# On the Visualization of Trade-offs and Reducibility in Many-Objective Optimization


Alan R. R. Freitas
Department of Computer Science
Universidade Federal de Ouro Preto
Ouro Preto, Brazil
alandefreitas@iceb.ufop.br

Rodrigo C. P. Silva
Department of Electrical and Computer Engineering
McGill University
Montreal, QC
Canada
rodrigo.silva@mail.mcgill.ca

Frederico G. Guimarães
Department of Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
fredericoguimaraes@ufmg.br



## ABSTRACT

This paper proposes a technique of Aggregation Trees to visualize the results of high-dimensional multiobjective optimization problems, or many-objective problems. The high-dimensionality makes it difficult to represent the relation between objectives and solutions. Most approaches in the literature are based on the representation of solutions in lower dimensions. The technique of Aggregation Trees proposed here is based on iterative aggregation of objectives which are represented in a tree. Besides, the location of conflict is also calculated and represented on the tree. Thus, the tree can represent which objectives and groups of objectives are harmonic the most, what sort of conflict is present between groups of objectives, and which aggregations would be more interesting in order to reduce the problem dimension.



## Categories and Subject Descriptors

G.1.6 [**Optimization**]: Stochastic programming; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

## Keywords

Aggregation Trees, Multiobjective Optimization, Many-Objective Optimization, Visualization, Objective Reduction


## 1. INTRODUCTION

Multiobjective optimization is a very important tool to solve real-world problems. As the number of objectives grow in those problems, we reach the field of Many-objective Problems (Section 2). Visualizing the relation between objectives in those problems becomes more difficult as most solutions become incomparable in relation to Pareto dominance.

In this paper, we propose a method of Aggregation Trees (Section 3) to represent the relation between objectives according to their reducibility in a way that we can represent the relation between groups of objectives for a problem.

Thus, Aggregation Trees can be used to easily visualize the results of Many-objective problems, group objectives according to their reducibility, show the amount of conflict and harmony between objectives. The technique is based on an objective reduction algorithm.

Also, in the context of many-objective optimization, the trees provide the decision maker with information on where it could be convenient to restrict the preference area for the next optimization round.

We then show results for some test problems (Section 6) and conclude our paper with discussions about the method and suggestions concerning possible future work (Section 7).

## 2. VISUALIZATION IN MANY-OBJECTIVE OPTIMIZATION

A multiobjective optimization problem can be mathematically defined as:

$$\min(f_1(x), f_2(x), \ldots, f_k(x)),\ x \in \mathcal{F} \tag{1}$$

where $f_i(x)$ is the i-th objective function to be minimized. Each function $f_i(x)$ maps the optimization variables of a candidate solution $x$ to an objective value represented in one dimension of the objective space. The set of all combinations of possible values for optimization variables defines the search space:

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

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

If the problem has constraints, those constraint functions define the feasible set of solutions:

$$\mathcal{F} = \{x \in \mathcal{S} : g_k(s) \leq 0, k = 1, \ldots, r\} \tag{3}$$

Thus, solving the optimization problem defined in Equation 1 means to find the set of solutions which are Pareto-optimal, that is, solutions that cannot be improved in any of the objectives without implying in a worse result for another objective. In mathematical terms, a feasible solution $x^1 \in \mathcal{F}$ dominates another solution $x^2 \in \mathcal{F}$ if the conditions



*GECCO'14,* July 12–16, 2014, Vancouver, BC, Canada.
Copyright 2014 ACM 978-1-4503-2881-4/14/07 ...$15.00.
http://dx.doi.org/10.1145/2598394.2610550.

below are attained:

$$f_i(x^1) \leq f_i(x^2) \qquad \text{for all i} \in \{1, 2, 3, \ldots, \text{k}\} \tag{4}$$

$$f_j(x^1) < f_j(x^2) \quad \text{for at least one j} \in \{1, 2, 3, \ldots, \text{k}\} \tag{5}$$

With a Pareto-based fitness assignment scheme, Evolutionary Algorithms have been the most common strategy to tackle Multiobjective Problems [7]. Those strategies usually permit diversity control [3], real-time interaction with the decision maker [34], and are elitist [8, 17].

However, the difficulty of the problem grows exponentially with the number of objectives and problems with more than three objectives have been called Many-objective Problems [15]. Besides the memory cost of representing a multidimensional Pareto-front, the convergence of Evolutionary Algorithms is compromised because Pareto-based ranking does not work as a good discriminator of solutions in those problems [16] as most solutions are non-dominated. Even when we have a reasonable solution, the visualization of those solutions in the objective space is also a difficult problem for the Decision Maker.

Palliative solutions for those problems are (i) considering more relaxed dominance relations to facilitate discrimination [3], (ii) modifying fitness measures [10, 2], (iii) optimizing scalarized single-objective functions [33], and (iv) dimensionality reduction through identification of redundant objectives [13].

Still in the category of objective reduction, the main approaches are (i) finding a minimum objective subset within a given error and according to a dominance relation[5, 6], (ii) Principal Component Analysis (PCA) or PCA techniques adapted to handle non-linear relations between objectives [27, 28], (iii) unsupervised feature selection [20], (iv) focusing on the corners among non-dominated solutions [29], and (v) mathematical formulation and aggregation of the most harmonic objectives [13].

The results for problems with 2 or 3 dimensions can be easily represented in 2 or 3 axis as in Figure 1. In that Figure, we have the quality of many solutions, each represented by a point. The non-dominated solutions are the ones marked with a number 1 in the first front and all the other solutions are worse than at least one solution in the first front. In most cases, the dominated solutions are not shown in the graph in order to make the information clearer.

For visualizing non-dominated results in many-objective optimization, the most common approach is parallel coordinates, as represented in Figure 2 for 7 objectives. In this graph, each line represents the quality of a solution and the objective values are normalized. When there are many lines crossing between 2 adjacent objectives, that means there is conflict between those objectives. Problems with this representation method are that (i) we can only see conflict between adjacent objectives; (ii) the representation can become very confusing when the number of solutions or objectives increase.

Other approaches for visualization of solution quality in many-objective optimization are (i) trying to map the values on lower dimensions with minimum loss of information [22, 18], (ii) cloud visualization [11], (iii) self-organizing maps (SOM) [23, 35], (iv) Interactive Decision Maps [21], (v) a Hyper-space Diagonal Counting method [1], (vi) level diagrams [4], and prosections [30].

In more recent approaches, Kurasova et al. [19] propose an approach based on neural gas clustering and multidimensional scaling dimension reduction to project the solution onto a plane, where the points are marked with their distance to the ideal point.

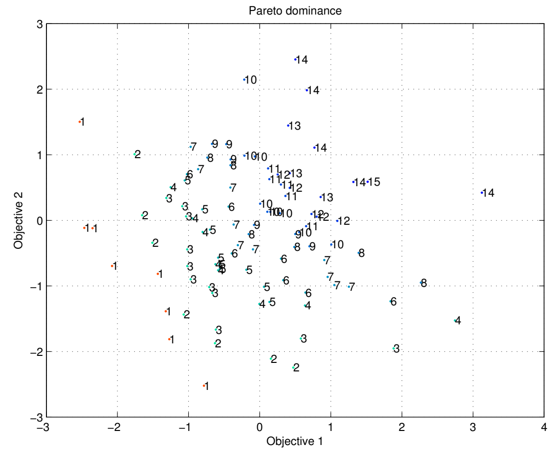


**Figure 1: Results for a 2-dimensional problem**

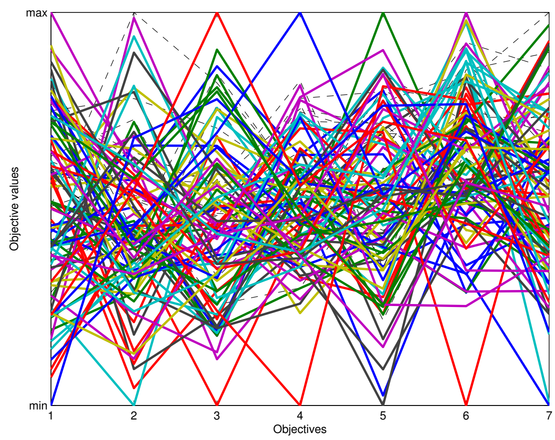


**Figure 2: Representing the quality of solution for Many-objective Problems**

Walker et al. [31] did an extensive review on 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 [12] 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 this context, we need visualization methods in which we can visualize the relation between objectives and groups of objectives with minimum loss of information. Besides, it is important to give relevant information to the decision maker so that they can choose how to best restrict the preference area until they get to a single solution.

## 3. AGGREGATION TREES

The Aggregation Trees presented on this paper are based on a specific objective reduction algorithm [13] that measures harmony between objectives. The more harmonious two objectives are the higher is their reducibility [24, 25].

In order to exemplify the utility of the proposed Harmony Trees, consider the set of solutions represented in the Parallel Coordinates of Figure 3. Objectives 1, 2, 4, 6, 8, 10, and 12 have the same values for all solutions and therefore are in complete harmony.

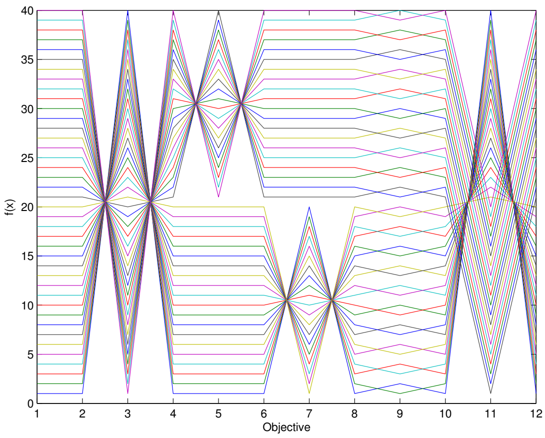


**Figure 3: A set of solutions with many sorts of conflict**

The Aggregation Tree in Figure 4 represents the relation between those objectives and their reducibility in the structure of a tree. The child nodes represent objectives and parents represent aggregations of those objectives. The percentages on parent nodes represent the conflict between its two children.

Parents represent a compound objective formed by the aggregation of the child objectives and renormalization of the values. Below every parent node, we have the children that are aggregated in that node and the conflict between its two direct children. For example, the node $f_{11} + f_3 -$ 25% represents a compound objective with the aggregation of objectives $f_{11}$ and $f_3$ through summation. Those two objectives have 25% of non-parametric conflict, which can be used as a measure of the loss of information implied by this aggregation on the representation of the Pareto front. Other parent nodes represent higher order compound objectives formed in a similar way but their value of conflict represent only their direct children.

By visualizing objectives in the Aggregation Tree, it is easy to see the relation between the objectives and in case the decision maker wants to make compound objectives to reduce their preference area, the nodes give information on which would be the best ones to group. Instead of only showing the conflict between every two objectives, the tree also shows the conflict between groups of objectives that could be put together.

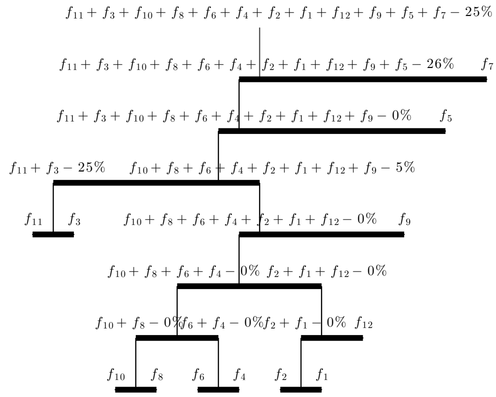


**Figure 4: Aggregation Tree on the set of solutions from Figure 3**

The order of aggregation of the objectives is according the their harmony and distant leaf nodes in the tree are objectives with little harmony. Thus, if we perform a depth-first search on the child nodes of the tree, we can have a recommendation of a convenient way to choose good adjacent objectives in parallel coordinates, such as the one in Figure 5, where it is easier to see the relation between the objectives as the most harmonious objectives are grouped together.

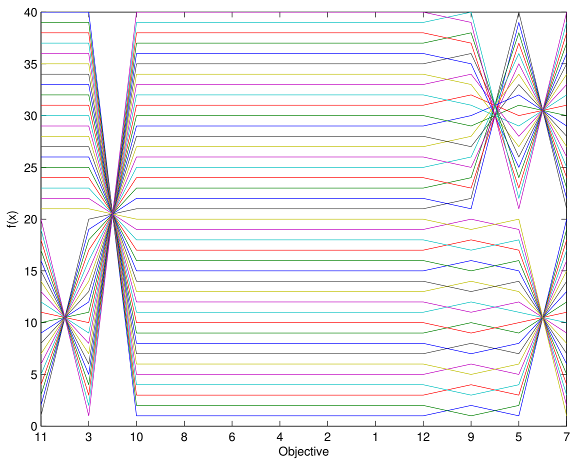


**Figure 5: Parallel coordinates with the objectives ordered by the tree.**

Thus, an Aggregation Tree can help decision makers visualize redundancy, conflict, and harmony between objectives and groups of objectives. The source code in MATLAB for all those algorithms is available from `http://www.alandefreitas.com/`.

# 4. OBJECTIVE AGGREGATION

Figure 3 represented many kinds of conflict and harmony among many objectives in parallel coordinate graphs. When good values for one objective imply bad values for another objective, there is conflict between these objectives [25, 14]. When improvement in one objective leads to improvement in another objective, there is harmony between these objectives.

Harmonious objectives can be perceived in parallel coordinates by non-crossing lines. This relation makes it related to the possibility of reducing the number of objectives without losing information in the Pareto front. According to those definitions, we use a non-parametric metric of harmony [13] to measure which pair of objectives should be aggregated next and represented on the tree.

The non-parametric measure $H_{ab}$ of harmony between objectives $f_a$ and $f_b$ is defined in Equation 6. $\mathbf{X}_{ij}$ is the value of the solution $i$ on the objective $j$ and $\mathbf{R}_{ij}$ is the rank of $\mathbf{X}_{ij}$ within $\mathbf{X}_{.j}$. Thus, $\mathbf{X'}_{ij}$ represents the values of $\mathbf{X}_{ij}$ normalized according to $\mathbf{R}_{ij}$. $C_{ab}$ is the non-parametric conflict between the objectives $f_a$ and $f_b$. This measure is equivalent to Spearman's footrule between the ranks in objectives $f_a$ and $f_b$. $c_{\min} = 0$ is the minimum value possible of conflict between two objectives. $c_{\max}$ is the maximum value of conflict for two objectives, being $n$ the existing number of points.

$$\begin{aligned} \mathbf{X'}_{ij} &= \mathbf{R}_{ij} \\ C_{ab} &= \sum_i |\mathbf{X'}_{ia} - \mathbf{X'}_{ib}| \\ c_{\max} &= \sum_{i=1}^{n} |2i - n - 1| \\ H_{ab} &= 1 - \frac{C_{ab}}{c_{\max}} \end{aligned} \tag{6}$$

This measure of harmony is insensitive to any previous non-disruptive normalization of the values, that is, insensitive to any normalization that does not alter the order or the values in the set of solutions. That means that the objectives can use different units. In fact, the measure reflects how much the lines would be crossing between objectives $f_a$ and $f_b$ on parallel coordinates. When we divide $C_{ab}$ by $c_{\max}$ to get our measure of harmony, we guarantee that all harmony values range from 0 to 1.

As mentioned before, the most 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. THE PSEUDOCODE

The construction of a tree is an iterative process in which we aggregate two objectives at each iteration. In order to aggregate two objectives, we calculate the harmony between every pair of objectives. The process is described in Algorithm 1.

A more detailed description is as follows:

- Line 1: structure of the tree is initialized with a root node as the parent of all objectives.

**Data**: Set $\mathbf{X}$ of Points in the Objective Space
**Result**: Harmony Tree $t$
1 Initializes tree $t$ with a root node and all objectives as children;
2 **while** *there are still objectives to be grouped* **do**
3     $\mathbf{X'} \leftarrow$ reduce($\mathbf{X}$);
4     $\mathbf{X'} \leftarrow$ normalize($\mathbf{X'}$);
5     $\mathbf{H} \leftarrow$ harmony_matrix($\mathbf{X'}$);
6     $a, b \leftarrow$ leaf or compound objectives of $\mathbf{X'}$ with the most harmony;
7     $c \leftarrow$ conflict($\mathbf{X'}, a, b$);
8     $t$ receives a new node $nn$;
9     $nn$ receives $a$ and $b$ as children;
10     $nn$ keeps the values ($c$);
11     $a$ and $b$ are grouped. Next iteration has one objective less;
12 **end**
13 Plot the Harmony Tree $t$;
14 *order* $\leftarrow$ leaf nodes of $t$ in the order as they appear in $t$;
15 Plot the objective values in parallel coordinates considering *order*;

**Algorithm 1:** Constructing an Aggregation Tree

- Line 2: the iterative loop begins. At each iteration, the two most harmonious objectives will be aggregated into a new parent node.
- Line 3: a new version of the objective values is created for the iteration of the loop. That new version considers all the aggregations done so far. Those aggregations are performed by ranking the objective values and summing them. This summation can lead to values that can go from 1 to $2n$ because the rank values of two objectives are being considered in the summation.
- Line 4: this new version of $\mathbf{X'}$ is normalized once more on the aggregated objectives to find the ranking values $\mathbf{R}_{ij}$ from 1 to $n$.
- Lines 5-6: the pair of objectives with the most harmony is calculated.
- Line 7: the conflict between the two most harmonious objectives is calculated.
- Line 8: a new node is included in the tree as a child of the root node.
- Line 9: this new node receives the nodes that were representing the most harmonious objectives so far.
- Line 10: the value of conflict for the most harmonious objectives is also kept by this new node. The root node has now one objective less and a new iteration of algorithm begins in line 3.
- Line 13: we plot the resulting tree.

Briefly, the reduction algorithm aggregates the two most harmonic objectives at each iteration until there is only one objective left. The tree represents all the aggregations until we form the single objective that represents the simple summation of all objective values and the values of conflict are represented. The order of the elements in the tree suggests an order for the absolute objective values to be plotted on parallel coordinates.

Being $n$ the number of solutions and $m$ the number of objectives, the normalization of the results is $O(mn \log n)$ at the first iteration and $O(n \log n)$ at other iterations. Calculating the harmony matrix has cost $O(nm^2)$. Thus, being $m$ also the number of iterations, the final cost of the algorithm is $O(\max(m^3 n, mn \log n))$. If we use a comparison-based sorting algorithm $O(n)$, we have a total cost $O(m^3 n)$ at all cases.

# 6. RESULTS ON TEST PROBLEMS

In order to show how the tree can represent the trade-offs involved in the results of the optimization of many objectives, we use the algorithm PICEA-g [32] to optimize solutions for the test problem DTLZ2 [9].

This test function is useful to investigate the scalability of MOEAs in the context of many-objectives. The function is optimized for 20 objectives during 600 generations with 100 individuals each. Figure 6 shows the solutions in the objective space. In order to facilitate visualization, the solutions clustered with a PSA algorithm [26] into 7 clusters and solutions in the same clusters have the same colors.

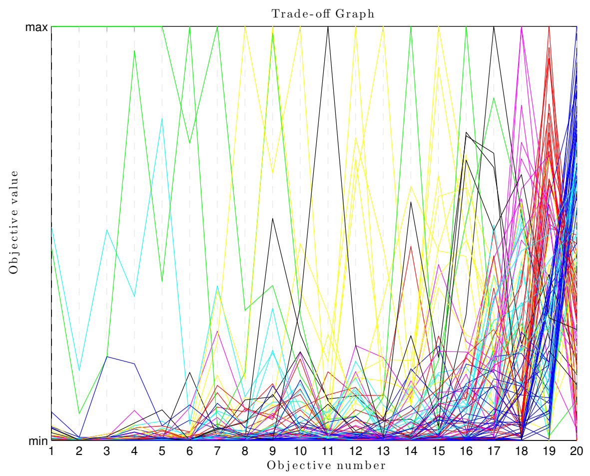


**Figure 6: DTLZ2 - Parallel Coordinates**

As expected, it is difficult to notice the exact relation between the objectives. Thus, we analyse the relation between objectives and show the resulting tree in Figure 7. The two most harmonious objectives are $f_2$ and $f_3$, with only 18.172% of non-parametric conflict.

According to the tree and the harmony values, the next consecutive aggregations involve objectives $f_1$, $f_4$, $f_{11}$, $f_{10}$, respectively. As we know that the aggregations with lowest conflict happen first, we can find all the aggregations between the objectives until we have only one objective, which would be the normalized summation of all objective values, represented by the root node.

By looking at intermediate parent nodes, we can also notice the relation between groups of objectives. This relation can be transferred back to parallel coordinates 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 parallel coordinates in such a way that harmonious objectives are put together and the contrast between conflicting groups is made clear. Figure 8 shows this rearranged representation of the solutions. The solutions were also clustered by a PSA algorithm.

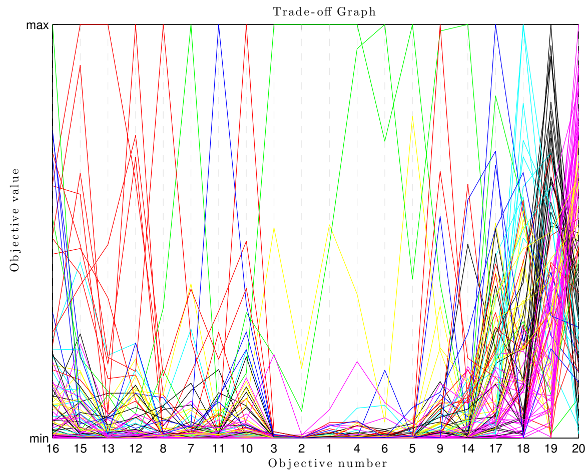


**Figure 8: DTLZ2 - Rearranged Parallel Coordinates**

In the rearranged graph it is easier to perceive the difference between groups recognized by the tree. Firstly, we have a group with objectives 3, 2, 1, 4, 6, and 5 in the middle of graph. Those are the most harmonious objectives according to the tree. By looking at their objective values, most solutions are very close to their minimum at each of the 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 tree with parallel coordinates can give a better global understanding of the relation between objectives and their values.

Due to space limitation, the results for all other DTLZ test problems are available from `www.alandefreitas.com`.

# 7. DISCUSSION AND FUTURE WORK

In this paper we presented the Aggregation Trees. The tree is a method for visualization of many-objective solutions. Nodes represent aggregations of objectives which are iteratively detected according to their harmony so the relation between groups of relevant objectives can be analysed.

Non-parametric measures of conflict and harmony are used to build the tree. Therefore, the tree does not depend on any implicit relation between the objectives to infer their reducibility. Each node has also the value of conflict involved in the last aggregation. This is useful for the decision maker as they can understand the cost involved in each of those reductions.

If the decision maker wants to reduce their preference area to focus on a more specific set of solutions, the way in which the tree represents iterative aggregations can show which aggregation should happen first.

The trees can also include dominated solutions in the analysis. That makes the measure of harmony detect if objec-

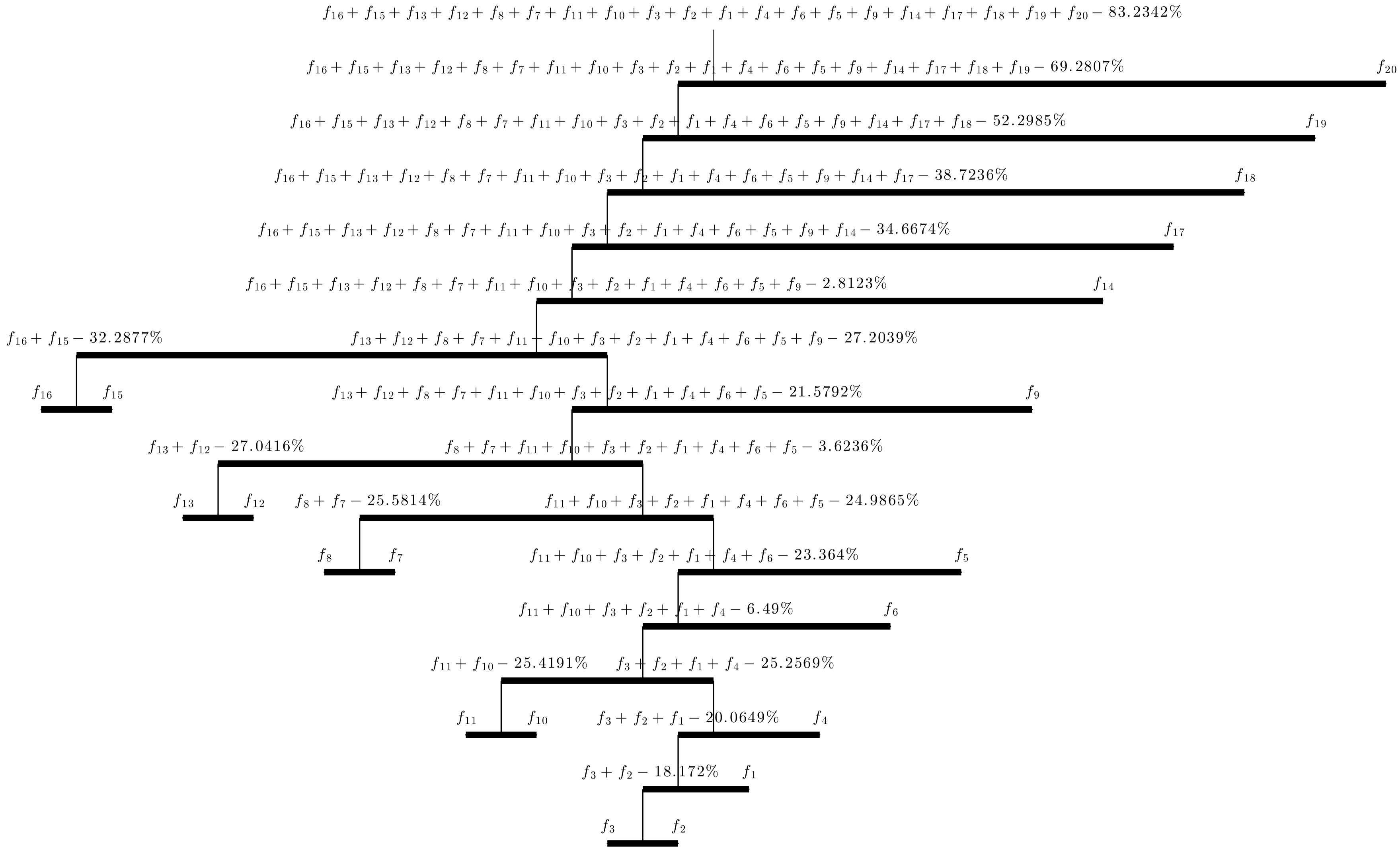


**Figure 7: DTLZ2 - Aggregation Tree**

tives are correlated even for non-optimal solutions. However, if the number of objectives is very high, most solutions will tend to be non-dominated.

As future work, it is possible to study the effect of other objective reduction techniques in the formation of the tree.

## Acknowledgements

This work has been supported by the Brazilian agencies CAPES, CNPq, and FAPEMIG. We would also like the thank the support from the Federal University of Ouro Preto (UFOP) and the Federal University of Minas Gerais (UFMG).

## 8. REFERENCES


[1] G. Agrawal, C. Bloebaum, and K. Lewis. *Intuitive Design Selection using Visualized n-Dimensional Pareto Frontier*. American Institute of Aeronautics and Astronautics, 2014/01/16 2005.

[2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425:75–103, 2012.

[3] L. Batista, F. Campelo, F. Guimarães, and J. Ramírez. Pareto cone $\varepsilon$-dominance: improving convergence and diversity in multiobjective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, pages 76–90. Springer, 2011.

[4] X. Blasco, J. Herrero, J. Sanchis, and M. Martínez. A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908 – 3924, 2008. <ce:title>Special Issue on Industrial Applications of Neural Networks</ce:title> <xocs:full-name>10th Engineering Applications of Neural Networks 2007</xocs:full-name>.

[5] D. Brockhoff and E. Zitzler. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. *Parallel Problem Solving from Nature-PPSN IX*, pages 533–542, 2006.

[6] D. Brockhoff and E. Zitzler. Dimensionality reduction in multiobjective optimization: The minimum objective subset problem. *Operations Research Proceedings 2006*, pages 423–429, 2007.

[7] C. Coello, G. Lamont, and D. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and evolutionary computation series. Springer, 2007.

[8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Lecture notes in computer science*, 1917:849–858, 2000.

[9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pages 105–145. Springer London, 2005.

[10] N. Drechsler, R. Drechsler, and B. Becker. Multi-objective optimisation based on relation favour. In *Evolutionary Multi-Criterion Optimization*, pages 154–166. Springer, 2001.

[11] J. Eddy and 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.

[12] J. Fieldsend and R. Everson. Visualising high-dimensional pareto relationships in two-dimensional scatterplots. In R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw, editors, *Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 558–572. Springer Berlin Heidelberg, 2013.

[13] A. R. Freitas, P. J. Fleming, and F. G. Guimaraes. A non-parametric harmony-based objective reduction method for many-objective optimization. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 651–656, 2013.

[14] I. Giagkiozis, R. Purshouse, and P. Fleming. Generalized decomposition and cross entropy methods for many-objective optimization. 2012.

[15] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2419–2426. IEEE, 2008.

[16] V. Khare, X. Yao, and K. Deb. Performance scaling of multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, pages 72–72. Springer, 2003.

[17] J. Knowles and D. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000.

[18] M. Koppen and K. Yoshida. Visualization of pareto-sets in evolutionary multi-objective optimization. In *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on*, pages 156–161. IEEE, 2007.

[19] O. Kurasova, T. Petkus, and E. Filatovas. Visualization of pareto front points when solving multi-objective optimization problems. *Information Technology And Control*, 42(4):353–361, 2013.

[20] A. López Jaimes, C. A. Coello Coello, and D. Chakraborty. Objective reduction using a feature selection technique. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, pages 673–680, New York, NY, USA, 2008. ACM.

[21] 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, and R. E. Steuer, editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[22] C. A. Mattson and A. Messac. Concept selection using s-pareto frontiers. *AIAA Journal*, 41(6):1190–1198, 2014/01/16 2003.

[23] S. Obayashi and D. Sasaki. Visualization and data mining of pareto solutions using self-organizing map. In *Evolutionary multi-criterion optimization*, pages 71–71. Springer, 2003.

[24] R. Purshouse and P. Fleming. Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, and K. Deb, editors, *Evolutionary Multi-Criterion Optimization*, volume 2632 of *Lecture Notes in Computer Science*, pages 16–30. Springer Berlin Heidelberg, 2003.

[25] R. Purshouse and P. Fleming. On the evolutionary optimization of many conflicting objectives. *Evolutionary Computation, IEEE Transactions on*, 11(6):770–784, 2007.

[26] S. Salomon, C. Domínguez-Medina, G. Avigad, A. Freitas, A. Goldvard, O. Schütze, and 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, and P. Legrand, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III*, volume 500 of *Studies in Computational Intelligence*, pages 233–259. Springer International Publishing, 2014.

[27] D. Saxena and 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*, pages 772–787. Springer, 2007.

[28] D. Saxena, J. Duro, A. Tiwari, K. Deb, and Q. Zhang. Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *Evolutionary Computation, IEEE Transactions on*, 17(1):77–99, 2013.

[29] H. Singh, A. Isaacs, and T. Ray. A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *Evolutionary Computation, IEEE Transactions on*, 15(4):539–556, 2011.

[30] T. Tušar and B. Filipič. Visualizing 4d approximation sets of multiobjective optimizers with prosections. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 737–744, New York, NY, USA, 2011. ACM.

[31] D. Walker, R. Everson, and J. Fieldsend. Visualizing mutually nondominating solution sets in many-objective optimization. *Evolutionary Computation, IEEE Transactions on*, 17(2):165–184, 2013.

[32] R. Wang, R. Purshouse, and P. Fleming. Local preference-inspired co-evolutionary algorithms. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pages 513–520. ACM, 2012.

[33] U. Wickramasinghe and 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*, pages 667–674. ACM, 2009.

[34] J. Xiong, Y. Chen, K. Yang, Q. Zhao, and L. Xing. A hybrid multiobjective genetic algorithm for robust resource-constrained project scheduling with stochastic durations. *Mathematical Problems in Engineering*, 2012, 2012.

[35] M. Yoshimi, T. Kuhara, K. Nishimoto, M. Miki, and T. Hiroyasu. Visualization of pareto solutions by spherical self-organizing map and it's acceleration on a gpu. *Journal of Software Engineering and Applications*, 5(3), 2012.