. In Section 4 we present several test cases we designed in order to test the behavior of different performance measures, included ours, and the result of the experiments. Finally, in Section 5 we state our conclusions based on the result of the experiments.

2 Related Work

Many performance measures (metrics, comparison methods) have been proposed in the past years in order to evaluate the quality of NSs. For example the Error Ratio [3], Generational Distance [3], Schott's Spacing Metric [11], U–Measure [1], C–Metric [12], $D1_R$ [14], and others. Different measures have different characteristics and some authors [4] [13] have analyzed their performance. Now we discuss some of the most popular ones.

S-metric (S). Proposed by [2] consist of the space enclosed by a NS and a fixed reference point r^*. For a minimization problem, the bigger the space the better the NS. It is a unary metric, in the sense that it takes only one NS as an argument and assigns a real value to it. This value is used to compare the NS with other NSs. The election of the reference point is vital for the good behavior of the S–metric. The evaluation of the S–metric can change depending on the position of r^*. Also, it is necessary to put r^* in some position where it will be dominated by all the elements of all the possible NSs to compare, otherwise we can have wrong results, such as negative values. Besides, it has some bias toward the central zone and convex zones of the Pareto Front. Its computational complexity is $O(n^{d/2})$. The S–metric has a lot of advantages, once the reference point has been correctly chosen. It has monotony, relativity, is compatible with all the outperformance relations, it is scale independent and has an intuitive meaning.

R **Metrics (R).** Hansen and Jaszkiewicz [8] proposed several performance measures based on utility functions. The general form is: $R = \int_{u \in U} W(A, B)p(u)du$, where A and B are NSs, $W(A, B)$ is a function that compares the best solution of A against the best solution of B according to a utility function u, $p(u)$ is the probability of occurrence of a utility function u and U is the domain of all possible utility functions for a problem. There are three versions of the R metric, the difference between them is the definition of W. If the utility functions are chosen wisely, these metrics are compatible with the outperformance relations. The only difficulty with the R metrics is that they need a lot of information, such as the utility functions and their distribution probability. This information is, in general, hard to obtain.

Modified ϵ–Indicator (Mϵ). Proposed by [13], Mϵ is defined as $M\epsilon(A, B) = ((I_\epsilon(A, B) \leq 1) \wedge (I_\epsilon(B, A) > 1))$, where $I_\epsilon(A, B)$ is the minimum factor for which we need to multiply all the elements of B, so for each one of these elements there exist an element in A that dominates it. It is a binary metric, in the sense that it takes two and only two NSs as arguments and determines if one is better than the other. It is designed to be compatible and complete respect to the outperformance relations. A comparison method R is compatible and complete respect to a outperformance relation O if it has two properties: is compatible with O and, if R evaluates A as better then B, this implies that $A\ O\ B$. If $\neg(A\ O\ B) \wedge \neg(B\ O\ A)$ a complete and compatible method can not make any conclusion from the NSs A and B. So, if $M\epsilon(A, B)$ is true, we conclude that A is better than B, if $M\epsilon(B, A)$ is true, than we conclude that B is better than A, if $\neg M\epsilon(A, B) \wedge \neg M\epsilon(B, A)$ we can not make any conclusion. Mϵ is easy to calculate and has a low computational complexity, but it is unable to evaluate pairs of NS when none of them outperform the other, even if one of them has only one point and the other has many.

When comparing $m > 2$ NSs with a binary method, it is necessary to compare each NS with all others for a total number of comparisons $O(m^2)$. The result of this comparisons must be interpreted and some times this comparisons are cycle inducing.

After this brief description of some of the performance measures y literature, we want to make some remarks. Several methods (some versions of R, the Error Ratio, Generational Distance and others), need a reference NS in order to make its evaluations. This reference is usually P* or the best NS known. This dependence on a reference can cause several problems, because if this information is not available these methods can not be used. Besides its evaluations can change depending on the reference chosen. Another important remark is that other metrics (R, S–metric), have parameters that need to be tuned in order to work correctly. The evaluations of these methods depend on the value of these parameters and usually it is necessary to have information of the multiobjective problem in order to establish proper values for these parameters. For the general case, when the only information we have is the NSs we are going to compare, many of these methods can not be used. So, it is desirable a performance measure able to work in the general case.

3 G-Indicator

In this paper we propose the G–Indicator (G), a m–ary performance measure. This method takes m NSs as argument and assigns a real number to every NS based on its convergence, dispersion and ED. We use these numbers to order the NSs from the best to the worst. G is weakly compatible with all performance measures (when comparing two NSs, it is compatible). Besides, it needs no extra information or parameter tuning, it only uses the information provided by the NSs to compare. Most important, the results of their evaluations agrees with the general sense of when a NS is better than another.

From now, we refer to the *known Pareto front*, in the context of our comparison method, as the set of non–dominated vectors from the union of the m NSs we are comparing.

In the following sections we describe the G–Indicator algorithm and its parts.

3.1 General Algorithm

The main idea of the G–Indicator is to create a partial order between the sets we are comparing, based on the complete outperformance. Then, to evaluate the non–comparable NSs based on their ED.

The converge of the NSs is evaluated grouping them by levels of complete outperformance. The details of how this grouping is decided is given later. For now, we only state that a NS in a level of complete outperformance is considered better than a NS in an inferior level. The NSs in the same group do not complete outperform each other. We call this procedure the *convergence component* and we chose complete outperformance instead of strong or weak, because it is the strongest of the three relations.

Once separated by levels, we need to evaluate what NSs in the same level are better than the others. For this we use a procedure we call the *DE component.* In this component we introduce a novel method to measure the DE. We describe this component later. Based on DE, this component assigns a real number (I, or I_A) to each NSs A in the same level. The bigger this number, the better DE of the NS.

After all this, we have that every NS A have two values associated: its level of complete outperformance and its value of I. Our final goal is to assign a real number $G(A)$ that considers convergence and DE, so we can order them from the best to the worst. The first idea to generate this number is to make $G(A) = I_A$, but this is a mistake, because a NS in a level of complete outperformance may have a value of I smaller than that of a NS in an inferior level. To solve this, we make $G(A) = I_A + K_A$ where K_A is a compensation that depends on the level of A. This compensation is designed in such a way that the value of G of a NS is always bigger than those of NSs in inferior levels. And, at the same time, the value of G stays bigger than those of NSs in the same level with worse DE.

As we see later, the DE component combines the values of the different objective functions in a single number. This makes the value of DE sensible to the scale of the functions. To solve this problem we make a normalization of all elements of the NSs.

With all this in mind, the general algorithm of G is as follows. Let $A_1, A_2, \ldots, A_m$ be m NSs to compare:

1. Scale the values of the vectors in the NSs (see Section 3.2).
2. Group the NSs by levels of complete outperformance (Convergence component, see Section 3.3).
3. For each level of complete outperformance and for every A_i in the level, calculate the zone of influence I_{A_i} (DE component, see Section 3.4).
4. For every A_i, $G(A_i) = I_{A_i} + K_{A_i}$, where $G(A_i)$ is the value of G for the A_i and K_{A_i} is a compensation that depends on the level of complete outperformance of A_i (see Section 3.4.2).

All parts of this algorithm will be explained in detail in the following subsections.

3.2 Scale and normalization.

A very important detail we must consider is the scale of the objective functions. If an objective function has a bigger scale than the others, its influence will be more important. For this, the first step in our algorithm is a normalization.

For this normalization we do not use the maximum and minimum value for all vector in all NSs for each objective function. Instead, we use the maximum and minimum value of the known Pareto front. The reason for this is that dominated vectors can have high values, adding noise to the comparison if we use them in the normalization.

The algorithm we use to normalize is the following:

1. Take the union of the m sets, $C = \bigcup_{i=1}^{M} A_i$.
2. From C take its non-dominated elements. C^* =*ND(C)*.
3. Find max_i and min_i as the max and min value respectively, for the component i for all points $p \in C^*$.
4. Using max_i and min_i make a linear normalization of all points in all sets A_i.

The computational complexity for this normalization is $O(| \cup_{i=1}^{m} A_i|^2)$, where m is the number of NSs. This complexity is always less or equal than $O(m^2|A_{max}|^2)$, where A_{max} is the NS with more elements.

We note that this normalization do not need extra information besides that provided by the NSs themselves.

3.3 Convergence Component

As mentioned before, convergence is the most important characteristic of a NS. So, it is obligatory to include a mechanism to measure this property in a performance measure. We use the complete outperformance as the motor of our performance operator.

Our convergence operator classifies the NSs by levels, based on the complete outperformance between every NS and the rest of the them. If we have m sets, the first level L_1 includes those NSs A_k so that $\neg(\bigcup_{i=1}^{m} A_i \; O_C \; A_k)$. The following levels include those NSs that are completely outperformed only by the union of NSs in previous levels.

The idea is to put in the first level those NS that apport al least one vector to the known Pareto front. As the other NS have all its elements dominated by the known Pareto front, we consider them as disposable, because all their information is suboptimal. Then, we repeat the process considering only the NSs with suboptimal vectors.

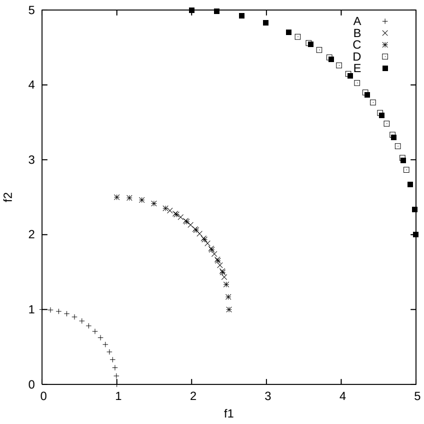

Figure 1: Five NSs, three levels of dominance

The algorithm is as follows:

Given a set $D = \{A_1, A_2, ..., A_m\}$ where A_i is a NS.

1. Make $j = 1$.
2. Make $L_j = \{\}$
3. Extract from D and put in L_j, those A_i such that $\neg(\bigcup_{A_k \in D} A_k \; O_C \; A_i)$.
4. If D is not empty, make $j = j + 1$ and return to step 2.
5. End.

Those $A_i \in L_1$ are in the first level, those $A_i \in L_2$ are in the second level and so on. If $A \in L_j$, $B \in L_k$, and $j < k$ we consider A better than B (A is in a superior level than B). As an example, in Figure 1, there are five NSs, A, B, C, D and E. For this case, we have three levels, where $L_1 = \{A\}$, $L_2 = \{B, C\}$ and $L_3 = \{D, E\}$.

This convergence operator creates a partial order in the NSs and it is compatible with O_C, because if $A \; O_C \; B$, then A will be in a superior level than B.

The computational complexity of this component in the worst case is $O(m^3|A_{max}|^2)$. This occurs in cases like the one of Fig. 8, where every NS is in a different level of complete outperformance.

3.4 Dispersion–Extension Component

A very difficult problem multi–objective optimization is how to measure the DE of a NS. In this work, we introduce a novel method to measure the DE of a NS, that have an excellent performance, as it is shown in Section 4.

This method consist on to associate to each element p of a NS S, a circle with a fixed radius U and center p. We call this circle *zone of influence* (I_p) . If the elements of S have a poor ED, many of them will be near each other and the circles associated will intersect. As a result, the area of the union of these circles (I_S) will be small. Suppose now that we relocate the elements of S to improve the DE. We do this by increasing the extension and distance between elements, and/or making more uniform their distance. As we improves the DE, the intersections of the circles will decrease and, at the same time, both the DE and I_S will increase. So we have that the area of the union of circles is directly proportional to the DE. We can conclude that this area is a good DE indicator. This definition can be extended to any dimension, changing the circles by spheres and hyper–spheres. In the next paragraph we explain this idea in a more general form.

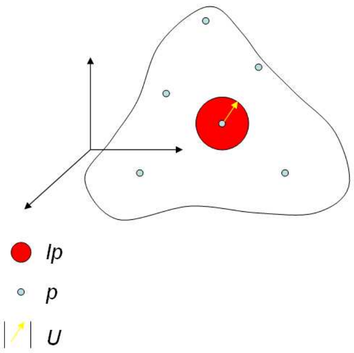

Figure 2: An example of Ip

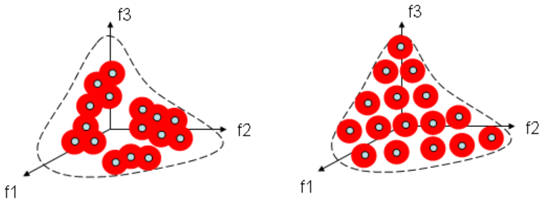

Figure 3: Size of I according to dispertion.

Consider the region around every element p of a NS that consists of all the points whose distance is inferior to a limit U. For example, the circle around p in Figure 2. Based on this, we give the following definitions:

Zone of influence of point p_i (I_{p_i}). It is the set of points whose distance from p_i is equal or less than a real, positive number U.

Zone of influence of the set S (I_S). It is the union of the I_{p_i} for all $p_i \in S$.

In general, we refer to a zone of influence as I.

Now, we review the behavior of I for a NS according to the configuration of its elements. In Figure 3, the dashed line represents the contour of the true Pareto front for a problem with three objective functions, the small circles are the elements p of the NS, and the big circles represent the I_p. If U is chosen wisely (we shall explain later how to choose U), the zone of influence for the set on the left is small due to the bad distribution of its points. In contrast, the I for the set on the right is bigger because of the good distribution of its elements. The behavior of the I is exactly what we expect from a good diversity measure.

The value of I_S is proportional to the DE of the set S, so it is a good indicator of this characteristic. It is important to say that the I_S is inversely proportional to the overlap between the I_{p_i} for $p_i \in S$. NSs with a good DE have less overlap than those with a bad one.

Calculating I_S for a NS S is complex, especially for NSs with many elements in high spatial dimensions. In order to reduce the complexity of this calculation, we first project the NS to a lower dimension. As we show later, this projection makes easier both the calculation of the I_S and the demonstration of its compatibility with the outperformance relations. After a projection, there is a loss of information, but the loss in this case is very

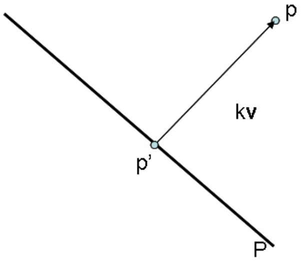

Figure 4: An example of an Standard Projection in 2d.

small. The reason of this is because we project in a direction of improvement of all objective function. As we explain before (Section 1), the information of the DE is orthogonal to this direction, so most of the information related to DE is preserved. The projection we use is described in the next subsection.

3.4.1 Standard Projection (*SP*).

Let the vector $\mathbf{v} \in R^n$ whose components are equal to 1, and the vectorial subspace $P \in R^n$ whose null space is generated by $\mathbf{v}$. Let the point $p \in R^n$ and the line $L_p \in R^n$ which is parallel to $\mathbf{v}$ and pass through p. We call the Standard Projection of p ($SP(p)$) to the point intersection of L_p with P.

An example of a Standard Projection in R^2 is shown is Figure 4. The formula to calculate $p' = SP(p)$ is $p' = p - K\mathbf{v}$, where K is $\frac{(\mathbf{p}.\mathbf{v})}{|\mathbf{v}|^2}$ and $\mathbf{p}$ is the vector that goes from the origin to p.

This is the projection we use before the computation of I. As all points projected are in a same line (in 3d they are in a plane, in more than 3d they are in a hyperplane), the union of the $I_{p'}$ for the projected points p' is easier to compute. We use the area of the union of the projected points as an approximation of that of the original points.

3.4.2 Computing the Zone of Influence I.

After a SP, all projected point p' are in the subspace P (in two dimensions P is a line, in three dimension it is a plane, etc.). We can define domains of integration in P for all projected points and calculate $I_{p'}$ as an integral in the corresponding domain. Finally, we sum the value of all these integrals to obtain the total value of I. The general algorithm is the following:

From now we represent the Euclidean distance between two points, a and b, as $d(a, b)$. Now, given a set of points S and a limit U.

1. For each $p_i \in S$ calculate its projection p'_i.
2. For each p'_i calculate:
$$I_{p'_i} = 2 \int_{\Omega_i} \sqrt{U^2 - d(u, p'_i)^2}\ d\Omega_i \tag{1}$$
where Ω_i is the set of points u in the vectorial space P so that $d(p'_i, u) \leq U$ and $d(p'_i, u) \leq d(p'_j, u)$ for $p'_j = SP(p_j)$, $p_j \in S$ and $j \neq i$.
3. Finally, calculate I_S as:
$$I_S = \sum_{i=1}^{|S|} I_{p'_i} \tag{2}$$

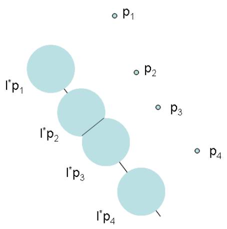

Figure 5: I

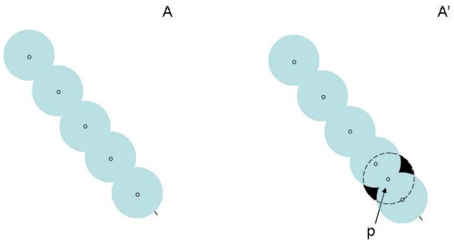

Figure 6: When a new point p is added to a NS, the I of the former increases

An example of I_S in 2d is shown in Figure 5.

In two dimension, this calculation is easy to compute. We have all projected points in a line, so we can sort the points and define the domains of the integration easily. For three dimensions, all projected points are in a plane. So, in order to define the domains of integration, we can calculate the Voronoi diagram for the projected point and use the polygons of Voronoi to establish the domains. More of this in Section A and B. For more than three dimension we recommend the use of a Monte Carlo method for the evaluation of the integral.

It is important to note that this definition of I implies that every different point adds at least a small amount to the size of I, as it can be seen in Figure 6, where the NS A' is equal to $A \cup p$. $I_{A'} > I_A$ because of the contribution of p. Besides, the standard projection guaranties that every different point have a different projection (see Section 3.6), so the I_S for a NS S improves its values as we add more points to S.

3.4.3 The parameter U.

As we mentioned before, the overlap between I_p is very important to the good behavior of I as a DE measure. At the same time, the overlap depends on the parameter U, so the numeric value of U must be chosen carefully. It is important to allow some level of overlap, so we can discriminate between sets with bad and good DE. If U is too small the overlap will be zero and the value of I will not be related to the distribution of the points. In this work we propose the following value of U for m NSs S_j:

$$U = 0.5\frac{\sum_{j=1}^{m}\sum_{i=1}^{|S_j|} r_{ij}}{\sum_{j=1}^{m}|S_j|} \tag{3}$$

where r_{ij} is the mean of the distance between $p_i \in S_j$ and its nearest neighbors. This value of U produces at least a small amount of overlap in the I_p of the NS with the worst DE. It is important to note that U depends on all the m sets, if a NS is added or eliminated, it must be recalculated. The definition of nearest neighbors we use is that given in [1].

3.4.4 Computational Complexity of the Zone of Influence I.

The computational complexity of this component depends on the complexity of its different parts. The standard projection has a linear cost for every NSs. The parameter U has a cost of $O(m|A_{max}|^2)$, where m is the number of NSs, and A_{max} is the NS with more elements. The cost of the integral 1 is $O(m|A_{max}|log|A_{max}|)$ for 2d and 3d (because of the cost of sorting $|A_{max}|$ points and the cost of the construction of a Voronoi diagram). We do not consider in this analysis the cost of a Monte Carlo integration. So the maximum cost of the DE component is $O(m|A_{max}|^2)$.

3.5 Computing G–Indicator and its Cost

Once we have explained all the components of the G–Indicator, we give a more detailed description of the general algorithm given in Section 3.1:

Given m non–dominated sets, A_1, A_2, ... , A_m.

1. Normalize all sets as described before.
2. Classify all sets by levels of dominance (convergence operator).
3. For $k = 1 : R$, where R is the number of levels.
   (a) For every $A \in L_k$ eliminate all points $p \in A$ dominated by (an)other point(s) $q \in B$ for any $B \in L_k$.
   (b) Calculate U based on all $A \in L_k$.
   (c) Calculate the I_A for each $A \in L_k$.
   (d) Calculate a compensation $D_k = max(I_A)$ for $A \in L_k$.
4. For $k = 1 : R - 1$ where R is the number of levels. For all $A \in L_k$ the value of the G–Indicator is:

$$G(A) = I_A + \sum_{i=k+1}^{R} (D_i + \epsilon) \ . \tag{4}$$

   where ϵ is a real constant greater than zero.

The compensation D and the constant ϵ in formula (4) are necessary to keep the order between the dominance levels and the compatibility with O_C. This way, no NS from an inferior level with a high I is better than a NS from a superior level with low I.

Note that in the calculation I we do not consider vectors dominated by other vectors in the same level of complete outperformance. This is because we want to calculate I using only information of non–dominated vectors.

In essence, the G–Indicator creates a partial order for the NSs and resolves the ties using a diversity operator.

The four main parts of this algorithm (normalization, convergence component, DE component and compensation) are sequential. Once a part is executed, it is no necessary come back to it again. This makes easier to calculate the computational complexity of the whole algorithm, because it is the sum of the complexity of all components. So, the computational complexity of the G–Indicator is equal to $O(m^2|A_{max}|^2+m^3|A_{max}|^2+m|A_{max}|^2)$.

3.6 Compatibility with the outperformance relations

Now, we demonstrate the compatibility of the G–Indicator with the outperformance relations. We demonstrate that our approach is compatible with O_W when comparing only two NSs, (as a binary measure). After, we demonstrate its weak compatibility with O_W for more than two NSs.

First we prove that the there is a mapping between a NS and its standard projection (Section 3.4.1).

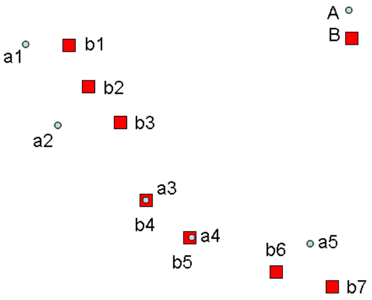

Figure 7: Two non-dominated sets.

Lemma. *Let $a, b \in A$ where A is a NS, $a' = SP(a), b' = SP(b)$ and $a \neq b$. Then $a' \neq b'$.*

Proof. We prove this by contradiction. Let suppose that $a' = b'$, then:

$$\begin{aligned} a' &= b' \\ a - K_1\mathbf{v} &= b - K_2\mathbf{v} \\ a &= b + (K_1 - K_2)\mathbf{v} \\ a &= b + K\mathbf{v} \end{aligned}$$

where $K = (K_1 - K_2)$. As K is a real number, we have three cases, $K > 0$, $K < 0$ and $K = 0$. If $K > 0$ then we have $a_i = b_i + K$ for $i = 1 : n$ (remember, all members of $\mathbf{v}$ are equal to one), so $b_i < a_i$, but this implies that b dominates a, a contradiction to the initial condition A is a NS. If $K < 0$ then we have $a_i + K = b_i$ for $i = 1 : n$, so $a_i < b_i$, but this implies that a dominates b, a contradiction to the initial condition A is a NS. If $K = 0$ then we have $a = b$, a contradiction to the initial condition $a \neq b$. As a result $a' \neq b'$.

□

The importance of this theorem is that every different point $p \in A$ will apport al least a small quantity to the value of I_A, because its projection p' will be different to those of other points. And, as discuted in Section 3.4.2, every different point apport at least a small amount to I.

Continuing with our explanation of the compatibility of our approach with the outperformance relations, we define the following subsets of index for two NSs, A and B:

J: it is the set of index of the vectors $a_j \in A$ so that $a_j = b_r$ for some vector $b_r \in B$.

K: it is the set of index of the vectors $a_k \in A$ so that a_k is dominated by some vector $b \in B$.

L: it is the set of index of the vectors $a_l \in A$ so that a_l is not in B and a_l is not dominated by any vector in B.

R: it is the set of index of the vectors $b_r \in B$ so that $b_r = a_j$ for some $a_j \in A$.

S: it is the set of index of the vectors $b_s \in B$ so that b_s is dominated by some vector $a \in A$.

T: it is the set of index of the vectors $b_t \in B$ so that b_t is not in A and b_t is not dominated by any vector in A.

As an example, for the NSs A and B in Figure 7, $J = \{3, 4\}$, because a3 = b4 and a4 = b5, $K = \{5\}$ because a5 is dominated by elements of B, $L = \{1, 2\}$ because a1 and a2 are elements of A not present in B and they are not dominated by elements of B. Similarly $R = \{4, 5\}$, $S = \{1, 2, 3\}$ and $T = \{6, 7\}$.

In order to simplify the following explanations we introduce some terminology. We represent an element of the sets just described, with the corresponding lower case. For example j represents an element of J. Also, we represent the union of the $I's$ for all elements of A that are also present in B (vectors in A whose index are in J) as $\bigcup I_{a_j}$, according to the example in Figure 7, $\bigcup I_{a_j} = I_{a_3} \cup I_{a_4}$. The union of I_p for the elements of B that are dominated by some element in A (vectors in B whose index are in S) is $\bigcup I_{b_s}$ and according to the example in Figure 7, $\bigcup I_{b_s} = I_{b_1} \cup I_{b_2} \cup I_{b_3}$. It is clear that $\bigcup I_{a_j} = \bigcup I_{b_r}$.

According to the nomenclature just described, we represent the value of I_A as $I_A = \left(\bigcup I_{a_j}\right) \cup (\bigcup I_{a_l})$. Similarly, the value of I_B is $I_B = (\bigcup I_{a_r}) \cup (\bigcup I_{a_t})$.

The sets K and S are not used, because they contain the index of dominated elements. Now we demonstrate the compatibility of the G-Indicator with the outperformance relations.

Theorem. *When comparing two NS, A and B, G is compatible with O_C, O_S and O_W.*

Proof. If $A\ O_C\ B$, this implies that A is in a better level of complete outperformance than B because of the convergence component of G (see Section 3.3), so A is always evaluated as better than B. If $A\ O_S\ B$, then T is an empty set and L contains at least one element, so:

$$\begin{array}{rcl} \left(\bigcup I_{a_j}\right) \cup (\bigcup I_{a_l}) & > & \bigcup I_{b_r} \\ I_A & > & I_B \end{array}$$

The inequality comes from the fact that every different point apport at least a small amount to the total I of a NS.

If $A\ O_W\ B$, then T is again empty and L contains at least one element, so this case is reduced to the previous one.

□

This proves that G is compatible with the outperformance relations when it evaluates two NSs. Now, respect to more than two NSs the G–Indicator is compatible with O_C, this is because of the convergence operator. Respect to O_S and O_W, G is weakly compatible, because suppose that there are three NSs A, B and C in the same level of complete outperformance and that $A\ O_S\ B$ or $A\ O_W\ B$. If we only consider A and B, this means that T is empty and L has at least one element, so $I_A > I_B$. But when we consider C, it is possible that all a_l are dominated by some elements in C, so the elements in L are not considered and $I_A = I_B$. It does not matter how many NSs we add to the comparison, if an element of A whose index is in J is dominated, the corresponding element in B whose index is in R is dominated too and the equality holds. As a consequence I_A will not be smallest than I_B, so the weak compatibility with O_S and O_W is guaranteed.

4 Experiments and Results

In order to evaluate the performance of our approach, we designed some examples that consider various topologies of the Pareto front. We compare our results with those of the S-metric and the $M\epsilon$, both methods described in section 2. We chose the S–metric because it stands out as one of the most popular in literature. We compare with the $M\epsilon$ because it is a representative of the measures that are compatible and complete, and we consider interesting to see the behavior of these measures.

We designed our test cases so it is easy to see which NS is better than the others, or if there are NSs that are equal, so it is possible to decide the right order from the best to the worst NS. The challenge for the performance measures is to evaluate the NSs in such a way that we can construct the right order. For the G–Indicator, we order the NS from the bigger to the smaller value of their corresponding evaluation of G. The same for the S–metric. For the $M\epsilon$ the interpretation of its evaluations is a little more difficult, because it is a binary operator. Suppose we are comparing the NSs A, B and C, if $M\epsilon(A,B)$ is true, $M\epsilon(A,C)$ is true and $M\epsilon(B,C)$ is true, we order from best to worst according to $M\epsilon$ as A, B and C. Special cases like cycles and when $M\epsilon$ are not able to make conclusions will be discussed in the experiment.

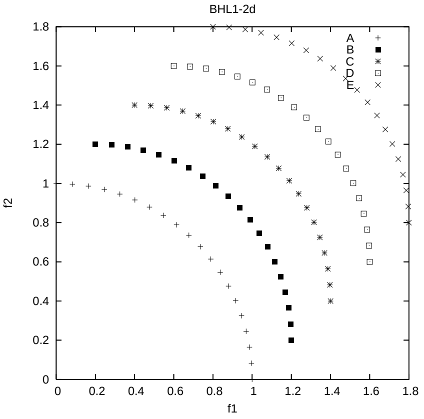

Figure 8: Experiment 1 in 2d

Table 1: Results of the Experiment 1

2d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	4.512	1°	0.751	1°
B	2°	3.409	2°	0.541	2°
C	3°	2.307	3°	0.356	3°
D	4°	1.204	4°	0.195	4°
E	5°	0.102	5°	0.060	5°

3d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	4.182	1°	0.903	1°
B	2°	3.146	2°	0.605	2°
C	3°	2.109	3°	0.374	3°
D	4°	1.073	4°	0.199	4°
E	5°	0.036	5°	0.074	5°

In the Tables, we show the numeric value of G and S–metric, under the column "V" (value) and the order according to these values under the column "R" (rank). The $M\epsilon$ only has a "R" column.

For each test case we made a 2d and a 3d version, except for Experiment 8 with only a 2d version. We consider all the test cases as minimization problems. The result of the experiments are the following.

Experiment 1. The goal of this experiment is to evaluate the ability of the algorithms to detect various levels of complete outperformance. There are five NSs, A, B, C, D and E, with the same DE and size. A O_C B O_C C O_C D O_C E. So, from the best to the worst, the order is A, B, C, D, E. The 2d version of the NSs is shown in Figure 8.

We can see in Table 1 that all measures identify correctly the levels of complete outperformance of the five NSs.

Experiment 2. In this experiment (Figure 1) we combine complete outperformance with extension. There are five NSs A, B, C, D and E, where A O_C B, C, D and E. Besides B and C O_C D and E. B and C have the same convergence, but C has a better extension than B. D and E have the same convergence, but E has a better

Table 2: Results of the Experiment 2.

2d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	3.079	1°	0.967	1°
B	3°	1.895	3°	0.452	2°
C	2°	1.937	2°	0.566	2°
D	5°	0.583	5°	0.065	3°
E	4°	0.750	4°	0.067	3°

3d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	2.680	1°	0.995	1°
B	3°	1.615	3°	0.368	2°
C	2°	1.643	2°	0.496	2°
D	5°	0.347	5°	0.059	3°
E	4°	0.572	4°	0.094	3°

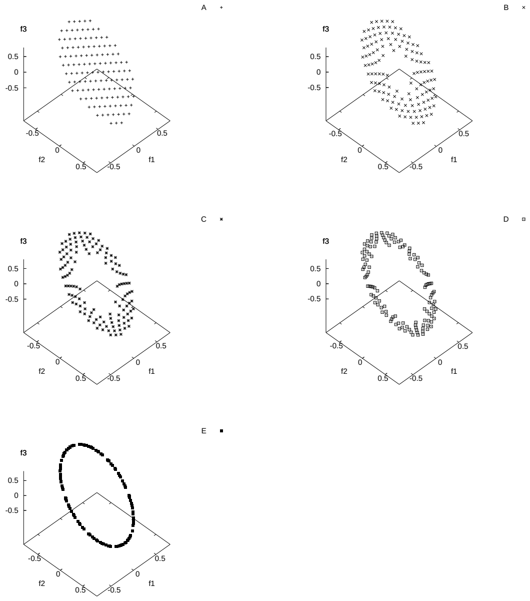

Figure 9: Experiment 3 in 3d.

extension than D. We conclude that from the best to the worst, the order of the sets is A, C, B, E, D.

Both the G and S passed the test finding the correct order of the sets, as it can be seen in Table 2. But the $M\epsilon$ was not able to detect the difference between B and C, and between D and E (in the table, we represent this by putting B and C in second place, and D and E in thirth place. This does not mean that $M\epsilon$ considers those NSs as equals, but that $M\epsilon$ can infer nothing about those NSs), so it failed the test.

Experiment 3. In order to test the ability of the metrics to detect holes in the NSs, we create this experiment . It consists of five NSs with circular shape and with a hole in the center (Figure 9). The size of the hole is different for different NSs. We consider that the bigger of the hole, the worse the NS. So, from the best to the

Table 3: Results of the Experiment 3

2d					
SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	0.212	1°	0.462	1°
B	2°	0.169	2°	0.419	1°
C	3°	0.134	3°	0.345	1°
D	4°	0.097	4°	0.235	1°
E	5°	0.039	5°	0.000	1°

3d					
SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	0.0131	2°	0.432	1°
B	2°	0.0129	1°	0.425	1°
C	3°	0.0123	3°	0.400	1°
D	4°	0.0110	4°	0.361	1°
E	5°	0.0023	5°	0.293	1°

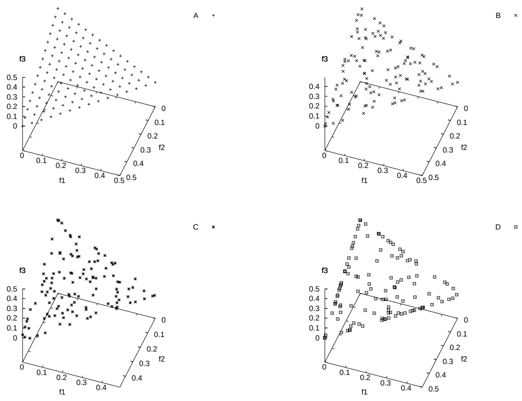

Figure 10: Experiment 4

worst, the order of the sets is A, B, C, D and E.

The result of the experiment is shown in Table 3. Again, G and S passed the test, ordering the NSs correctly. $M\epsilon$ was not able to find any difference between any pair of NS, failing the test.

Experiment 4. This case is based in problem DTLZ1 [9]. In this problem, the Pareto front consists of all the points p with components $p^k \geq 0$ and $\sum_{i=1}^{d} p^i = 0.5$, all coordinates positive. There are five NSs, where A has the best dispersion. The other NSs were obtained adding different levels of noise to the positions of the points in A. We consider that the bigger of the noise in the NS, the worse its dispersion. Figure 10 shows four of the five NSs in 3d. The convergence for all NSs is the same. From the best to the worst, the order of the sets is A, B, C, D and E.

The result of the experiment is shown in Table 4. Again, G and S passed the test, while $M\epsilon$ failed it.

Experiment 5. In this experiment (Figure 11) we evaluate the ability of the algorithms to detect the levels of weak outperformance. We created five set A, B, C, D and E where E O_W D O_W C O_W B O_W A, so from the best to the worst the order of the sets is E, D, C, B, A.

Table 4: Results of the Experiment 4

2d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	0.107	1°	0.464	1°
B	2°	0.096	2°	0.460	1°
C	3°	0.068	3°	0.421	1°
D	4°	0.064	4°	0.322	1°
E	5°	0.035	5°	0.262	1°

3d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	0.0212	1°	0.795	1°
B	2°	0.0177	2°	0.785	1°
C	3°	0.0172	3°	0.784	1°
D	4°	0.0134	4°	0.780	1°
E	5°	0.0110	5°	0.769	1°

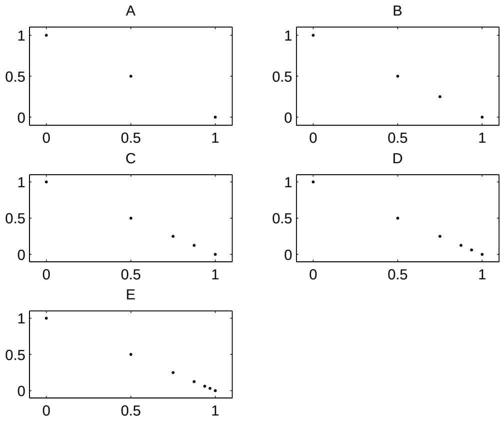

Figure 11: Experiment 5.

Considering that all measures are compatible with O_W (G-Indicator is weakly compatible), it is not surprising that all of them found the correct order of the sets as it can be seen in Table 5.

Experiment 6. The goal of this experiment is to compare the sensibility of the measures to the convexity of the Pareto front. Both NSs, A and B have the same DE and convergence, but A is on a non-convex zone while B is on a convex zone (Figure 12). We expect the same value for both NSs.

It is clear, according to Table 6, that S-metric has a bias towards the convex zones of the Pareto's front thus it failed the test. The G-Inidicator is not affected by the convexity of the sets, so it gave the same value to both NSs passing the test. The $M\epsilon$ again was not able to make any conclusion from the NSs failing the test. It is important to remember that when G or S put several NSs in the same position in the tables (as in this case, G put all NSs in the first position), it means that those NSs received the same evaluation, so they are equal. When $M\epsilon$ puts several NSs in the same position, it means that it was not able to make a decision about these NSs. A compatible and complete method is not designed to detect ties.

Table 5: Results of the Experiment 5

2d					
SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	5°	0.341	5°	0.250	5°
B	4°	0.450	4°	0.312	4°
C	3°	0.469	3°	0.328	3°
D	2°	0.470	2°	0.332	2°
E	1°	0.471	1°	0.333	1°
3d					
SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	5°	0.067	5°	0.000	5°
B	4°	0.152	4°	0.233	4°
C	3°	0.157	3°	0.256	3°
D	2°	0.162	2°	0.260	2°
E	1°	0.169	1°	0.261	1°

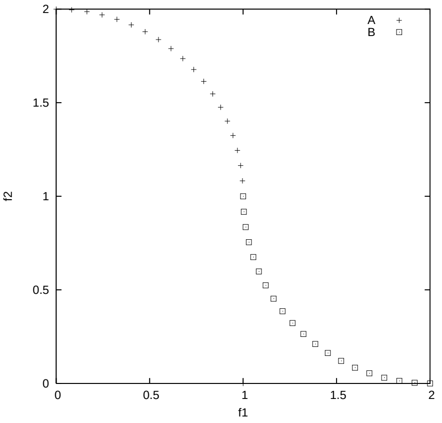

Figure 12: Experiment 6.

Table 6: Results of the Experiment 6

2d					
SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	0.0256	2°	0.298	1°
B	1°	0.0256	1°	0.440	1°
3d					
SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	0.0111	2°	0.174	1°
B	1°	0.0111	1°	0.364	1°

Experiment 7. To evaluate the sensibility of the performance measures to the relative position of the different NSs, we create five NSs with the same convergence and DE but with different positions on the Pareto front. An image of the 2d version is shown in Figure 13.

As it is clear from Table 7, S gave different values to different NS, so it failed the test. $M\epsilon$ again was not able to make conclusions about the NSs. G gave the same value to all NSs, so it was the only method that passed

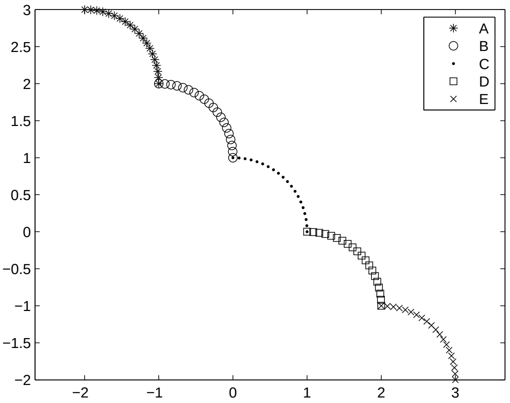

Figure 13: Experiment 7

Table 7: Results of the Experiment 7

2d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	0.0041	3°	0.167	1°
B	1°	0.0041	2°	0.287	1°
C	1°	0.0041	1°	0.327	1°
D	1°	0.0041	2°	0.287	1°
E	1°	0.0041	3°	0.167	1°

3d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	7.08e-5	3°	0.195	1°
B	1°	7.08e-5	1°	0.251	1°
C	1°	7.08e-5	2°	0.211	1°
D	1°	7.08e-5	3°	0.123	1°
E	1°	7.08e-5	4°	0.035	1°

Table 8: Results of the Experiment 8

2d

SET	G–Indicator		S–measure		$M\epsilon$
	R	V	R	V	R
A	1°	0.156	1°	0.421	1°
B	2°	0.123	2°	0.189	1°

the test.

Experiment 8. We created this experiment to see the behavior of the performance measures with crossed NSs. An image of the expeiment is shown in Figure 14. Set B has better distribution but A has better convergence. The right order is A, B. Only a 2d case was made.

From Table 8, we can conclude that the S–metric and the G–Indicator gave the correct order for the NSs, while the $M\epsilon$ failed the test.

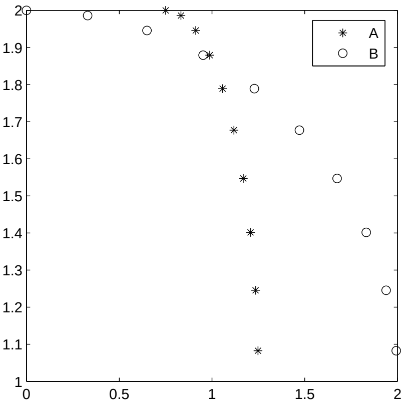

Figure 14: Experiment 8

5 Conclusions

We presented the G–Indicator, and m–ary performance measure for non–dominated sets. It does not need any extra information, neither it needs further parameter tuning. Besides, it combines successfully convergence and diversity in a single number, and its evaluations agree with intuition giving better scores to NSs with better convergence, extension and dispersion. It is weakly compatible with the ourperformance relations. It is robust in misleading cases, like NSs with convex and non–convex zones. In order to evaluate G, we created several test cases. In all of them our approach gave the correct answer showing a better performance than the S-measure, and the $M\epsilon$ other performance measures that represents the state of the art in the literature. We recommend the use of complete and compatible methods only if combined with a good diversity operator, because as they do not consider diversity, can became useless when comparing NSs with similar convergence, as it can be seen in the experiments.

A How to calculate I in 2d

The evaluation of the integral 1 (see Section 3.4.2) in 2d is easy, because the space P where all points are projected is a line. All p'_i are aligned and the domains Ω_i are easy to determine. First we rotate the P to make it horizontal, and order the points according to their horizontal position. Then, for every p'_i we calculate the distances with its neighbors $d(p'_{i-1}, p'_i)$ and $d(p'_i, p'_{i+1})$. If $d(p'_{i-1}, p'_i) > 2U$ or p'_i is the leftmost element, than the inferior limit l_i, of the integral for p'_i, is equal to $-U$. Otherwise $l_i = d(p'_{i-1}, p'_i)/2$. If $d(p'_i, p'_{i+1}) > 2U$ or p'_i is the rightmost element, than the superior limit s_i, of the integral for p'_i, is equal to U. Otherwise $s_i = d(p'_i, p'_{i+1})/2$.

Finally, translating the position of p_i to the origin, the integral 1 is simplified as:

$$I_{p_i} = 2\int_{l_i}^{s_i} \sqrt{U^2 - x^2}dx \tag{5}$$

Then we make the following transformation, $x = U\cos(\theta)$, $dx = -U\sin(\theta)$. Where $x = l_i$, $\theta_{il} = \arccos(l_i/U)$. When $x = s_i$, $\theta_{is} = \arccos(s_i/U)$. So the integral 5 is transformed and evaluated as follows:

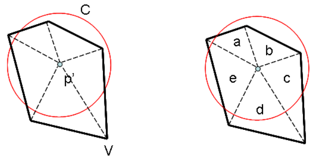

Figure 15: Domains for the integral 1

$$\begin{aligned}
I_{p_i} &= 2\int_{\theta_{il}}^{\theta_{is}} -\sqrt{U^2 - U^2\cos(\theta)^2}U\sin(\theta)d\theta \\
&= 2\int_{\theta_{il}}^{\theta_{is}} -\sqrt{U^2(1-\cos(\theta)^2)}U\sin(\theta)d\theta \\
&= 2\int_{\theta_{il}}^{\theta_{is}} -\sqrt{U^2\sin(\theta)^2}U\sin(\theta)d\theta \\
&= 2\int_{\theta_{il}}^{\theta_{is}} -U^2\sin(\theta)^2 d\theta \\
&= 2U^2\int_{\theta_{il}}^{\theta_{is}} -0.5(1-\cos(2\theta))d\theta \\
&= -U^2(\theta - 0.5\sin(2\theta))|_{\theta_{il}}^{\theta_{is}}
\end{aligned}$$

The sum of this integral for all points in a NS S is the value of I_S in 2d.

B How to calculate I in 3d

The evaluation of I in 3d is more complicated than in 2d. The procedure we use is to calculate the Voronoi diagram of the projected points p'. This is possible because for 3d, all points p' are in the same plane P. Then, we use the Voronoi polygons to determine the domain of the integral 1.

If we consider the circumference C in P with radius U and center in a point $p' \in S'$ and the Voronoi polygon V for p' (Figure 15, left), the domain Ω for the integral 1 (see Section 3.4.2) is the intersection of the points inside C and the points inside V. In order to simplify the evaluation of the integral, we divide the domain in several triangles (subdomains) as shown in Figure 15, right. So we evaluate the integral in every triangle and add the results. Depending on how C and V intersect, we have several cases for the evaluation of integral 1 in the subdomains. Now we review these cases

B.1 Case 1

Case 1 occurs when we have a domain as the triangle d, (Figure 15, right). In this case C does not intersects the edge opposed to p', but intersects both the edges that coincides in p'. If we use polar coordinates and translate p' to the origin, integral 1 is reduced to:

$$I_d = 2\int_{\theta_1}^{\theta_2}\int_0^U \sqrt{U^2 - r^2}\; r dr d\theta \tag{6}$$

where θ_1 and θ_2 are the angles between the polar axis and the edges of the triangle that coincides in p'. Evaluating this integral we have:

$$I_d = 2U^3\theta_f/3 \tag{7}$$

where θ_f is the angle of the vertex in p'.

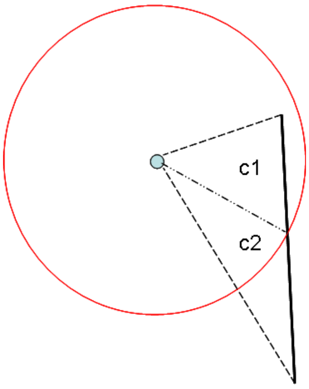

Figure 16: It is possible to divide a triangle in two subdomains

B.2 Case 2

Case 2 occurs with domains like in triangle b, (Figure 15, right), in this case C does not intersect any edge of the triangle. We made a translation so p' coincides with the origin, and transformed to polar coordinates. The integral 1 takes the form:

$$I_b = 2\int_{\theta_1}^{\theta_2}\int_0^{\frac{A}{\sin(\theta)-B\cos(\theta)}} \sqrt{U^2 - r^2}\; r dr d\theta \tag{8}$$

where θ_1 and θ_2 are defined as in case 1, $r = \frac{A}{\sin(\theta)-B\cos(\theta)}$ is the equation in polar coordinates of the line that pass through the edge of the triangle opposed to p'. B is a constant factor and A is the inclination of the line in rectangular coordinates. After evaluating the inner integral, we have:

$$I_b = \frac{2}{3}\int_{\theta_1}^{\theta_2}\left(U^2 - \left(\frac{A}{\sin(\theta) - B\cos(\theta)}\right)^2\right)^{\frac{3}{2}} d\theta \tag{9}$$

we evaluate this integral using a numeric method.

B.3 Case 3

The triangle c in Figure 15 (right) is an example of this case , where the circumference C intersects the edge opposed to p' in just one point. As it is shown in Figure 16, the domain can be reduced to two sub–domains, *c1* and *c2*, *c1* is identical to case 2 and *c2* is identical to case 1.

B.4 Case 4

This is the case of the triangle e in Figure 15 (right), where the circumference C intersects the line opposed to p' in two points. As it is shown in Figure 17, a domain can be reduced to three subdomains, *e1*, *e2* and *e3*, *e1* and *e3* are identical to case 1 and *e2* is identical to case 2.

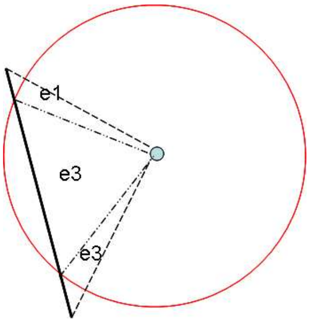

Figure 17: It is possible to divide a triangle in three subdomains

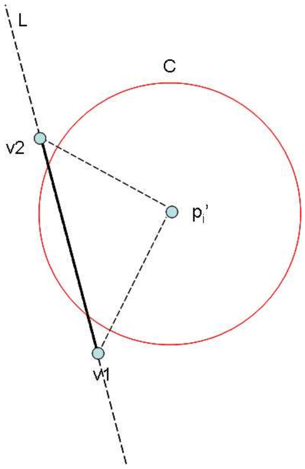

Figure 18: Intersection of C with L

B.5 How to identify the different cases?

When we calculate the Voronoi diagram, we end up with a set of edges and the point they are related to. An edge E_j is defined by its extreme points v_{j1} and v_{j2}, so the Voronoi diagram can be represented by a set of triplet $T_j = \{p_i', v_{j1}, v_{j2}\}$. These triplet define the domains (triangles) for the integral 1, described in the previous section. So, we can take these triplets, one by one, and evaluate the integral in each one and add every result to obtain the value of I.

Once given the value of U, it is necessary to identify the case for the domain defined by the triplets T. These cases are defined by the intersection between the circumference C and de line L that pass trough the edge opposed to the point p_i', (see Figure 18). The equation of C is $x^2 + y^2 = U^2$, and an equation for L in parametric form is $x = v1_x + tv_x$, $y = v1_y + tv_y$, where $v1_x$ is the component in direction x of $v1$, $v1_y$ is the component in direction y of $v1$, v is the vector that goes from $v1$ to $v2$, v_x is the component in x of v, v_y is the component in y of v, t is a parameter. We can obtain every point in L giving values to t. When $t = 0$, we obtain $v1$ and when $t = 1$, we obtain $v2$. If we use the formula of L in the formula for C, we have the following:

$$\begin{aligned} U^2 &= x^2 + y^2 \\ U^2 &= (v1_x + tv_x)^2 + (v1_x + tv_x)^2 \\ U^2 &= (v1_x)^2 + 2t(v1_x)(v_x) + t^2(v_x)^2 + (v1_y)^2 + 2t(v1_y)(v_y) + t^2(v_y)^2 \\ 0 &= (v1_x)^2 + (v1_y)^2 + t(2(v1_x)(v_x) + 2(v1_y)(v_y)) + t^2((v_x)^2 + (v_y)^2) - U^2 \\ 0 &= Co + Bot + Aot^2 \end{aligned}$$

where $Co = (v1_x)^2 + (v1_y)^2 - U^2$, $Bo = 2(v1_x)(v_x) + (v1_y)(v_y)$, $Ao = (v_x)^2 + (v_y)^2$. In order to know the intersections between the circumference C and the line L we need to solve this quadratic equation for t. Using the general formula we have two possible values for t:

$$\begin{aligned} t1 &= \frac{-Bo-\sqrt{det}}{2Ao} \\ t2 &= \frac{-Bo+\sqrt{det}}{2Ao} \end{aligned}$$

where $det = -Bo^2 - 4AoCo$. Based on the values of $t1$ and $t2$ we can know what case for the domain we need to solve as we describe next.

Case 1 This case occurs when $det < 0$, this means that C and L never intersects. This case also occurs when both $t1$ and $t2$ are less than zero or bigger than 1.

Case 2 When $t1 < 0$ and $t2 > 1$, C intersects L outside the edge opposed to p', so we have case 2.

Case 3 We have case 3 when $t1 < 0$ and $0 < t2 < 1$, or $0 < t1 < 1$ and $t2 > 1$.

Case 4 Case 4 occurs when $0 < t1, t2 < 1$.

References

[1] Yiu-Wing Leung, Yuping Wang. “U-measure: a quality measure for multiobjective programming”. IEEE Transactions on Systems, Man, and Cybernetics, Part A 33(3): 337-343 (2003).

[2] Eckart Zitzler. “Evolutionary Algorithms Multiobjective Optimization: Methods and Applications”.PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[3] David A. Veldhuizen. “Multiobjective Evolution Algorithms: Classifications, Analyses, and New Innovations”. PhD thesis, Department Electrical Computer Engineering. Graduate School Engineering. Force Institute Technology,Wright Patterson AFB, Ohio, May 1999.

[4] Knowles, J. Corne, D. “On metrics for comparing nondominated sets”, Congress on Evolutionary Computation, 2002. CEC ’02. Proceedings of the 2002.

[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga II”. In Proceedings of the Parallel Problem Solving from Nature VI Conference , 16-20 September. Paris, France, pages 849–858. 2000.

[6] Knowles, J. and Corne, D. “The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation”. Proceedings of the 1999 Congress on Evolutionary Computation, Piscatway: New Jersey: IEEE Service Center, 98105.

[7] Deb, K. “Multi-objective Optimization Using Evolutionary Algorithms”. John Wiley and Sons, Chichester, UK (2001)

[8] Michael Pilegaard Hansen and Andrzej Jaszkiewicz “Evaluating the quality of approximations to the non-dominate d set”. Technical Report IMM-REP-1998-7, Technical University of Denmark, March 1998.

[9] Deb et all. ”Scalable Test Problems for Evolutionary Multi-Objective Optimization”. TIK-Technical Report No. 112 Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich Gloriastrasse 35., ETH-Zentrum, CH-8092, Zürich, Switzerland, July 17 2001.

[10] Mehr, A.F., Azarm, S. “Minimal Sets of Quality Metrics”, Lecture Notes in Computer Science, Springer–Verlag Heidelberg, Vol 2632, pp. 405–417 (2003).

[11] Schott, J.R., “Fault tolerant design usin single and multicriteria genetic algorithm optimization”. Master Thesis, Departament of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.

[12] Zitzler, E., Thiele, L. “Multiobjective optimization using evolutionary algorithms – a comparative case study”. In Agoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans–Paul Schwefel, editors, Fifth International Conference on Parallel Problem Solving from Nature (PPSN–V), pages 292–301, Berlin, Germany, 1998. Springer.

[13] 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). LNCS 2632, Springer, Berlin (2003)529-533 (2003).

[14] Czyzak, P., Jaszkiewicz, A. “Pareto simulated annealing a metaheuristic technique for multiple–objective combinatorial optimization”. Journal of Multi–Criteria Decision Analysis, 7, 34–47 (1998).