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

Giovanni Lizárraga Lizárraga
Center of Research in Mathematics
Jalisco S/N, Valenciana
Guanajuato, México
giovanni@cimat.mx

Arturo Hernández Aguirre
Center of Research in Mathematics
Jalisco S/N, Valenciana
Guanajuato, México
artha@cimat.mx

adsafsd Salvador Botello Rionda
Center of Research in Mathematics
Jalisco S/N, Valenciana
Guanajuato, México
botello@cimat.mx

ABSTRACT

An open problem in multiobjective 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 non–dominated sets, that ranks a set of NSs based on their convergence and dispersion. 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.

Categories and Subject Descriptors

I.2.m [**Artificial Inteligence**]: Miscellaneous

General Terms

Measurement

Keywords

Multiobjective optimization, performance measures, Pareto optimality

1. INTRODUCTION

Multiobjective optimization (MOO) consist on maximizing or minimizing (or a mixture) a vector of objective functions $F(x) = \{f_1(x), f_2(x), \ldots, f_d(x)\}$ subject to constraints. The objective functions and constraints depend on a vector of variables $x \in \mathbf{R}^n$. The set Ω is defined as all vectors x that do not violate the constraints. Without loss of generality, we consider hereafter that we are minimizing the objective functions.

A popular way to deal with multiobjective problems is to use the Pareto Optimality Criteria (POC). POC is defined through the relation between two vectors $x, y \in \mathbf{R}^n$ known as Pareto dominance, or dominance. We have that x dominates y ($x \succeq y$) if $\forall i \in \{1, 2, \ldots, d\},\ f_i(x) \leq f_i(y) \ \wedge\ \exists j \in \{1, 2, \ldots, d\} \mid f_j(x) < f_j(y)$. The goal of multiobjective optimization is to find a set of vectors known as the Pareto Set (PS) defined as $PS = \{x \in \Omega \mid \forall y \in \Omega, y \not\succeq x\}$. According to POC, all elements of PS are optimal, because they represent the different tradeoffs between the objective functions where it is not possible to improve one objective without degrading another. The projection of PS in objective functions space is called the Pareto Front (PF), and is usually described as a surface that represents the best tradeoff possible between the objective functions.

In recent years, many evolutionary algorithms based on POC have been developed [4] [9] [3]. Instead of generating a single solution, these algorithms generate a set X of vector solutions x that approximate the PS. These approximation sets have the characteristic that $\forall x, y \in X, x \not\succeq y \ \wedge\ y \not\succeq x$ and are usually called non–dominated sets (NS).

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, metric or comparison method) to evaluate a NS. In order to create such a criteria we need to decide what we want from a NS. Hereafter in the article we locate points, sets, vectors, and solutions in the space of the objective functions.

We focus on these two characteristics in order to decide how good is a NS:

Convergence: it refers to how close is a NS to PF. It comes directly from the definition of the Pareto's optimality. It is usual to use PF to evaluate this characteristic. When PF is unknown it is used the best known Pareto Front.

Dispersion–Extension (DE): we are interested in the different tradeoffs of 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 and extended as possible. In order to avoid zones of the Pareto Front with too many solution and zones with too few.

Convergence is considered the most important property, because it is related with the optimality of the points in the

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-130-9/08/07 ...$5.00.

NS. A lot of the research in performance measures for NSs is focused on convergence. A classical work is that of Hansen and Jaszkiewicz [7], where they define the three following relationships between two NSs A and B.

Weak outperformance: A weakly outperforms B ($A\ O_W\ B$), if for every point $b \in B$ there exists a point $a \in A$ so that $a \succeq b$ or $a = b$ and there exists at least a point $c \in A$ so that $c \notin B$.

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

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

It is clear that $O_C \subset O_S \subset O_W$. These outperformance relations are used 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 than B, because for every vector in B there is a better one in A. So, if we have a comparison method R, and it evaluates B as better than A, then R is not reliable. We expect the same with respect to O_S, but in this case we have a weaker criteria because $A\ O_S\ B$ implies that not all vectors in B are dominated by vectors in A and some of the vectors in B are also in A. The weakest criteria comes from O_W, because if $A\ O_W\ B$, no vector in B is dominated by vectors in A, and all vectors in B are in A. With all this in mind, Hansen and Jaszkiewicz [7] 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 $A\ O\ B$ implies that R will evaluate A as (not worse) better than B.

The compatibility with the outperformance relations is desirable because it makes a comparison method more robust to misleading cases, and with a behavior according to intuition.

In this work we propose a performance measure that is weakly compatible with the outperformance 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 non–dominated sets as argument, and evaluates them to decide which one is better.

The organization of the rest of the paper is as follows. In Section 2 we review some of the performance measures available in the 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 have been proposed in the past years in order to evaluate the quality of NSs. For example the Error Ratio [11], Generational Distance [11], Schott's Spacing Metric [10], U–Measure [12], C–Metric [14], $D1_R$ and *R–Metrics* [2], and others. Different measures have different characteristics and some authors [8] [15] have analyzed their performance.

One of the most popular metrics available in the literature is the S–metric (S). Proposed by [13], the goal of this metric is to calculate the size of the space enclosed by a non–dominated set 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 set as an argument and assigns a real value to it. This value is used to compare the set with others. The election of the reference point is vital for the good behavior of the S–metric because 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 $O(Nlog(N))$ for 2 and 3 objective functions [6] and $O(N^{d/2})$ for more objective functions [1], where N is the size of the NS. The S–metric has many advantages. It is compatible with all the outperformance relations, it is scale independent and has an intuitive meaning.

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 the PF 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–metrics, 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 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-METRIC

In this paper we introduce the G–Metric (G), a m–ary performance measure. This method takes m NSs as argument and assigns a real number to each of them based on its convergence and DE. G is weakly compatible with all outperformance relations (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 one.

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–Metric algorithm and its parts.

3.1 General Algorithm

The main idea of the G–Metric is to create a partial order between the NSs we are comparing, based on the complete outperformance, (like the first criterion in the NSGA–II [4]). Then, it evaluates the non–comparable NSs based on their DE.

The convergence of the NSs is evaluated grouping them by levels of complete outperformance. The details of how this grouping is done 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 level 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. Based on DE, this component assigns a real number (I, or I_A) to each NSs A. The bigger this number, the better DE of the NS. We describe this component later.

After all this, we have that every NS has two values associated, its level of complete outperformance and its value of I. The next step is to combine this information in a single number, so we can order them from the best to the worst.

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, combine its convergence and DE to create a number that represents its relative performance respect to the other NSs (see Section 3.5).

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 compared to non–dominated ones, 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_j and min_j as the max and min value respectively, for the component j for all points $p \in C^*$.
4. Using max_j and min_j make a linear normalization of all points in all A_i.

The computational complexity for this normalization is $O(|\bigcup_{i=1}^{m} A_i|^2)$, 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.

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

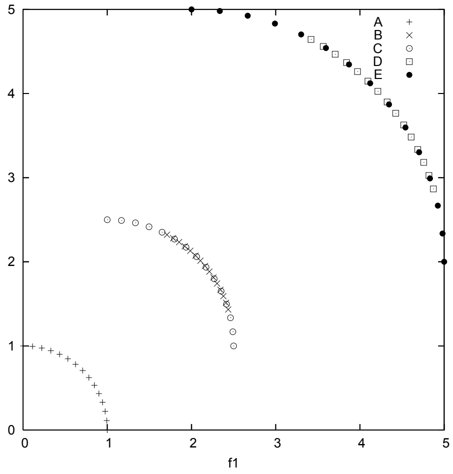

Figure 1: Five NSs, three levels of dominance

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.

Our convergence operator classifies the non–dominated sets 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 A_k so that $\neg((\bigcup_{i=1}^{m} A_i)\ O_C\ A_k)$. In other words, in L_1 are those non–dominated sets that are not complete outperformed by the union of the rest of the NSs. The following levels include those sets that are completely outperformed only by the union of sets in previous levels.

The idea is to put in the first level those NSs that contribute to the known Pareto Front with at least one vector. As the other NSs have all its elements dominated by the known Pareto Front, we consider them less important because all their information is suboptimal. Then, we repeat the process considering only the NSs not included in previous levels of complete outperformance. The algorithm is as follows:

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

1. Set $j = 1$.
2. Set $L_j = \{\}$
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 $O(m^3|A_{max}|^2)$.

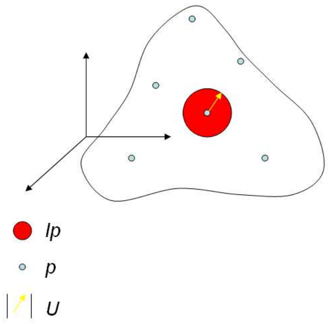

Figure 2: An example of *Ip*

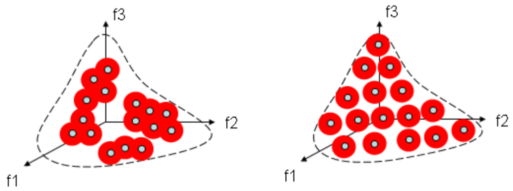

Figure 3: Size of *I* according to DE

3.4 Dispersion–Extension Component

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

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 sphere 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. We can consider U as a radius.

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 (for a point or for a set), we refer to a zone of influence as I.

Measure of I ($\mu(I)$). It is the measure of the zone of influence of a point or NS. In 2d it means area, in 3d it means volume, etc.

If the elements of S have a poor DE, many of them will be near each other and the corresponding I's will intersect (see Figure 3, left). As a result, $\mu(I_S)$ 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 distances (see Figure 3, right). As we improve the DE, the intersections of the I_{p_i}'s will decrease and, at the same time, both the DE and $\mu(I_S)$ will increase. So we have that $\mu(I_S)$ is proportional to DE and $\mu(I_S)$ is a good DE indicator.

I_S is inversely proportional to the overlap between the I_{p_i}. NSs with a good DE have less overlap than those with a bad one. The explanation on how we calculate $\mu(I_S)$ and the radius U is given in Section 3.6, where it is also shown that the computational complexity is $O(m|A_{max}|^2)$.

3.5 Computing the G–Metric

Once we have explained all the components of the G–Metric, 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 L_k.
3. For $k = 1 : Q$, where Q is the number of levels.
   (a) For every $A_i \in L_k$ eliminate all points $p \in A_i$ dominated by another point $q \in A_j$ for any $A_j \in L_k$.
   (b) Calculate the radius U based on all $A_i \in L_k$ (Section 3.6.3).
   (c) Calculate $\mu(I_{A_i})$ for each $A_i \in L_k$ (Section 3.6.2).
4. For $k = 1 : Q - 1$.
   For all $A_i \in L_k$ the value of the G–Metric is:

$$G(A_i) = \mu(I_{A_i}) + \sum_{j=k+1}^{Q} \mu_{max}(L_j) \ . \qquad (1)$$

   where $\mu_{max}(L_j)$ is the max value of $\mu(I_{A_i})$ for $A_i \in L_j$.

Note that in the calculation I (Step 3), 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 step 4 we combine the convergence with the DE of the NSs, using the levels of complete outperformance and $\mu(I)$. The procedure is designed to assign a better value of G to NSs in a better level. And, for the NSs in the same level, to assign a better values of G to those with better DE. As an example of how it works, we go back to the example in Figure 1, where $L_1 = \{A\}$, $L_2 = \{B, C\}$ and $L_3 = \{D, E\}$. Suppose that $\mu(I_A) = 0.142$, $\mu(I_B) = 0.145$, $\mu(I_C) = 0.187$, $\mu(I_D) = 0.583$ and $\mu(I_E) = 0.750$. Because of the levels they belong and their value of $\mu(I)$, the order from best to worst is A, C, B, E, D. We make $G(A) = \mu(I_A) + \sum_{j=2}^{3} \mu_{max}(L_j) = \mu(I_A) + \mu(I_C) + \mu(I_E) = 0.142 + 0.187 + 0.750 = 1.079$, $G(B) = \mu(I_B) + \sum_{j=3}^{3} \mu_{max}(L_j) = \mu(I_B) + \mu(I_E) = 0.145 + 0.750 = 0.895$, $G(C) = \mu(I_C) + \mu(I_E) = 0.937$, $G(D) = \mu(I_D) = 0.583$ and $G(D) = 0.75$. So $G(A) > G(C) > G(B) > G(E) > G(D)$ as desired.

The computational complexity of the G–Metric is equal to the sum of that of its parts (normalization, convergence component and DE component). So, it is $O(m^2|A_{max}|^2 + m^3|A_{max}|^2 + m|A_{max}|^2) \approx O(m^3|A_{max}|^2)$.

3.6 Further Details About Computing the G–Metric

Calculating I_S is complex, especially for sets 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. The projection we use is described in the next subsection.

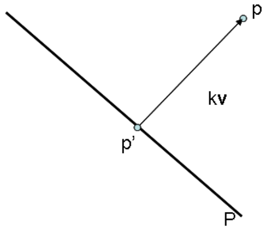

Figure 4: An example of Standard Projection in 2d

3.6.1 *Standard Projection (*SP*)*

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

An example in $\mathbf{R}^2$ is shown in Figure 4. The formula to calculate $p' = SP(p)$ is $p' = p - K\mathbf{v}$, where $K = \frac{(p \cdot \mathbf{v})}{|\mathbf{v}|^2}$.

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, etc.), the union of the $I_{p'}$ for the projected points p' is easier to compute. We use $\mu(I)$ of the projected points as an approximation of that of the original points.

3.6.2 *Computing the Zone of Influence* I_S

After a SP, all projected points p' are in the subspace P (in 2d P is a line, in 3d it is a plane, etc.). To calculate $\mu(I_S)$, we define domains of integration in P for all projected points and calculate the sub–$\mu(I_{p'})$ as an integral in the corresponding domain. Finally, we sum the values of all these integrals to obtain the total value of I_S. The general algorithm, where we represent the Euclidean distance between two points, a and b, as $d(a,b)$, is the following:

Given a set of points S and a radius U.

1. For each $p_i \in S$ calculate its projection p'_i.
2. For each p'_i calculate:

$$\text{sub–}\mu(I_{p'_i}) = 2\int_{Q_i} \sqrt{U^2 - d(u, p'_i)^2}\; dQ_i \qquad (2)$$

   where Q_i is the set of points $q \in P$ so that $d(p'_i, q) \leq U$ and $d(p'_i, q) \leq d(p'_j, q)$ for $j \neq i$.

3. Finally, calculate $\mu(I_S)$ as:

$$\mu(I_S) = \sum_{i=1}^{|S|} \text{sub–}\mu(I_{p'_i}) \qquad (3)$$

In Figure 5 we see three projected points in 2d, were P has been rotated to make it horizontal, so we can consider p_i as points in 1d. $Q_1 = [a,b]$, $Q_2 = [c,d]$ and $Q_3 = [d,e]$. So, we have that $\mu(I_S) = \int_a^b \sqrt{U^2 - (p'_1 - x)^2}\; dx + \int_c^d \sqrt{U^2 - (p'_2 - x)^2}\; dx + \int_d^e \sqrt{U^2 - (p'_3 - x)^2}\; dx$. 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 points and use the polygons of Voronoi to establish the domains. For more than three dimension we recommend the use of a Monte Carlo method for the calculation of $\mu(I_S)$.

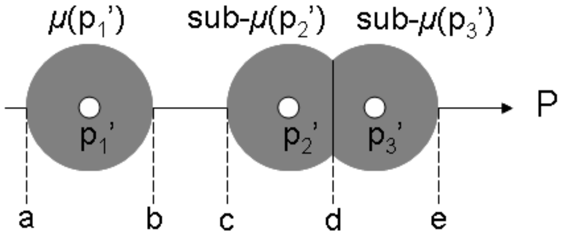

Figure 5: Domains of integration in 2d

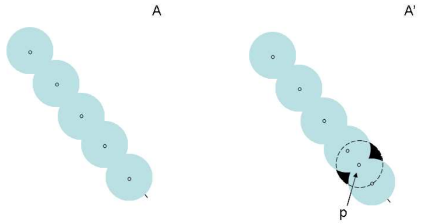

Figure 6: When a new point p is added to a NS, the I of the former increases

It is important to note that this approximation of I implies that every different point adds at least a small amount to the size of I, as seen in Figure 6, where the A' is equal to $A \cup p$. $\mu(I_{A'}) > \mu(I_A)$ because of the contribution of p. Besides, the standard projection guaranties that every different point have a different projection (see Section 3.7), so $\mu(I_A)$ increases its values as we add more points to A.

3.6.3 *The radius 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:

$$U = 0.5\frac{\sum_{j=1}^{m}\sum_{i=1}^{|S_j|} r_{ij}}{\sum_{j=1}^{m}|S_j|} \qquad (4)$$

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 [12].

3.6.4 *Computational Complexity*

The computational complexity of this component depends on that 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 2 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 $O(m|A_{max}|^2)$.

3.7 Compatibility with Ow, Os and Oc

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

First we prove that the standard projection(Section 3.6.1) is an injection when its domain is a non–dominated set.

LEMMA 1. *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 elements of $\mathbf{v}$ are equal to one), so $b_i < a_i$, but this implies that $b \succeq a$, a contradiction to the premise 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 $a \succeq b$, a contradiction to the premise A is a NS. If $K = 0$ then we have $a = b$, a contradiction to the premise $a \neq b$. As a result $a' \neq b'$.

□

The importance of this theorem is that every different point $p \in A$ will apport at least a small quantity to the value of I_A. This is because its projection p' will be different to those of other points, and, as discussed in Section 3.6.2, 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 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 $T = \{6, 7\}$.

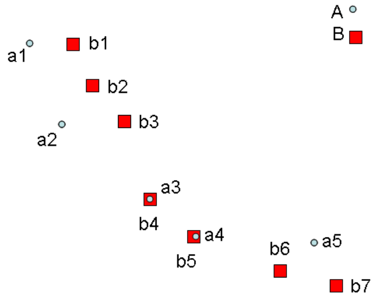

Figure 7: Two non-dominated sets.

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, $I_A = (\bigcup I_{a_j}) \cup (\bigcup I_{a_l})$. Similarly, $I_B = (\bigcup I_{a_r}) \cup (\bigcup I_{a_t})$.

The sets K and S are not used, because they contain the index of dominated elements. Now we demonstrate the following themorem.

THEOREM 1. *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:

$$\begin{aligned} \mu((\bigcup I_{a_j}) \cup (\bigcup I_{a_l})) &> \mu(\bigcup I_{b_r}) \\ \mu(I_A) &> \mu(I_B) \end{aligned}$$

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, the G–Metric 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 $\mu(I_A) > \mu(I_B)$. But when we consider C, it is possible that all a_l are dominated by some elements in C, so

Table 1: Results of Experiment 1

	2d		3d	
Set	G	S	G	S
A	1.079	0.967	1.680	0.995
B	0.895	0.452	0.615	0.368
C	0.937	0.566	0.643	0.496
D	0.583	0.065	0.347	0.059
E	0.750	0.067	0.572	0.094

the elements in L are not considered and $\mu(I_A) = \mu(I_B)$. 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 $\mu(I_A)$ will not be smallest than $\mu(I_B)$, 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 experiments that consider various topologies of the Pareto Front. We compare our results with those of the S-metric, described in section 2. We chose the S–metric because it stands out as one of the most popular in literature. We used as r^* the max value of all sets in all dimensions.

We designed our test cases so it is evident which NS is better than the others, so it is possible to decide the right order from the best to the worst. The challenge for the performance measures is to evaluate the NSs in such a way that we can construct the right order.

In the Tables, we show the numeric value of G-Metric and S–metric in the column G and S respectively. We made a 2d and a 3d version of each experiment.

Experiment 1. 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 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 1.

Experiment 2. In this problem the Pareto Front consists of all the points p with components $p_i \geq 0$ and $\sum_{i=1}^{d} p_i = 0.5$ [5]. 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 8 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, E.

The result of the experiment is shown in Table 2. Again, G and S passed the test.

Experiment 3. The goal of this experiment is to evaluate the sensibility of the measures to the convexity of the Pareto Front. Both A and B (Figure 9) have the same DE and convergence, but A is on a non-convex zone while B is on a convex zone. We expect the same value for both NSs.

It is clear, according to Table 3, that S-metric has a bias towards the convex zones of the Pareto's Front thus it failed

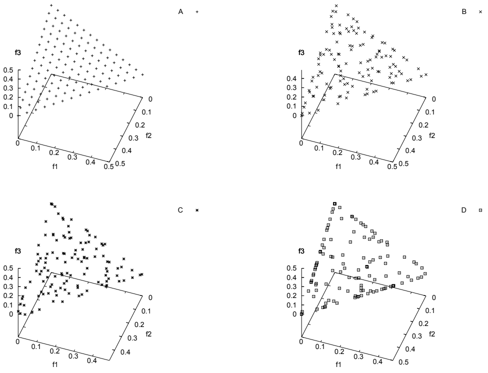

Figure 8: Experiment 2

Table 2: Results of Experiment 2

	2d		3d	
Set	G	S	G	S
A	0.107	0.464	0.0212	0.795
B	0.096	0.460	0.0177	0.785
C	0.068	0.421	0.0172	0.784
D	0.064	0.322	0.0134	0.780
E	0.035	0.262	0.0110	0.769

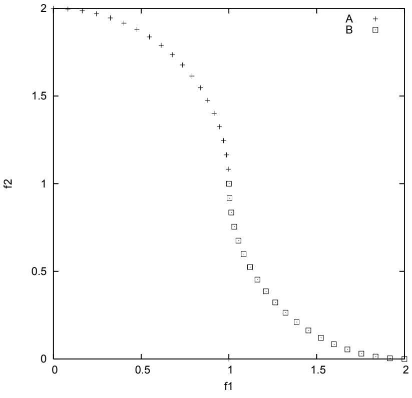

Figure 9: Experiment 3

Table 3: Results of Experiment 3

	2d		3d	
Set	G	S	G	S
A	0.0256	0.298	0.0111	0.174
B	0.0256	0.440	0.0111	0.364

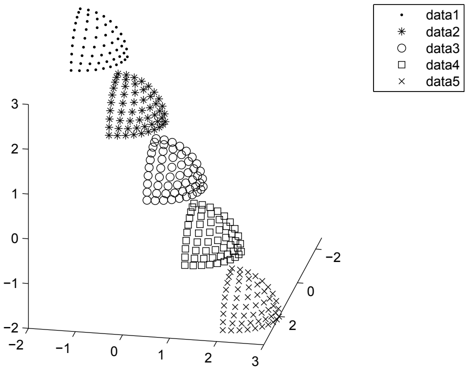

Figure 10: Experiment 4

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.

Experiment 4. 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 3d version is shown in Figure 10.

As it is clear from Table 4, S gave different values to different NS, so it failed the test. G gave the same value to all NSs, so it was the only method that passed the test.

5. CONCLUSIONS

We presented the G–Metric, an 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 another performance measure very popular in the literature.

Table 4: Results of Experiment 4

	2d		3d	
Set	G	S	G	S
A	0.0041	0.167	7.08e-5	0.195
B	0.0041	0.287	7.08e-5	0.251
C	0.0041	0.327	7.08e-5	0.211
D	0.0041	0.287	7.08e-5	0.123
E	0.0041	0.167	7.08e-5	0.035

6. REFERENCES

[1] N. Beume and G. Rudolph. Faster s-metric calculation by considering dominated hypervolume as klee's measure problem. In B. Kovalerchuk, editor, *Computational Intelligence*, pages 233–238. IASTED/ACTA Press, 2006.

[2] P. Czyzak and A. Jaszkiewicz. Pareto simulated annealing a metaheuristic technique for multiple- objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, 1998.

[3] K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Chichester, UK, 2001.

[4] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algo- rithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Zurich, Switzerland, 2001.

[6] C. M. Fonseca, L. Paquete, and M. López-Ibáñez. An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006)*, pages 1157–1163, 2006.

[7] M. P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, 1998.

[8] J. Knowles and D. Corne. On Metrics for Comparing Nondominated Sets. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716.

[9] J. Knowles and D. Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 98–105, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.

[10] J. Schott. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. PhD thesis, Massachusetts Institute of Technology, May 1995.

[11] D. A. Veldhuizen. *Multiobjective Evolution Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Force Institute Technology, Wright Patterson AFB, May 1999.

[12] L. Yiu-Wing and W. Yupinge. U-measure: a quality measure for multiobjective programming. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 33(3):337–343, May 2003.

[13] E. Zitzler. *Evolutionary Algorithms Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), November 1999.

[14] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 292–304, London, UK, 1998. Springer-Verlag.

[15] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):529–533, May 2003.