Aggregation Trees for visualization and dimension reduction in many-objective optimization

Alan R.R. de Freitas^{a,*}, Peter J. Fleming^b, Frederico G. Guimarães^c

^a Department of Computer Science, Federal University of Ouro Preto, Ouro Preto 35400-000, MG, Brazil

^b Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 4DU, UK

^c Department of Electrical Engineering, Federal University of Minas Gerais, Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG, Brazil

ARTICLE INFO

Article history:

Received 10 March 2014

Received in revised form 25 November 2014

Accepted 28 November 2014

Available online 5 December 2014

Keywords:

Aggregation Trees

Multi-objective optimization

Many-objective optimization

Visualization

Evolutionary computation

Polar graphs

ABSTRACT

This paper introduces the concept of Aggregation Trees for the visualization of the results of high-dimensional multi-objective optimization problems, or many-objective problems and as a means of performing dimension reduction. The high dimensionality of many-objective optimization makes it difficult to represent the relationship between objectives and solutions in such problems and most approaches in the literature are based on the representation of solutions in lower dimensions. The method of Aggregation Trees proposed here is based on an iterative aggregation of objectives that are represented in a tree. The location of conflict is also calculated and represented on the tree. Thus, the tree can represent which objectives and groups of objectives are the most harmonic, what sort of conflict is present between groups of objectives, and which aggregations would be helpful in order to reduce the problem dimension.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Real-world optimization problems often have many conflicting objectives and, as a consequence, multi-objective optimization problems have become a major field of research. Sometimes these problems can be solved by aggregating all of the objectives into a single objective function that gives some weight to each of the objectives. However, very often, it is more important to understand the relationship between the objectives in order to make decisions with a reasonable understanding of the trade-off involved among the possible choices.

When there are few objectives to be considered, evolutionary algorithms are usually suitable to find the best combination of solutions to a problem and it is not very difficult to represent that set of possibilities (Section 2). As the number of objectives grow, we reach the field of many-objective problems (Section 2.3). The optimization of these problems becomes more challenging as most solutions become incomparable as the Pareto dominance selection becomes less discriminating. That means that any solution in our set of candidates is almost always better than all other solutions for at least one specific combination of objective weights. Besides, representing quality for a set of solutions in this context is equally a problem.

In order to address this problem, we propose the concept of Aggregation Trees (Section 3). Aggregation Trees iteratively group objective functions in tree nodes according to their harmony (a concept mathematically defined in Section 4), similar

* Corresponding author.

E-mail addresses: alandefreitas@iceb.ufop.br (A.R.R. de Freitas), p.fleming@sheffield.ac.uk (P.J. Fleming), fredericoguimaraes@ufmg.br (F.G. Guimarães).

URL: <http://www.alandefreitas.com> (A.R.R. de Freitas).

to an agglomerative hierarchical clustering method. Thus, it represents the relationship between conflicting groups of objectives for a problem. It also represents the globality or locality of the conflict, the region of concentration of the calculated conflict (as mathematically defined in Section 5) through branch colors and it shows which is the best combination of objectives to be aggregated for dimension reduction.

The contributions of the proposed approach are:

- Visualization of results for many-objective problems including particularly:
 - The amount of harmony and conflict between relevant groups of objectives.
 - The globality or locality of the conflict.
 - The region where the conflict is more intense in the objective space, when conflict is not equally spread in the objective space.
- Representation of the most useful aggregations in order to iteratively perform dimensionality reduction on any number of objectives of the original problem.
- Robustness in relation to problems where the relationship between objectives is not linear. This is obtained by employing a non-parametric analysis.
- The position of nodes in the tree suggest convenient positions for the representation of absolute objective values in parallel coordinate plots.
- Polynomial computational cost and ease of implementation.

Thus, as the majority of solutions in many-objective problems tend to be non-dominated, Aggregation Trees can be used as a method for providing the decision maker with enough information to restrict the preference area for a problem, restrict the domain of search variables, create new constraints for the problem or reduce the number of objectives in a further analysis.

We describe the algorithm in Section 6, analyze the polynomial time complexity of the suggested method in Section 6.1, and apply it to test instances in Section 7. We then conclude our paper with discussions about the method and suggestions concerning possible future work (Section 8).

2. Historical perspective

Complex problems often require the consideration of many criteria of performance, and, as a consequence, multi-objective problems are a well studied topic in the scientific literature. In simple approaches, objectives are aggregated into a single objective function that takes into account the utility of each objective. However, in a multi-objective problem, the solution set can be found to represent the trade-off between those objectives [12,10,21]. The solutions that represent this compromise are in the Pareto-optimal set, which has only non-comparable optimal solutions in the objective space. This means that no solution in the Pareto-set is preferable to another in relation to all objectives.

2.1. Multi-objective optimization problems

Definition 2.1. A multi-objective optimization problem can be defined as:

$$\min(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \mathbf{x} \in \mathcal{F} \quad (1)$$

where $\mathbf{x}$ is a solution for the problem and $f_i(\mathbf{x})$ is the i -th objective function to be minimized.

Each function $f_i(\mathbf{x})$ maps the optimization variables of a candidate solution $\mathbf{x}$ to an objective value represented in one dimension of the objective space, i.e., $f_i(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$.

Definition 2.2. The set of all combinations of possible values for optimization variables defines the search space $\mathcal{S}$:

$$\mathcal{S} = \{\mathbf{x} = \{(x_1, v_1), \dots, (x_n, v_n)\} : v_i \in \mathcal{D}_i\} \quad (2)$$

where each variable x_i assumes the value v_i in its respective domain $\mathcal{D}_i$.

Definition 2.3. If the problem has constraints, those constraint functions $g_i(\mathbf{x})$, $i = 1, \dots, r$ define the feasible set of solutions $\mathcal{F}$:

$$\mathcal{F} = \{\mathbf{x} \in \mathcal{S} : g_i(\mathbf{x}) \leq 0, i = 1, \dots, r\} \quad (3)$$

In multi-objective problems, the comparison between solutions has to consider all objective functions. We can only say $\mathbf{x}^1$ is a better solution than $\mathbf{x}^2$ if $\mathbf{x}^1$ is better on all the objective functions.

Definition 2.4. Let $f(\mathbf{x}^1) \prec f(\mathbf{x}^2)$ denote that a feasible solution $\mathbf{x}^1 \in \mathcal{F}$ dominates another solution $\mathbf{x}^2 \in \mathcal{F}$. $f(\mathbf{x}^1) \prec f(\mathbf{x}^2)$ iff the conditions below are attained:

$$f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \quad \text{for all } i \in \{1, 2, 3, \dots, m\} \quad (4)$$

$$f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2) \quad \text{for at least one } j \in \{1, 2, 3, \dots, m\} \quad (5)$$

Thus, solving the optimization problem defined in Eq. (1) involves finding the set of solutions (or a representative set of solutions) which are Pareto-optimal, that is, solutions that cannot be improved in any of the objectives without incurring an inferior result for another objective.

Definition 2.5. A feasible solution $\mathbf{x}^* \in \mathcal{F}$ is Pareto optimal iff $\nexists \mathbf{y} \in \mathcal{F}$ such that $f(\mathbf{y}) \prec f(\mathbf{x}^*)$. Solving a multi-objective optimization problem involves finding the Pareto set $\mathcal{P}$ denoted as:

$$\mathcal{P} = \{\mathbf{x}^* \in \mathcal{F} : \nexists \mathbf{y} \in \mathcal{F}, f(\mathbf{y}) \prec f(\mathbf{x}^*)\} \quad (6)$$

2.2. Evolutionary multi-objective evolutionary algorithms

Due to their flexibility and inherent set-based strategy, evolutionary algorithms have been widely used for addressing multi-objective problems. The original algorithms for multi-objective problems used Pareto-based ranking to assign fitness to different solutions [10,21]. Many other Evolutionary Algorithms have been proposed since then with good results for multi-objective problems [13,68,34]. Recent algorithms often involve decomposition of the original problem into a set of single objective subproblems [66,14,39] or make use of auxiliary indicators to evaluate solutions [3].

Besides the flexibility of evolutionary algorithms, these approaches also permit mechanisms for controlling diversity [69,37,34,29,38,26,51,5], integration of the decision-making into the process [20,45,42,64,2], and ease of creation of hybrid approaches [25].

2.3. Many-objective problems

Despite the success of those algorithms for addressing real-world problems, their performance reduces dramatically for problems that have more than three objectives. This fact led to the definition of a new term to better refer to this context, namely, many-objective problems [32]. In those problems, Pareto-based ranking is no longer an effective discriminator and the majority of solutions are non-comparable, thus compromising the convergence of the algorithms.

Besides, there is a memory cost of representing a Pareto set that grows exponentially with the number of conflicting objectives and the visualization of that Pareto set also becomes difficult in many-dimensions, which in turn become a burden for the decision-making process as well.

Khare et al. [33] showed the poor scalability of NSGA-II, PESA and SPEA2 in scalable test functions. Hughes [27] showed that aggregation methods with multi-start perform better than MOEAs. Knowles & Corne [35] showed that MOEAs do not perform better than random search in problems with more than 10 objectives. Purshouse & Fleming [48] found that the ability of variation operators to produce solutions that dominate their parents decrease with increasing numbers of objectives.

In order to represent this effect, Fig. 1 shows the percentage of non-dominated solutions in a set of objective values randomly generated with a Gaussian distribution. The x-axis is the number of objectives, while the y-axis represents the percentage of non-dominated solutions on an initial set of solutions for a problem of that dimension.

Fig. 1. Number of non-dominated solutions in a randomly generated set with a Gaussian distribution.

It is easy to verify that the more objectives we have, the higher the percentage of initial solutions that are already non-dominated. When we use a smaller set of solutions, this percentage increase is even faster. Each combination of a number of objectives and set size was tested 100 times and the dotted lines show the 25th and 75th percentiles. These percentiles are always very close to the average values. Usually, if those initial solutions are used in an elitist evolutionary algorithm, all results become non-dominated after a very few generations. Observations like this have been made elsewhere in the literature, see for instance [4].

There is still another way to intuitively see how the number of solutions needed to represent the Pareto-set grows exponentially. For a single-objective problem, the solution is represented by a single point. For 2, 3, 4, ... objectives, the Pareto set may be represented by a curve, a surface, a hyper-surface, and so on. Thus, even if we keep only the extreme solutions for each dimension (which are not generally the most appropriate ones) we would need at least 2^{M-1} solutions to represent the trade-offs.

This exponential relationship shows the importance of treating many-objective optimization as a distinct class of problems. Since there are mainly non-dominated solutions in sets of any practical size, understanding the relationship between the objectives and providing the most useful information to decision makers in polynomial time is more fruitful and representative than the non-polynomial task, in space and time, of finding a front with the best solutions.

Alternatives for solving many-objective problems often introduce certain preferences or measures of the decision maker into the optimization process: different dominance relations to improve discrimination at each generation [51,55,5], modifying ranking methods [17,70,52,62,35], other fitness measures involving indicators [67,6,8,9,2,53], or the simultaneous optimization of many scalarized objective functions [28,30,31,63].

2.4. Objective reduction

Another approach to address these problems is dimensionality reduction through identification of redundant objectives [8,9,53,57,22].

Brockhoff and Zitzler [8,9] propose a method of objective reduction geared towards direct integration into the evolutionary search. Their algorithm depends on a change in the dominance structure. They also present the problem of finding a minimum objective subset, maintaining the given dominance structure with a given error. A greedy algorithm removes objectives if the dominance relations do not change with the removal of the objective because it is perceived as a non-conflicting one.

Saxena et al. [53] employ Principal Component Analysis (PCA) of the objective values in order to identify related objectives. The PCA is tailored to handle non-linear relations between the objectives and the technique is then embedded into NSGA-II. Other papers have also presented new techniques based on maximum variance unfolding and PCA for nonlinear reduction of objectives [54]. Those techniques remove dependencies in the non-dominated solutions.

Jaimes et al. [40] employ unsupervised feature selection to reduce the number of objectives according to their correlation. Distant objectives are considered as conflicting. In a similar approach to that of Brockhoff and Zitzler [9], the algorithm also seeks the minimum objective subset with minimum error.

Singh et al. [56] present a Pareto Corner Search. This algorithm focuses only on corners among the non-dominated solutions. After that, the number of resulting non-dominated solutions from a reduction of the achieved solutions indicates whether the objectives are essential or whether they can be reduced.

Freitas et al. [22] present a mathematical formulation of the concept of harmony previously defined by other articles [47,48,9,24]. The more harmonic two objectives are, the more it indicates that they can be aggregated into a compound objective function without much loss in the representation of the Pareto-set.

2.5. Visualization of results

The results of multi-objective problems in 2 dimensions can be easily represented in a simple two-axis Cartesian representation as in Fig. 2(a). In that representation, each solution is displayed as a point. The non-dominated solutions are identified by a number 1 in the first front. Those are the non-comparable, or non-dominated, solutions. All other solutions are worse than at least one solution in the first front, assuming minimization of the objectives. Usually, dominated solutions are not presented in the graph when we want to make the information clearer.

By means of a simple extension, Fig. 2(b) can be used for problems with 3 objectives. In that case, the shape of the non-dominated solutions also has an additional dimension. One can see the reduction in the number of fronts, as also shown in [4].

For visualizing non-dominated results in many-objective optimization, a common approach is to use parallel coordinate plots, as it is shown in Fig. 3 for 7 objectives. In this plot, each connected line represents a solution and the objective values are normalized. When there are many lines crossing between 2 adjacent objectives, this indicates that there is conflict between those objectives. In this Figure, also, different colors are assigned to the lines to make visualization easier and dominated solutions are represented with a thin black dotted line, which is difficult to see because there are only 3 dominated solutions in the whole set.

Fig. 2. Representation of the quality of many solutions for MOP.

Fig. 3. Representing the quality of solutions for many-objective problems.

A problem with the representation method used in Fig. 3 is that we can only see conflict between adjacent objectives. In order to better represent solutions in high dimensions, other approaches often involve mapping the objectives onto lower dimensions. However, it is not possible to perform any mapping to a lower dimensional space without losing information and general existing methods for such mapping do not tend to take care to preserve the underlying Pareto-dominance relationship between solutions in a higher dimension.

Eddy and Lewis propose cloud visualization [18], in which the designer can view design information as well as the objective space simultaneously.

Obayashi and Sasaki [46] use self-organizing maps to mine the solutions and find trade-offs in the Pareto solutions of 4 dimensions. Furthermore, the authors create a second self-organizing map of cluster-averaged design variables to analyze their roles on the quality of the solutions.

Mattson and Messac [44] used an s-Pareto frontier as the result of a filter applied to the original non-dominated objective values. However, this approach leads to a loss of dimension representation.

Lotov [41] uses a technique of Interactive Decision Maps to represent solutions in the objective space as an alternative to parallel coordinate plots for problems having between three and seven objectives.

Agrawal et al. [1] use a Hyper-space Diagonal Counting method as a visualization tool for multi-objective solutions.

Köppen and Yoshida [36] use an approach that aims for the preservation of Pareto-dominance relations in lower dimensions. Their algorithm searches for a mapping that has few wrong dominance relations and preserves nearest neighbor relations. Thus, mapping itself is a multi-objective problem which is handled by an evolutionary algorithm.

Blasco et al. [7] employ level diagrams for n -dimensional Pareto front analysis. In this technique, each objective is represented on a separate diagram. The method utilizes an ideal point to classify the points.

Mascia and Brunato [43] analyze techniques for visualization of search landscapes in combinatorial optimization. They propose techniques that explicitly render geographic metaphors used by researchers. Adaptations on those algorithms could make them useful for visualizing objective spaces.

Tušar and Filipič [58] describe a methodology for presenting the quality of multi-objective solutions with 4 dimensions. In their method projections (projections of a section [23]) are used to perform dimension reduction.

Yoshimi et al. [65] present a spherical self-organized map to visualize solutions with many objectives. The spherical approach addresses the issue that similar data differing in few variables are often placed at opposite ends of the map.

Walker et al. [59] conducted an extensive review of methods for Pareto-visualization and presented their own methods. They use spectral seriation to rearrange the solutions and objectives plotted on a heatmap. They also present two methods to visualize solutions in a plane: one that maps a set to the interior of a polygon on a plane, and another that uses a measure of dominance distance between solutions to yield visualizations in two dimensions.

Fieldsend and Everson [19] propose a method to visualize Pareto relationships in two-dimensional scatterplots. They attempt to create a two-dimensional projection with minimal loss of dominance information. In their plot, points represent solutions and connecting lines represent dominance relations.

In a multi-objective problem, there are many features the decision maker will want to see from the data returned by any optimization algorithm. Among those features, it is important to have a visualization method in which it is possible to easily perceive components such as:

1. The amount of conflict between the objectives.
2. The amount of harmony between the objectives.
3. The nature of those features in the set of points.
4. The region where those features appear, if they are local.

This last feature is important for the decision maker because if there is only conflict for solutions which are poor in regard to two given objectives, the decision maker would probably take decisions very different from a case in which the conflict happens only among good solutions.

Besides that, in many-objective optimization, we have interest in also visualizing:

1. A comparison of the conflict between the combinations of objectives.
2. Which combination and reduction of objectives into a simple summary would cause less distortion in the final representation of the Pareto front
3. The conflict we have between groups of objectives, not only pairs of individual objectives

The appropriate answer to these questions would provide useful information for the decision maker to understand the problem and then take the appropriate measures to filter those solutions and restrict the preference area until there is only a single solution. We can see that the accurate measures of those features would involve (i) the robust calculation of conflict and harmony between objectives that are often in different units, (ii) the definition of the cause of the final values to define locality and globality, and (iii) a method of visualization that could contain all this information in a way that is easy to interpret.

3. The concept of an Aggregation Tree

In this paper, we introduce a form of representing the relationship between objectives in a many-objective problem. An aggregation tree is based on the idea that the more harmonious two objectives are, the better suited they are to be reduced or grouped into a single new scalarized objective value [47,48]. With the help of a formalized non-parametric mathematical definition of harmony [22], a tree can show the relationships between conflicting objectives.

In order to exemplify the utility of the proposed Aggregation Trees, consider the set of solutions represented in the parallel coordinate plot of Fig. 4(a). Every adjacent pair of objectives in this set of 40 solutions contains a distinctive harmony and conflict relationship. Objectives 1, 2, 4, 6, 8, 10, and 12 are in complete harmony because they have the same objective values. A mathematical formulation of conflict and harmony is given in Section 4.

Fig. 4(b) represents the relationship between those objectives and their reducibility in the structure of a tree. The objectives are iteratively clustered by their amount of harmony while percentages on parent nodes show the conflict between the children objectives. Any parent node represents the aggregation of the children objectives into a new objective. More specifically, this Aggregation Tree has the following properties:

- Leaf nodes represent objectives in the form f_i , where i is the objective number.
- Parents of leaf nodes represent a compound objective formed by the summation of objectives and renormalization of the values. It is represented in the form $f_{i1} + f_{i2} - c\%$, where $i1$ and $i2$ are the objectives used and c is the conflict between

Fig. 4. A set of solutions and its Aggregation Tree.

those objectives, given the measure to be explained in Section 4. For example, the node $f_{10} + f_8 - 0\%$ represents a compound objective with objectives 10 and 8. It also shows that there is 0% of conflict between those two objectives, so that they can be grouped because they are in complete harmony.

- Other parent nodes represent higher order compound objectives formed in a similar way. The values of conflict in those parents include only the conflict between their direct children's compound objectives. For instance, besides representing the compound objective formed with objectives 10, 8, 6, and 4, the node $f_{10} + f_8 + f_6 + f_4 - 0\%$ also shows that there is 0% of conflict between the compound objectives $f_{10} + f_8$ and $f_6 + f_4$. The conflict 0% between even objectives occurs because the objective values of even objectives in Fig. 4(a) always return to the same values as in objective f_2 , while the odd objectives are being used to didactically employ various possible relationships between objectives.
- Objectives are respectively grouped according to their harmony and therefore distant cousins in the tree represent less harmonious objectives.
- Nodes in black represent global conflict. Values in red represent local conflict more concentrated on good values for those objectives. Values in blue represent conflict for bad solutions in the objectives. Measures of conflict concentration at a region of the objective space are given in Section 5.

When two compound objectives are grouped into one new compound objective, the conflict in the previous compound objectives is not considered in the conflict of the new objective. Thus, an optional feature (not included in this example to facilitate visualization) is to have the size of each branch of the tree proportional to its conflict, in order that we can perceive the accumulated conflict from the root node to any position of the tree.

The values of conflict in the tree can also be normalized or rescaled if high values of conflict would make cognition more difficult. This strategy has the benefit of not depending on the measure of conflict being used. However, we lose the absolute values of conflict by making use of this approach. Thus, we cannot infer the general amount of conflict between objectives a problem.

The first advantage of visualizing the relationship between objectives in an Aggregation Tree is that it is straightforward to see the conflict between the objectives and the objectives are grouped according to their harmony. Thus, in a case where the decision maker wishes to make compound objectives according to their harmony and therefore their reducibility, the nodes give information on which would be the best ones to group. Instead of only showing the conflict between every pair of objectives, the tree also shows the conflict between groups of objectives that could be aggregated instead of showing the conflict between many groups of objectives which are not related. Also, the colors represent the globality or locality, and regions of the conflict.

One more useful application of the tree comes from the idea that if two leaf nodes are distant cousins in the tree, there is little harmony between those objectives. Thus, the final position of leaf nodes gives a convenient way to choose good adjacent objectives in a parallel coordinate plot, such as the one in Fig. 5(a). In this Figure, it is easier to identify the relationship between the objectives using their absolute values because harmonious objectives are nearby, not disrupting the representation of conflicting objectives as occurs for each pair of objectives in Fig. 4(a).

Given the usefulness of this ordering of the objectives and the fact that we might also want to visualize the two most distant cousins on the tree, we propose the use of an improved representation of the absolute values in a Polar Graph, such as the one in Fig. 5(b). Similarly to parallel coordinate plots, each line in the Polar Graph represents a solution and the main difference between them is that the Polar Graph has a toroidal structure that allows the representation of extreme objectives in the parallel coordinate plot, which, according to the tree, must be of interest because the nodes representing those objectives are the furthest cousins possible. Also, the interior of the circle represents inferior values (high values for minimization objectives) for each of the objectives while the most external part represents better values in that objective. The values are linearly normalized and the range of absolute values in each objective is represented outside the circle. A clustering technique, PSA [49] in this example, is also applied here to facilitate visualization of the results.

Thus, in our example, the toroidal structure of the Polar Graph is capable of showing the conflicting relationship between objectives 11 and 7. Also, as close cousins in the tree represent objectives in harmony, it is possible to visualize the effect of each jump from one branch of the tree to the other. For instance, it is easy to visualize that solutions in green, in general, have high values in the harmonious objectives $f_{10}, f_8, f_6, f_4, f_2, f_1, f_{12}, f_1$, and f_9 . We can see that those good values in the harmonious objectives are compensated with low values for f_3 and medium values for f_{11} and f_5 . Yet, low values in the harmonious objectives for solutions in red are generally compensated by high values in f_{11} and f_3 .

One last interesting use of the data provided by this analysis of conflict is the direct first-order conflict between each of the original objectives, such as the one in Table 1. The table represents the conflict between every pair of objectives, according to the metric to be presented in Section 4.

Thus, an Aggregation Tree is a tool to help decision makers compute and visualize redundancy and conflict between objectives. Its main applications are grouping objectives for dimensionality reduction and visualization of the analyses using trees and polar graphs. The MATLAB source code for all these algorithms is available from <http://www.alandefreitas.com/>.

Sections 4 and 5 describe the mathematical formulation of conflict, harmony, and conflict region. In Section 6 we describe in detail the algorithm for constructing a Aggregation Tree; its time complexity is described in Section 6.1.

4. Conflict and harmony as a form of aggregation

Fig. 6 represents many kinds of conflict and harmony in parallel coordinate plots of two objectives. We say there is conflict [47,48,9,24] between two objectives when good values for one of them imply bad values for another. On the other hand, we say there is harmony between two objectives when improvement in the first objective leads to improvement in the second. We can see from the Figure how not only the amount of conflict varies but the conflict can also be concentrated in a certain region of the objective space.

Thus, the concept of conflict may depend on what the user expects as “good” and “bad” values for each of the objectives. Our perception of conflict also depends on the shape of the Pareto front formed by the best solutions because it might be that the range of values known for each objective is very large due to outliers, thereby changing our perception of conflict.

However, the concept of harmony only implies that improvement in one objective leads to improvement in another. In this sense, harmony may not always be the exact opposite of conflict. Harmonious objectives are presented in a parallel coordinate plots by non-crossing lines. Thus, it has to do with the possibility of joining the objectives through summation without loss of quality in the Pareto front. If we want to group two objectives into a new compound objective, it is best to group those objectives with greater harmony even if there is perception of some level of conflict between them.

According to these loose definitions, we propose [22] specific mathematical formulations of conflict and harmony by specifying three general types of conflict: direct, max–min, and non-parametric conflict. Harmony, on the other hand is inversely proportional to non-parametric conflict.

Table 2 shows the mathematical formulation of those three kinds of conflict. C_{ab} represents the conflict between objectives a and b . $c_{\min}$ and $c_{\max}$ represent the minimum and maximum values possible of conflict for that measure. $\mathbf{X}_{ij}$ is the value of the solution i (of a total of n solutions) on objective j . Also, $\mathbf{R}_{ij}$ is the rank of $\mathbf{X}_{ij}$ within $\mathbf{X}_j$. If there are ties in the calculation of $\mathbf{R}_{ij}$, we break the ties by using the rank that causes least conflict with the other objective being compared.

Each formulation is built on a different view of what can be considered conflict between two objectives. These different views translate into different forms of normalizing objectives. The measure of harmony is a mathematical formulation of the idea that crossing lines in parallel coordinate plots represent a lack of harmony. The 3rd and 4th columns contain the minimum and maximum values of conflict that can be obtained in that metric. This is helpful for normalizing the results, if required.

Direct conflict considers the absolute difference between one objective and another. Thus, it works for objectives that can be converted to the same units (for example, by reducing the objective values to financial cost in a practical application). Max–min normalizes those values on a range from 0 (minimum) to 1 (maximum) before doing the comparison, therefore implying that the importance of the objectives is inversely proportional to their achievable range of values. This is a realistic scenario for our cognitive perception of the parallel coordinate plot. The maximum value (for a minimization problem) can be replaced by a value that reflects the minimum expectation of the decision maker for that objective. Finally, the non-parametric approach ranks the absolute values in each of the objectives before they are compared. This metric measures only the degree to which the lines would be crossing each other in a parallel coordinate plot. Table 3 summarizes when to apply each measure of conflict.

(a) Parallel Coordinate Plot with a Different Choice of Adjacent Objectives

(b) Polar Graph

Fig. 5. Polar graph for the set of solutions in Figs. 4 and 5(a).

A non-parametric measure of conflict works without the assumption of comparability between objectives. As this metric reflects the degree by which lines would be crossing between objectives a and b in a parallel coordinate plot, it can also be used to formulate the measure of harmony defined in Eq. (7). This measure of harmony returns values that range between 0 to 1.

$$H_{ab} = 1 - \frac{C_{ab}}{C_{\max}} \quad (7)$$

Table 1

Conflict between the objectives in the solutions of Fig. 4(a). The values range from 0 (minimum conflict) to 1 (maximum conflict).

	1	2	3	4	5	6	7	8	9	10	11	12
1		0	1	0	.25	0	.25	0	.05	0	1	0
2	0		1	0	.25	0	.25	0	.05	0	1	0
3	1	1		1	1	1	1	1	1	1	.25	1
4	0	0	1		.25	0	.25	0	.05	0	1	0
5	.25	.25	1	.25		.25	.5000	.25	.275	.25	1	.25
6	0	0	1	0	.25		.25	0	.05	0	1	0
7	.25	.25	1	.25	.5	.25		.25	.275	.25	1	.25
8	0	0	1	0	.25	0	.25		.05	0	1	0
9	.05	.05	1	.05	.275	.05	.275	.05		.05	1	.05
10	0	0	1	0	.25	0	.25	0	.05		1	0
11	1	1	.25	1	1	1	1	1	1	1		1
12	0	0	1	0	.25	0	.25	0	.05	0	1	

It is important to mention that, if we invert the solution qualities for the dimensions, the calculation of C_{ab} is equivalent to Spearman's footrule [16] and it is similar to Spearman's Rho [11] in the sense that it is a nonparametric measure of statistical dependence between two variables. As mentioned already, the more harmonious two objectives are, the better candidates they are to be grouped into a new compound objective, and this fact is conveniently used by the Aggregation Trees to represent the relationship between objectives.

5. Conflict concentration regions

Since non-parametric conflict and harmony are measured as the summation of rank differences, the location of those differences is also generally relevant information. It is important for the decision maker to know whether the conflict is occurring for the best or worst values for the objectives being compared. That information is relevant because if the conflict only occurs for solutions with poor objective values on those objectives, that means that the conflict would not exist if those were the only objectives being considered.

For instance, Fig. 6 presents many examples of trade-offs between two objectives in a parallel coordinate plot and all the results are shown in terms of ranks. In Fig. 6(a), an improvement in objective 1 always leads to an improvement in objective 2. In this case, there is no conflict between the objectives and the behavior is global. This is the best case for which to make a new compound objective. If there is little non-parametric conflict and much harmony, most of the behavior will still be global because the harmony is global. In Fig. 6(b), the behavior is also global because there is conflict everywhere. This is the extreme case of conflict, where an improvement in one objective necessarily causes deterioration in another.

Fig. 6(c) and (d), on the other hand, are examples in which the conflict is only local. In the first case of Fig. 6(c), conflict only occurs for solutions which are poor in relation to those objectives, considering a minimization problem. In those cases, the removal of other objectives would eventually cause the trade-off to disappear, as for only the objectives being analyzed, only one solution would be in the Pareto front. In Fig. 6(d), however, the elimination of other objectives could not lead to less conflict between the objectives being considered because five solutions would still be in the first front because the conflict between those objectives is inherent. That demonstrates the importance of representing not only the globality of conflict but also its region.

Having less than maximum conflict does not necessarily mean that conflict is local. The example in Fig. 6(e) shows a case of global medium conflict. Another special case is the one represented in Fig. 6(e) in which there is local conflict but the region of conflict is different for the objectives. In that case, there would be a Pareto front even if just those two objectives are taken into consideration, because the conflict is located around good solutions for objective 1. In that case, the conflict is located in the good values for objective 1 and bad values for objective 2.

Since harmony is expressed in terms of the accumulated difference of ranks in the solutions, it is then important to see where those differences come from and whether those differences are more located around good or bad solutions. Fig. 7 shows the difference of values for the same examples in Fig. 6. The x axis has the rank of each solution in objective 1 while the y axis has the absolute difference to the corresponding solution in objective 2. We could also use this sort of plot for objective 2 in the x axis with different results because, as the example in Fig. 6(e) has shown, we can have the same conflict located in different regions of different objectives.

In the first case of Fig. 7(a), there is no difference at all between any of the solutions because there is no conflict. In the opposite case of maximum conflict, Fig. 7(b) shows how the globality of conflict leads to a symmetric difference of values. Also we can see that in the case of maximum conflict it does not mean that there will be high values of difference everywhere because median solutions have less conflict.

Fig. 7(c) and (d) show cases of conflict which are specifically local, that is, concentrated on one particular region of the objective space. In those cases, the differences are respectively concentrated on the worst and best values. Symmetrical difference value graphs reflect on a smaller scale the graph of Fig. 7(b) because even though conflict is local, it is locally the

(a) Minimum Global
Non-Parametric Conflict

(b) Maximum Global
Non-Parametric Conflict

(c) Local Non-Parametric Conflict
for High Objective Values

(d) Local Non-Parametric Conflict
for Low Objective Values

(e) Global Medium
Non-Parametric Conflict

(f) Local Non-Parametric Conflict in
Different Locations in Each Objective

Fig. 6. Examples of many different possible kinds of conflict.

Table 2
Formulation of different measures of conflict.

Conflict	Formula	$C_{\min}$	$C_{\max}$
Direct Conflict	$C_{ab} = \sum_i \mathbf{X}'_{ia} - \mathbf{X}'_{ib} $	0	–
Maxmin Conflict	$\mathbf{X}'_{ij} = \mathbf{X}_{ij} - \min(\mathbf{X}_j)$ $C_{ab} = \sum_i \mathbf{X}'_{ia} - \mathbf{X}'_{ib} $	0	n
Non-parametric Conflict	$\mathbf{X}'_{ij} = \frac{\mathbf{X}_{ij} - \min(\mathbf{X}_j)}{\max(\mathbf{X}_j) - \min(\mathbf{X}_j)}$ $C_{ab} = \sum_i \mathbf{X}'_{ia} - \mathbf{X}'_{ib} $ $\mathbf{X}'_{ij} = \mathbf{R}_{ij}$	0	$\sum_{i=1}^n 2i - n - 1 $

Table 3
When each measure of conflict is best applicable.

Conflict	Objective importance	Objective unit	Insensitive to
Direct conflict	Equally important	Same units	Summation or subtractions
Maxmin conflict	Proportional to its range of values	Any comparable	Any linear normalization
Non-parametric conflict	It does not matter	Any	Any non-disruptive normalization

same behavior of maximum conflict. These two graphs illustrate how the distribution of the differences is concentrated in certain areas if the conflict is local.

Fig. 7(c) is a case of global conflict again. Similarly to the first two cases, the distribution of the differences is spread around the values on the right. However, the area is smaller than in Fig. 7(b) because there is less conflict. The last case of Fig. 7(f) is conflict located in a different region for the two objectives. In relation to objective 1, the conflict is for low rank solutions and the values of difference have formed areas of similar value because even though the lines for high rank values are parallel, there is a large difference in ranks for the group and, therefore, conflict for this whole set of solutions. However, what indicates the concentration of conflict in a certain region is that the area is more concentrated in low rank values for the first solutions, that is, solutions 1, 2, 3, 4, and 5. For the second half, the area is then spread on those high rank solutions, indicating the lower the solution's rank, the more conflict there is.

From graphs such as the one presented, we have then to identify whether the values of conflict are coming from low or high values. In order to do that, let $\mathbf{R}_{ij}$ represent the rank of the solution $\mathbf{x}_i$ on objective $f_j(\mathbf{x})$. Let $\mathbf{R}'_{ij}$ be these rank values linearly scaled from -1 to 1 for objective $f_j(\mathbf{x})$. The values of $\mathbf{R}'_{ij}$ now represent the weight of that solution for low (-1) or high ($+1$) rank solutions.

Each difference $|\mathbf{R}_{ia} - \mathbf{R}_{ib}|$ between ranks (as it is used by the non-parametric measure of conflict presented in Section 4) is multiplied by its weight $\mathbf{R}'_{ia}$ in order that we can infer whether those values are concentrated on high or low values. Positive values indicate concentration on high values and negative values indicate low values. This summation is then divided by $L(a, b)_{\max}$ in order that we can have a $L(a, b)$ value that ranges from -1 to $+1$. Thus, the following measure of conflict region $L(a, b)$ of objective a in relation to objective b is used for the Aggregation Trees:

$$L(a, b) = \frac{\sum_{i=1}^n |\mathbf{R}_{ia} - \mathbf{R}_{ib}| \mathbf{R}'_{ia}}{L(a, b)_{\max}} \quad (8)$$

$L(a, b)_{\max}$ is the maximum value of $L(a, b)$, which is described by the following equation:

$$L(a, b)_{\max} = \sum_{i=\lfloor n/2 \rfloor + 1}^n |2i - \lfloor 3n/2 \rfloor - 1| \mathbf{R}'_{ia} \quad (9)$$

$$L(a, b)_{\max} = \frac{2\lfloor n/2 \rfloor^2 - 2\lfloor n/2 \rfloor}{2}$$

Negative values indicate disharmony more concentrated on low rank solutions while positive values indicate disharmony concentrated around high rank values. This information is important for the decision maker because if conflict is concentrated on the best solutions, that means that there is an inherent conflict between the objectives even if other objectives are not present. This case is represented in red in the Aggregation Tree. When conflict is more concentrated on inferior solutions, the tree node is represented in blue. The intensity of those colors is defined according to the value $0 < |L(a, b)| < 1$ and if there is no concentration on any region, the node is represented in black.

(a) Minimum Global Conflict

(b) Maximum Global Conflict

(c) Local Conflict for High Objective Values

(d) Local Conflict for Low Objective Values

(e) Global Medium Conflict

(f) Local Conflict in Different Regions

Fig. 7. Representing the difference between the values represented in Fig. 6.

6. The algorithm for constructing an Aggregation Tree

The process of constructing an Aggregation Tree involves the measures of conflict and harmony between objectives as well as the region of that conflict. Those measures have been discussed in Section 4. The structure of the algorithm to generate the tree is described in Algorithm 1.

Algorithm 1. Constructing an aggregation tree.

Data: Set of Points $\mathbf{X}$
Result: Aggregation Tree t

- 1 Initialize tree t with a root node and all objectives as children;
- 2 $\mathbf{X} \leftarrow \text{normalize}(\mathbf{X})$;
- 3 **while** there are still objectives to be grouped **do**
- 4 $\mathbf{X}' \leftarrow \text{reduce}(\mathbf{X})$;
- 5 $\mathbf{X}' \leftarrow \text{normalize}(\mathbf{X}')$;
- 6 $\mathbf{H} \leftarrow \text{harmony_matrix}(\mathbf{X}')$;
- 7 $a, b \leftarrow$ leaf or compound objectives of $\mathbf{X}'$ with the greatest harmony;
- 8 $c \leftarrow \text{conflict}(\mathbf{X}', a, b)$;
- 9 $L \leftarrow \text{region}(\mathbf{X}', a, b)$;
- 10 t receives a new node nn ;
- 11 nn receives a and b as children;
- 12 nn keeps the values (c, L) ;
- 13 a and b are grouped. Next iteration has one less objective;
- 14 **end**
- 15 Plot the Aggregation Tree t ;
- 16 $order \leftarrow$ leaf nodes of t in the order as they appear in t ;
- 17 Plot the Polar Graph considering $order$;

In line 1, the structure of the tree is initialized with a root node as the parent of all objectives. In line 2, all the objective values are normalized with their rank values as demanded by the measures of non-parametric conflict described in Table 2. From line 3, an iterative loop begins. At each iteration of this loop, the two most harmonious objectives or compound objectives of the root node are grouped into a new parent node in the tree.

In line 4, a new version of the objective values is created for the iteration of the loop. This new version already includes the summation of the objectives grouped so far. In the last iteration of the algorithm, the reduction step transforms the objectives aggregated into only one objective. That can be done in many ways but in this work, we use the following strategy: (i) the rank values are summed, (ii) the results are ranked once more in a normalization step. Instead of using the average rank (as in [60]) we use re-ranked values so that the aggregated values are still in the range from 1 to n . Otherwise, summing rank values iteratively for a number of times tends to converge to a value equal to $n/2$ in all dimensions. That would lead the algorithm to lose its reference when aggregating groups with various objectives.

In line 5, this new version is normalized once more on the objectives that were grouped so that all objectives are in the same range of values. This step is used to re-rank the objective values.

In lines 6 and 7, the pair of objectives with the most harmony is calculated. This step is similar to what happens in a hierarchical agglomerative clustering technique with three important characteristics: (i) instead of aggregating the points at every step, we are aggregating objectives, (ii) the measure of distance between two dimensions is the difference of ranks between their solutions, and (iii) aggregated dimensions are treated as a new dimension where rank values are summed and then ranked once more.

In lines 8 and 9, the conflict and region of conflict are calculated for the most harmonious pair of objectives. In line 10, a new node is included in the tree as a child of the root node. In line 11, this new node receives the nodes that represent the most harmonious objectives so far. In line 12, the values of conflict and region of conflict for the most harmonious objectives are also kept by this new node. At this point, the root node has one objective less and a new iteration of the algorithm begins.

The idea behind the algorithm is that at every iteration the two most harmonic objectives are grouped into a new compound objective until there is only one compound objective, which would represent the simple normalized summation of all the objectives for mono-objective optimization. During the process of building the tree, we need simple matrices with information concerning the conflict between every two objectives. After the iterative process, in lines 15–17, we plot the resulting Aggregation Tree and its corresponding Polar Graph.

6.1. Time complexity

As mentioned in Section 6, the algorithm behind the trees is based on the iterative normalization and reduction of objectives. The more harmonious two objectives are, the better candidates they are to be reduced or grouped.

At each iteration, the values are normalized with their ranks. Ranking the values of one objective has the same cost as sorting its elements, that is, $O(n \log n)$ for comparison-based algorithms such as Quicksort and $O(kn)$ for other algorithms such as Radix sort, where k is the number of digits in each number. Thus, the first step of the algorithm costs $O(mn \log n)$ or $O(mn)$, where n is the number of solutions and m is the number of objectives.

This normalization process only costs $O(mn \log n)$ or $O(mn)$ for the first time the values are normalized and then $O(n \log n)$ or $O(n)$ for other iterations because only the objectives aggregated at the last iteration are not normalized.

Given a table **H** with all harmony values, the processes of calculating C_{ab} for every pair of objectives a and b ($O(nm^2)$), finding the two most harmonious objectives ($O(m^2)$), and reducing the objectives with maximum harmony ($O(n)$) have a total cost $O(\max(nm^2, m^2, n)) = O(nm^2)$.

Thus, the total cost of each iteration is $O(\max(n \log n, nm^2))$ (comparison-based sorting) or $O(\max(n, nm^2)) = O(nm^2)$ (counting-based sorting). As we have $m - 1$ reductions and iterations, we have a total cost of $O(\max(nm^3, mn \log n))$ or $O(nm^3)$ in our iterative loop. Thus, the complexity of the algorithm is either $O(\max(nm^3, mn \log n))$ for the general case comparison-based sorting algorithms or $O(\max(mn, knm^2)) = O(nm^3)$ for other sorting algorithms.

This time complexity $O(nm^3)$ is also closely related to the idea that the step of calculating the two most harmonious objectives, apart from the calculation of locality, can be equivalent to the calculation of distance in a hierarchical agglomerative clustering method where the dimensions are clustered instead of the points and the distance is measured by the difference of ranks. Those clustering techniques have time complexity of $O(mn^3)$, which become $O(nm^3)$ if we invert the points and their dimensions.

In order to show the relationship between the number of objectives and solutions, Fig. 8 shows the relationship between the number of objectives and the number of solutions in a real execution of the algorithm. The quality of solutions were randomly generated numbers with uniform distribution between 0 and 1. For each parameter, we run 50 replicates in order to have the median values, the maximum and minimum values, and the 25th and 75th percentiles.

We can see that the execution time has a cubic relationship to the number of objectives and a linear relationship to the number of solutions. Those results are expected according to the time complexity $O(nm^3)$. In the case of Fig. 8(b) we can also see that even though the execution time varies linearly there is a significant constant amount of time that is spent regardless of the number of solutions.

7. Results of benchmark instances

7.1. Test instances

We have run experiments on the DTLZ benchmark instances [15] in order to show how the tree can represent the relationship between objectives. For each instance, we use the algorithm PICEA-g [61] for 600 generations and 100 individuals to find a set of solutions for the problem with 20 objectives.

7.2. Results on the test function DTLZ-2

Fig. 9 shows the parallel coordinate plot with the solutions in the objective space for the test function DTLZ2. In order to facilitate visualization, solutions are clustered with a PSA algorithm [50] into 7 clusters and solutions in the same clusters have the same colors.

Even when allied to the PSA clustering technique, the parallel coordinate approach does not facilitate the cognitive perception of the relationship between objectives.

In Fig. 10, we plot the aggregation tree for the solutions to analyze the relationship between objectives. Objectives f_1 and f_2 are the two most harmonious ones, with 16.3% of non-parametric conflict, or 83.7% of harmony.

According to the tree and the harmony values, the next consecutive aggregations involve objectives f_3, f_4 in one branch, and then f_{10} and f_9 in another branch, respectively. Afterwards, there can be an aggregation of all those objectives into a new objective with only 5.6% of conflict.

As aggregations with lowest conflict happen first, we can find all the aggregations between the objectives until we have only one objective represented by the root node.

By looking at intermediate parent nodes, we can also monitor the relationship between groups of objectives. For instance, for the DTLZ2 instance, the groups which are most in conflict with one another are objectives (i) $f_{16} + f_{15}$, (ii) f_{17} , (iii) f_{18} , (iv) f_{19} , (v) f_{20} , and (vi) another group representing all other objectives. By looking at the branches that are close to the root node, the tree can help us make this kind of analysis for any number of clusters we want.

The relationship between groups can be transferred back to parallel coordinate plots to give us the representation of absolute objective values for all the objectives. However, the position of leaf nodes in the tree can suggest their position in the parallel coordinate plots in such a way that harmonious objectives are put together and the contrast between conflicting groups is clarified. Fig. 11 shows this rearranged representation of the solutions. The solutions were also clustered by a PSA algorithm.

(a) Varying the number of objectives (1000 solutions)

(b) Varying the number of solutions (20 objectives)

Fig. 8. Time spent to construct a tree.

In the rearranged graph, it is easier to perceive the difference between groups recognized by the tree. Firstly, we have a group with objectives 10, 9, 2, 1, 3, 4, 7, 6, 8, 5, 11, and 12 in the middle of graph. According to the tree, those are the most harmonious objectives.

To the left and to the right of this middle group, there are two conflicting groups separated by the tree. Solutions with high values for the objectives on the left tend to have low values for objectives on the right and vice versa. Thus, the combination of the Aggregation Tree with the parallel coordinate plots can lead to a better global understanding of the relationships between absolute objective values.

As mentioned in Section 4, we can represent these results on a polar graph in Fig. 12 to analyze the absolute objective values. We can see in this graph that the groups clustered by the PSA algorithm roughly coincide with the groups identified by the tree. This is especially visible in objectives 17, 18, 19, 20.

7.3. Results on the test function DTLZ-5

Similarly to the DTLZ2 experiments, Fig. 13 shows solutions for the DTLZ5 instance. In this instance, the relationship between adjacent objectives is already less chaotic than DTLZ2. This happens because the order of the objectives in this test problem is already in such a way that objective 2 is related to objective 1, objective 3 is related to objective 2, and so on.

Fig. 9. DTLZ2 – parallel coordinate plot.

Fig. 10. DTLZ2 – Aggregation Tree.

This scenario is virtually never the case in a real-world problem, where the order of the objectives is usually arbitrary. To represent this scenario, Fig. 14 shows the same solutions in a graph where the objectives have been randomly permuted. In this case, it is necessary to have a robust strategy to reorganize the objectives.

In Fig. 15 we have the aggregation tree for the DTLZ5 instance. In this tree, we have more well marked groups of conflict. The six biggest groups of conflict are (i) $f_{18} + f_{17}$, (ii) $f_{16} + f_{15} + f_{14} + f_{13} + f_5 + f_4 + f_9 + f_8 + f_7 + f_6 + f_{12}$, (iii) $f_{11} + f_{10} + f_3 + f_2$, (iv) f_1 , (v) f_{19} , and (vi) f_{20} .

The transposition of this data to the parallel coordinate plot representation and a polar graph is illustrated in Figs. 16 and 17. All the groups represented in the tree can be viewed in those representations. It is further noted that, in general, it is more difficult to optimize the objectives of the DTLZ5 instance.

Fig. 11. DTLZ2 – rearranged parallel coordinate plot.

Fig. 12. DTLZ2 – polar graph.

As a result of space limitations, the results for all other DTLZ test problems are available from www.alandefreitas.com.

7.4. Decision-making scenarios

Here we present four examples of strategies in which the decision-maker can use the Aggregation Trees and their respective polar graphs to restrict the preference area for the problem and search for an appropriate final solution.

Fig. 13. DTLZ5 – parallel coordinate plot.

Of course, the examples provided here are only a small representative selection of strategies in which the decision-maker can use the information available to reach a definitive solution. There is a myriad of contextual ways in which this can be accomplished in a real-world problem. Also, the strategies presented here can be used and reused in any order as the problem at hand demands.

7.4.1. Defining a harmony threshold for automatic objective aggregation

In Fig. 12, we can see that there is a group of objectives ($f_{10} + f_9 + f_2 + f_1 + f_3 + f_4 + f_7 + f_6 + f_8 + f_5 + f_{11}$) which could be optimized in most solutions, with the exception of outliers. This group of objectives corresponds to the objectives which, on inspection of the tree in Fig. 10, should be aggregated first.

As those are objectives that are in harmony, we aggregate all the original objectives that have less than 25% of conflict into a new objective function f_a . We then optimize this new problem and the new results are presented in Figs. 18 and 19.

Apart from the aggregated objectives, we obtain a similar tree for all other objectives. In the polar coordinates plot, we can see that f_a , and its component objectives, can be optimized in most solutions.

7.4.2. Transforming some objectives into constraints

In Figs. 18 and 19, we can now see that the group formed by objectives $f_{12} + f_a + f_{13} + f_{14} + f_{15} + f_{16}$ represents the first recommendations for aggregation in the tree. Besides, all clusters identified by the PSA algorithm have solutions with significantly low values for all of those objectives.

Fig. 14. DTLZ5 – parallel coordinate plot (objectives randomly permuted).

Fig. 17. DTLZ5 – polar graph.

Fig. 18. DTLZ2 (10 objectives) – Aggregation Tree.

We have now a small set of five optimized solutions that focus on a preference area defined by many strategies. This set can be readily analyzed by the decision-maker. For instance, the solution with best quality in f_{17} has objective values $f_{17} = 0.0000$, $f_{18} = 0.0015$, and $f_{19} = 0.0482$. The quality of those solutions in the original objective functions is presented

Fig. 19. DTLZ2 (10 objectives) – polar graph.

in Fig. 23. There is a great amount of diversity between those solutions, but it is important to remember that many of the aggregated objectives are only represented over a narrow range of values now as they have become constraints.

7.4.4. Optimizing large groups of relevant objectives

In this subsection, we reveal another possibility for the use of the information available in Figs. 20 and 21. This time, instead of removing the objective f_{20} , which is presented by the tree as the objective with the most conflict with other objectives, we optimize two groups of final objectives $f_b = f_{17} + f_{18} + f_{19}$ and f_{20} in order to explore and visualize the conflict between them.

Fig. 24 illustrates the quality of the solutions in 2 dimensions. As representative examples, the solutions with the best value in each the objectives have objective values $f_b = 0.0085, f_{20} = 1.8822$, and $f_b = 0.6363, f_{20} = 0.0148$. If the decision-maker chooses a solution with median values for both objectives, this solution has objective values $f_b = 0.3132, f_{20} = 1.4001$.

Also, the decision-maker can visualize the quality of those solutions in the original objectives, as it is shown in Fig. 25. Again, it might be difficult to analyze all solutions with the original objectives as many different strategies have been employed to reduce the preference area in the objective space. However, we can still recognize the objectives which were

Fig. 20. DTLZ2 (4 objectives) – Aggregation Tree.

Fig. 21. DTLZ2 (4 objectives) – polar graph.

Fig. 22. DTLZ2 (3 objectives) – polar graph.

restricted to a short range of values as constraints. Still, we can see the clear conflict between objective f_{20} and all other adjacent objectives in this set of solutions.

8. Discussion and future work

This paper proposes a tree-based method to visualize the relationship between objectives in many-objective problems. The trees are based on measures of conflict and harmony between objectives to infer their reducibility. In addition, a measure of region of conflict is used to color the aggregation nodes to represent where non-parametric conflict is occurring most.

Fig. 23. DTLZ2 (5 solutions – 20 objectives) – polar graph.

Fig. 24. DTLZ2 (20 solutions – 2 objectives).

The first main advantage of the method is the facility of iterative reduction to show the relationship between groups of objectives, regions of conflict, and their reducibility. The measure of conflict also informs the decision maker of the size of the penalty involved in the reduction of those objectives.

A second important feature is that all metrics used are non-parametric and can be generally used for any many-objective problem, being independent of any kind of specific linear or non-linear relationship between the objectives. The represen-

Fig. 25. DTLZ2 (20 solutions – 20 objectives) – polar graph.

tation of iterative aggregation also makes it convenient for the decision maker to choose which objectives can be effectively grouped. The measures of conflict make this decision easier.

The tree can also be used with (i) only non-dominated solutions to infer and analyze the relationship between the objectives on the Pareto front and therefore represent which objectives have the possibility of being optimized in a group, (ii) any set of random solutions to analyze the presence or absence of inherent harmony between objectives for any solutions and indicate which objectives could be reduced, or (iii) even solutions limited to a preference area in order to have a more specific analysis of the objectives in that area. Either way, unless we already have a very restricted preference area, as the number of objectives grows, this difference has a tendency to disappear because most solutions tend to be non-dominated.

Given the possibility of analysis of a Pareto front and a comparison between many objectives being plotted, the proposed approach is very simple to implement and has a low computational cost. In the case of a MOEA that uses objective reduction, a partially constructed tree could be used to indicate objectives to be reduced with low computational cost.

The iterative step of the algorithm always chooses the two most harmonious objectives in a greedy fashion. This choice may not always lead to the best tree if we consider the objectives that have already been paired. For this reason, the order of reduction to this objective reduction problem might be only a local optimum in relation to the number of objectives to be reduced.

An alternative approach might be to sum the conflict of all the children of a compound objective when choosing the next aggregation, in order that all aggregations are considered on a given iterative step.

As the algorithm uses iterative steps of objective aggregation, it is also possible that other techniques of objective reduction can be tested as the main aggregation method for the tree.

It is likely that the order of aggregation of the objectives and new measures of harmony that consider more than two objectives per aggregation will lead to a combinatorial model to be optimized. In many-objective optimization, from the point of view of Pareto-dominance, most points are on the Pareto front anyway. Thus, we might consider that estimating a good tree to the decision maker in order that they can narrow their preference area may be more important than being concerned with applying optimization algorithms to the problem itself unless there is a good measure of comparability between the objectives. Optimization could be better utilized, applied to the trees instead. Thus, estimating an optimal tree representing an optimal reduction strategy could become a primary goal of these many-objective problems.

Acknowledgements

This work has been supported by the Brazilian Agencies State of Minas Gerais Research Foundation – FAPEMIG, Coordination for the Improvement of Higher Level Personnel – CAPES, National Council of Scientific and Technological Develop-

ment – CNPq (Grants 30506/2010-2, 312276/2013-3); and the Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme.

References

- [1] G. Agrawal, C. Bloebaum, K. Lewis, Intuitive design selection using visualized n-dimensional pareto frontier, in: *Intuitive Design Selection using Visualized n-Dimensional Pareto Frontier*, American Institute of Aeronautics and Astronautics, 2005, <http://dx.doi.org/10.2514/6.2005-1813>.
- [2] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Hypervolume-based multiobjective optimization: theoretical foundations and practical implications, *Theor. Comput. Sci.* 425 (2012) 75–103.
- [3] J. Bader, E. Zitzler, Hype: an algorithm for fast hypervolume-based many-objective optimization, *Evol. Comput.* 19 (2010) 45–76, http://dx.doi.org/10.1162/EVCO_a_00009.
- [4] L. Batista, F. Campelo, F. Guimarães, J. Ramírez, A comparison of dominance criteria in many-objective optimization problems, in: *IEEE Congress on Evolutionary Computation*, IEEE, 2011, pp. 2359–2366.
- [5] L. Batista, F. Campelo, F. Guimarães, J. Ramírez, Pareto cone ϵ -dominance: improving convergence and diversity in multiobjective evolutionary algorithms, in: *Evolutionary Multi-Criterion Optimization*, Springer, 2011, pp. 76–90.
- [6] N. Beume, B. Naujoks, M. Emmerich, Sms-emoa: multiobjective selection based on dominated hypervolume, *Eur. J. Oper. Res.* 181 (2007) 1653–1669.
- [7] X. Blasco, J. Herrero, J. Sanchis, M. Martínez, A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization, *Inform. Sci.* 178 (2008) 3908–3924. Special Issue on Industrial Applications of Neural Networks, 10th Engineering Applications of Neural Networks 2007. <http://dx.doi.org/10.1016/j.ins.2008.06.010>.
- [8] D. Brockhoff, E. Zitzler, Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization, *Parallel Problem Solving from Nature-PPSN IX* (2006) 533–542.
- [9] D. Brockhoff, E. Zitzler, Dimensionality reduction in multiobjective optimization: the minimum objective subset problem, *Oper. Res. Proc.* 2006 (2007) 423–429.
- [10] C. Coello, G. Lamont, D. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and Evolutionary Computation series, Springer, 2007.
- [11] G.W. Corder, D.I. Foreman, *Nonparametric Statistics for Non-statisticians: A Step-by-step Approach*, John Wiley & Sons, 2009.
- [12] K. Deb, Multi-objective optimization using evolutionary algorithms, in: *Wiley-Interscience Series in Systems and Optimization*, Wiley, 2009.
- [13] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii, *Lecture Notes in Computer Science* 1917 (2000) 849–858.
- [14] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints, *IEEE Trans. Evol. Trans.* 18 (2014) 577–601, <http://dx.doi.org/10.1109/TEVC.2013.2281535>.
- [15] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: A. Abraham, L. Jain, R. Goldberg (Eds.), *Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing*, Springer, London, 2005, pp. 105–145, http://dx.doi.org/10.1007/1-84628-137-7_6.
- [16] P. Diaconis, R.L. Graham, Spearman's footrule as a measure of disarray, *J. Roy. Stat. Soc. Ser. B (Methodol.)* (1977) 262–268.
- [17] N. Drechsler, R. Drechsler, B. Becker, Multi-objective optimisation based on relation favour, in: *Evolutionary Multi-Criterion Optimization*, Springer, 2001, pp. 154–166.
- [18] J. Eddy, K.E. Lewis, Visualization of multidimensional design and optimization data using cloud visualization, in: *ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, 2002.
- [19] J. Fieldsend, R. Everson, Visualising high-dimensional pareto relationships in two-dimensional scatterplots, in: R. Purshouse, P. Fleming, C. Fonseca, S. Greco, J. Shaw (Eds.), *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, vol. 7811, Springer, Berlin, Heidelberg, 2013, pp. 558–572.
- [20] C. Fonseca, P. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation, *IEEE Trans. Syst. Man Cybernet. Part A: Syst. Humans* 28 (1998) 26–37, <http://dx.doi.org/10.1109/3468.650319>.
- [21] C. Fonseca, P. Fleming, et al., Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in: *Proceedings of the Fifth International Conference on Genetic Algorithms*, vol. 1, San Mateo, California, 1993, p. 416.
- [22] A.R. Freitas, P.J. Fleming, F.G. Guimaraes, A non-parametric harmony-based objective reduction method for many-objective optimization, in: *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2013, pp. 651–656. <http://dx.doi.org/10.1109/SMC.2013.116>.
- [23] G.W. Furnas, A. Bujia, Projection views: dimensional inference through sections and projections, *J. Comput. Graph. Stat.* 3 (1994) 323–353.
- [24] I. Giagkiozis, R. Purshouse, P. Fleming, Generalized decomposition, in: *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, vol. 7811, Springer, Berlin, 2013, pp. 428–442.
- [25] C. Grosan, A. Abraham, Hybrid line search for multiobjective optimization, in: *High Performance Computing and Communications* (2007) 62–73.
- [26] A. Hernández-Díaz, L. Santana-Quintero, C. Coello Coello, R. Caballero, J. Molina, A new proposal for multi-objective optimization using differential evolution and rough sets theory, in: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ACM, 2006, pp. 675–682.
- [27] E. Hughes, Evolutionary many-objective optimisation: many once or one many?, *The 2005 IEEE Congress on Evolutionary Computation*, vol. 1, IEEE, 2005, pp. 222–227.
- [28] E. Hughes, Msops-ii: A general-purpose many-objective optimiser, in: *IEEE Congress on Evolutionary Computation*, 2007 (CEC 2007), IEEE, pp. 3944–3951.
- [29] K. Ikeda, H. Kita, S. Kobayashi, Failure of pareto-based moeas: does non-dominated really mean near to optimal?, *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 2, IEEE, 2001, pp. 957–962.
- [30] H. Ishibuchi, T. Doi, Y. Nojima, Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms, *Parallel Problem Solving from Nature-PPSN IX* (2006) 493–502.
- [31] H. Ishibuchi, Y. Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning, *Int. J. Approx. Reason.* 44 (2007) 4–31.
- [32] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: A short review, in: *IEEE Congress on Evolutionary Computation*, 2008 (CEC 2008), (IEEE World Congress on Computational Intelligence), IEEE, pp. 2419–2426.
- [33] V. Khare, X. Yao, K. Deb, Performance scaling of multi-objective evolutionary algorithms, in: *Evolutionary Multi-Criterion Optimization*, Springer, 2003, pp. 72–72.
- [34] J. Knowles, D. Corne, Approximating the nondominated front using the pareto archived evolution strategy, *Evol. Comput.* 8 (2000) 149–172.
- [35] J. Knowles, D. Corne, Quantifying the effects of objective space dimension in evolutionary multiobjective optimization, in: *Evolutionary Multi-Criterion Optimization*, Springer, 2007, pp. 757–771.
- [36] M. Koppen, K. Yoshida, Visualization of pareto-sets in evolutionary multi-objective optimization, in: *7th International Conference on Hybrid Intelligent Systems*, 2007 (HIS 2007), IEEE, pp. 156–161.
- [37] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, *Evol. Comput.* 10 (2002) 263–282.
- [38] M. Laumanns, E. Zitzler, L. Thiele, On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization, in: *Evolutionary Multi-criterion Optimization*, Springer, 2001, pp. 181–196.

- [39] H.L. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, *IEEE Trans. Evol. Comput.* 18 (2014) 450–455, <http://dx.doi.org/10.1109/TEVC.2013.2281533>.
- [40] A. López Jaimes, C.A. Coello Coello, D. Chakraborty, Objective reduction using a feature selection technique, in: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO '08*, ACM, New York, NY, USA, 2008, pp. 673–680, <http://dx.doi.org/10.1145/1389095.1389228>.
- [41] A. Lotov, Approximation and visualization of pareto frontier in the framework of classical approach to multi-objective optimization, in: J. Branke, K. Deb, K. Miettinen, R.E. Steuer (Eds.), *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany, 2005.
- [42] M. Luque, K. Miettinen, P. Eskelinen, F. Ruiz, Incorporating preference information in interactive reference point methods for multiobjective optimization, *Omega* 37 (2009) 450–462.
- [43] F. Mascia, M. Brunato, Techniques and tools for local search landscape visualization and analysis, in: T. Stützle, M. Birattari, H. Hoos (Eds.), *Engineering Stochastic Local Search Algorithms: Designing, Implementing and Analyzing Effective Heuristics*, Lecture Notes in Computer Science, vol. 5752, Springer, Berlin Heidelberg, 2009, pp. 92–104.
- [44] C.A. Mattson, A. Messac, Concept selection using s-pareto frontiers, *AIAA J.* 41 (2003) 1190–1198, <http://dx.doi.org/10.2514/2.2063>.
- [45] J. Molina, L. Santana, A. Hernández-Díaz, C. Coello Coello, R. Caballero, g-dominance: reference point based dominance for multiobjective metaheuristics, *Eur. J. Oper. Res.* 197 (2009) 685–692.
- [46] S. Obayashi, D. Sasaki, Visualization and data mining of pareto solutions using self-organizing map, in: *Evolutionary Multi-criterion Optimization*, Springer, 2003, pp. 71–71.
- [47] R. Purshouse, P. Fleming, Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation, in: C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, K. Deb (Eds.), *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, vol. 2632, Springer, Berlin, Heidelberg, 2003, pp. 16–30, http://dx.doi.org/10.1007/3-540-36970-8_2.
- [48] R. Purshouse, P. Fleming, On the evolutionary optimization of many conflicting objectives, *IEEE Trans. Evol. Comput.* 11 (2007) 770–784.
- [49] S. Salomon, G. Avigad, A. Goldvard, O. Schütze, Psa – a new scalable space partition based selection algorithm for moeas, in: O. Schütze, C.A. Coello Coello, A.A. Tantar, E. Tantar, P. Bouvry, P. Del Moral, P. Legrand (Eds.), *EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, Advances in Intelligent Systems and Computing, vol. 175, Springer, Berlin, Heidelberg, 2013, pp. 137–151.
- [50] S. Salomon, C. Domínguez-Medina, G. Avigad, A. Freitas, A. Goldvard, O. Schütze, H. Trautmann, Psa based multi objective evolutionary algorithms, in: O. Schuetze, C.A. Coello Coello, A.A. Tantar, E. Tantar, P. Bouvry, P.D. Moral, P. Legrand (Eds.), *EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III*, Studies in Computational Intelligence, vol. 500, Springer International Publishing, 2014, pp. 233–259, http://dx.doi.org/10.1007/978-3-319-01460-9_11.
- [51] H. Sato, H. Aguirre, K. Tanaka, Controlling dominance area of solutions and its impact on the performance of moeas, in: *Evolutionary Multi-Criterion Optimization*, Springer, 2007, pp. 5–20.
- [52] H. Sato, H. Aguirre, K. Tanaka, Local dominance moea including control of dominance area of solutions on 0/1 multiobjective knapsack problems, *Inform. Media Technol.* 4 (2009) 33–43.
- [53] D. Saxena, K. Deb, Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: employing correntropy and a novel maximum variance unfolding, in: *Evolutionary Multi-Criterion Optimization*, Springer, 2007, pp. 772–787.
- [54] D. Saxena, J. Duro, A. Tiwari, K. Deb, Q. Zhang, Objective reduction in many-objective optimization: linear and nonlinear algorithms, *IEEE Trans. Evol. Comput.* 17 (2013) 77–99, <http://dx.doi.org/10.1109/TEVC.2012.2185847>.
- [55] D. Saxena, T. Ray, K. Deb, A. Tiwari, Constrained many-objective optimization: a way forward, in: *IEEE Congress on Evolutionary Computation, 2009 (CEC'09)*, IEEE, pp. 545–552.
- [56] H. Singh, A. Isaacs, T. Ray, A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems, *IEEE Trans. Evol. Comput.* 15 (2011) 539–556, <http://dx.doi.org/10.1109/TEVC.2010.2093579>.
- [57] S. Singh, V. Singh, Three-level ahp-based heuristic approach for a multi-objective facility layout problem, *Int. J. Product. Res.* 49 (2011) 1105–1125.
- [58] T. Tušar, B. Filipič, Visualizing 4d approximation sets of multiobjective optimizers with projections, in: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO '11*, ACM, New York, NY, USA, 2011, pp. 737–744.
- [59] D. Walker, R. Everson, J. Fieldsend, Visualizing mutually nondominating solution sets in many-objective optimization, *IEEE Trans. Evol. Comput.* 17 (2013) 165–184.
- [60] D.J. Walker, R.M. Everson, J.E. Fieldsend, Rank-based dimension reduction for many-criteria populations, in: *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '11*, ACM, New York, NY, USA, 2011, pp. 107–108, <http://dx.doi.org/10.1145/2001858.2001919>.
- [61] R. Wang, R. Purshouse, P. Fleming, Preference-inspired co-evolutionary algorithms for many-objective optimisation, *IEEE Trans. Evol. Comput.* 17 (2013) 474–494.
- [62] T. Wang, Z. Wu, J. Mao, A new method for multi-objective tdma scheduling in wireless sensor networks using pareto-based pso and fuzzy comprehensive judgement, *High Performance Computing and Communications* (2007) 144–155.
- [63] U. Wickramasinghe, X. Li, Using a distance metric to guide pso algorithms for many-objective optimization, in: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM*, 2009, pp. 667–674.
- [64] J. Xiong, Y. Chen, K. Yang, Q. Zhao, L. Xing, A hybrid multiobjective genetic algorithm for robust resource-constrained project scheduling with stochastic durations, *Math. Probl. Eng.* 2012 (2012).
- [65] M. Yoshimi, T. Kuhara, K. Nishimoto, M. Miki, T. Hiroyasu, Visualization of pareto solutions by spherical self-organizing map and its acceleration on a gpu, *J. Softw. Eng. Appl.* 5 (2012).
- [66] Q. Zhang, H. Li, Moea/d: a multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (2007) 712–731, <http://dx.doi.org/10.1109/TEVC.2007.892759>.
- [67] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: *Parallel Problem Solving from Nature-PPSN VIII*, Springer, 2004, pp. 832–842.
- [68] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, L. Thiele, Spea2: improving the strength pareto evolutionary algorithm, 2001.
- [69] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, *Trans. Evol. Comput.* 3 (1999) 257–271, <http://dx.doi.org/10.1109/4235.797969>.
- [70] S. Zou, H. Zhang, K. Wang, Fuzzy-ga and multi-objective transportation optimization, in: *2008 IEEE Conference on Cybernetics and Intelligent Systems*, IEEE, 2008, pp. 270–273.