Studying the influence of algorithmic parameters and instance characteristics on the performance of a multiobjective algorithm using the Promethee method

DOI: 10.1080/01969722.2019.1598705
Handle: http://hdl.handle.net/1942/29611

Studying the influence of algorithmic parameters and instance characteristics on the performance of a multi-objective algorithm using the PROMETHEE method

Jochen Janssens[1], Kenneth Sörensen[1], and Gerrit K. Janssens[2]

[1]University of Antwerp Operations Research Group ANT/OR, Prinsstraat 13, 2000 Antwerp, Belgium
[2]Research Group Logistics, Hasselt University, Diepenbeek, Belgium

March 2019

A method is proposed to analyse the effect of the algorithmic parameters and instance characteristics on the quality of a Pareto front produced by a multi-objective algorithm. This method is applied to a variable neighborhood tabu search that is used to solve a multi-objective microzone-based vehicle routing problem.

Our method can accommodate many different performance indices for Pareto fronts and uses the PROMETHEE multicriteria decision analysis method to select the best configuration based on the characteristics of the instance being solved.

Keywords: Multi-criteria decision making, PROMETHEE, GDSS.

1 Introduction

Researchers and practitioners in the field of evolutionary algorithms (EA) and heuristic optimization in general acknowledge that an appropriate structural configuration of an algorithm, as well as the setting of good parameter values is essential for an EA to perform well. This process is often called *tuning*. Yet, configuring an EA and setting its parameters in an optimal way is an optimization problem in itself, and is generally considered to be a significant challenge. Most algorithms can be configured in many different ways and/or feature a large number of parameters, each with many different possible

settings. Moreover, limited general knowledge has been obtained on the effect of EA parameters and configuration on the performance of the algorithm (Eiben and Smit, 2011a; Eiben and Smit, 2011b).

If tuning a single-objective algorithm is a non-trivial task, configuring the structure and setting the parameters of a *multi-objective algorithm* is even more challenging. Whereas a single-objective algorithm has the advantage that the solutions it finds under different parameter configurations can be trivially compared using their objective function, the same cannot be said of a multi-objective algorithm. Multi-objective algorithms produce a Pareto set of non-dominated solutions, and many different ways exist to assess the quality of such a Pareto front. As a result of this complexity, a relatively large number of papers have appeared on the subject of tuning single-objective algorithms, but contributions describing methods to set the parameters of multi-objective algorithms are few and far between.

As mentioned, there is no straightforward way to compare the quality of multiple Pareto sets, as there are different properties a Pareto set should have in order to be considered "good". A large number of papers have been dedicated to measuring the quality of a Pareto set, and several performance indices have been proposed as a result. Okabe et al. (2003) provides an overview of various performance indices (PIs) and categorises them in three classes: cardinality-based PIs, accuracy PIs, and distribution and spread PIs. The class of *cardinality-based* indices, in which the number of solutions in the Pareto set and dominance of solutions from one Pareto set over an other Pareto set are bundled. *Accuracy* indices that measure how close a Pareto set is to a theoretical Pareto front, or how much of the area of the solution space is dominated by a Pareto set. And finally, *distribution and spread* indices, that measure how well and evenly spread the solutions of a Pareto set are in the solution space.

An algorithm might not work well in the same configuration on all instances. In other words, the instance that is being solved might have a significant impact on the ideal parameter settings, producing Pareto sets that perform differently when measured on the various performance indices used. For example, large instances might require a large mutation rate to achieve an acceptable accuracy, whereas small instances might need a small mutation rate to achieve this. Therefore, not only the parameters of the algorithm should be studied, but also the properties of the instances, as well as the interaction between both.

In this paper, we propose a novel approach to optimize the performance of a multi-objective optimization approach, that determines the best parameter settings and configuration for different instance characteristics. It is is illustrated by applying it to the multi-objective variable neighborhood tabu search proposed in J. Janssens et al. (2015).

Our method makes use of a full-factorial experiment, in which the algorithm is executed for multiple

levels of the algorithm parameters on instances with varying characteristics. The experiment returns a Pareto set for each combination of parameter settings and instance. Several indices (we use three, i.e., one for each of the classes, in our experiments) are used to measure the quality of the Pareto sets.

Our method uses a well-known multi-criteria method, i.e., the PROMETHEE method of (Brans, Vincke, et al., 1986), to compare the Pareto sets obtained using different parameter settings. The PROMETHEE method returns a *net flow* for each of the Pareto sets. Statistical analysis (analysis of variance (ANOVA)) is then used to measure the influence of the parameter settings and their interaction effects with instance properties on this net flow. The results and conclusions from that analysis are compared with the results obtained by the PROMETHEE GDSS Procedure for verification purposes.

This paper is organized as follows. Section 2 reviews the literature on parameter tuning. Section 3 introduces metrics for the evaluation of the quality of Pareto sets. Section 4 reports on the Promethee method, more specifically the implementation which is used in this research. Section 5 briefly introduces how the concept of a Group Decision Support System can be integrated into our solution method. Section 6 applies the complete procedure to an application on vehicle routing with courier companies. Results are analysed in Section 7. The paper ends with some conclusions in Section 9.

2 Literature

While the early developers of EAs either chose parameter values at hoc or at random, or copied them from other articles, gradually some interest has grown in how to choose the right parameters in order to obtain good performance of a developed optimization algorithm. Hooker (1995) argues that researchers should focus on scientific testing rather than competitive testing. Esquivel et al. (2002) proposed that evolutionary computation techniques are very able to build well-delineated Pareto fronts in multi-objective optimization problems and applied it to the job shop scheduling problem. They stressed that good parameter settings can enhance the behavior of the evolutionary algorithm. This would lead to a sound understanding of why certain algorithms work better than others, rather than merely listing algorithms and their computational times. In Eiben and Smit (2011b), this leads to two perspectives of parameter tuning: configuring an evolutionary algorithm by choosing parameter values that optimise its performance, and analysing an evolutionary algorithm by studying how its performance depends on its parameter values. In their paper, a conceptual framework for parameter tuning is presented. They also classify tuning methods.

CALIBRE is proposed by Adenso-Diaz and Laguna (2006), and attempts to find the best values for up

to five search parameters associated with a procedure under study. It performs a systematic search for parameter values within specified ranges employing a measure of performance as a guiding mechanism. CALIBRE makes use of Taguchi's fractional factorial experimental designs, coupled with a local search procedure.

Instead of estimating the performance of an evolutionary algorithm for multiple parameter values or ranges of values, REVAC estimates the expected performance when parameter values are chosen from a probability density distribution with maximised Shannon entropy (Nannen and Eiben, 2007). This method is only appropriate for quantitative parameters (Eiben and Smit, 2011b).

Racing is a technique that tests a set of configurations in parallel, quickly discards those configurations that are clearly inferior and concentrates the computational effort on differentiating among the better configurations (Maron and Moore, 1997).

F-Race is a racing algorithm that starts by considering a number of candidate parameter settings and eliminates inferior ones as soon as enough statistical evidence arises against them (Balaprakash et al., 2007). An iterated F-Race is proposed in Birattari et al. (2010), and an implementation in R is presented in López-Ibáñez, Dubois-Lacoste, et al. (2016). López-Ibáñez and Stützle (2010) apply I/F-Race on a bi-objective TSP using both the hypervolume and the epsilon measures.

Sequential parameter optimization (SPO) (Bartz-Beielstein, Parsopoulos, et al., 2004; Bartz-Beielstein and Preuss, 2006) is an iterative model based approach that entails two phases. First a primary model is built, which is afterwards evaluated and improved in a second step, that is executed iteratively.

Many of the aforementioned methods only deal with single-objective optimization methods, but could possibly be extended to cope with multi-objective optimisation problems. Furthermore, most of the methods mentioned before deal with the solutions and their objective values directly rather than with quality of the Pareto set. The method proposed in this paper deals with indices that measure the quality of the Pareto set. To the best of our knowledge, only López-Ibáñez and Stützle (2010) and G. K. Janssens and Pangilinan (2010) took into account quality indices before.

Many real-world optimisation problems are multi-objective by nature and have objectives that are in conflict. Different strategies can be used to solve such optimization problems. Traditionally, popular strategies have combined all objectives in a single function, hence rendering the problem a single-objective one. These strategies are more commonly known as *aggregating functions* of *scalarizing strategies*. They are easy to implement, but several disadvantages have been noted: (1) these methods may miss some Pareto-optimal solutions, (2) they are influenced by the shape of the search spaces, and (3) they are time consuming because they should be performed in a series of different runs (with

different weights for the various objective functions) in order to obtain a set of Pareto-optimal solutions (Das and Dennis, 1997). Some scalarizing strategies are the *weighted sum*, *epsilon-constraint*, and goal attainment methods. On the other hand, decision making is applied to a set of solutions, generated by an optimization run. The optimal solution in this case is the Pareto-optimal set. However, the size of the Pareto-optimal set may be infinite in some instances and may be impossible to find even when it only consists of a finite number of solutions. In such cases, the preferred result is a subset of the Pareto-optimal set, called a *Pareto set approximation*. To generate such a Pareto set approximation, evolutionary algorithms have been named as natural candidates, since they naturally maintain a set of solutions throughout the search (Bosman and Thierens, 2003).

Zitzler, Laumanns, et al. (2004) stress the quality assessments of Pareto set approximations. In single-objective optimisation, quality can be defined by means of the objective function. While comparing several solutions in the presence of multiple optimisation criteria, the concept of Pareto dominance can be used, but mostly leads to solutions being incomparable, i.e. neither of them dominates the others. It becomes more complicated when sets of solutions are compared as one Pareto set completely dominating another is rare.

Deb (2001) states that there are two orthogonal goals for any multi-objective optimisation algorithm: (1) to identify solutions as close as possible to the true Pareto-optimal set and (2) to identify a diverse set of solutions distributed evenly across the entire Pareto-optimal front. This has led to the development of several metrics that characterise either closeness to the Pareto-optimal front, or diversity of the solutions, or both.

In our experiment, we select one measure from each of the classes, proposed by Okabe et al. (2003). The Set coverage metric is a cardinality-based metric which computes the relative spread of solutions between two non-dominated sets (Zitzler, 1999). The hypervolume metric is an accuracy metric which measures both closeness and diversity (Zitzler, 1999). Schott's spacing metric belongs to the class of distribution and spread metrics (Schott, 1995). It measures the diversity across the Pareto surface. Most measures are unary quality measures, i.e., the measure assigns to each Pareto set approximation a single number that reflects a certain quality aspect.

3 Metrics for the evaluation of the quality of Pareto sets

3.1 Set Coverage

Coverage (C), first proposed in Zitzler and Thiele (1998), is used to compare two sets to each other in terms of dominance, and falls in the class of cardinality based performance indices. A solution vector $\mathbf{a}$ is said to cover solution vector $\mathbf{b}$ if $\mathbf{a}$ weakly[1] dominates $\mathbf{b}$, notated as $\mathbf{a} \succeq \mathbf{b}$. The coverage of set S_1 over set S_2 is given by:

$$C(S_1, S_2) = |\{s_2 \in S_2; \exists s_1 \in S_1 : s1 \succeq s_2\}| \, / |S_2| \tag{1}$$

The reader should note that $C(S_1, S_2)$ does not have to be equal to $1 - C(S_2, S_1)$. The metric value $C(S_1, S_2) = 1$ means all members of S_2 are weakly dominated by S_1. On the other hand, $C(S_1, S_2) = 0$ means that no member of S_2 is weakly dominated by S_1.

Set coverage is a pairwise metric. The coverage of each Pareto front versus all other fronts is calculated. The metric values need to be aggregated into a single number, for example by averaging. The larger the value of the coverage metric, the better the Pareto front.

3.2 Hypervolume

As no optimal Pareto set is available, evaluating the quality of the obtained Pareto sets on accuracy can only be done by comparing them to each other. For each Pareto set we calculate the objective value space covered by the set of non-dominated solution vectors, refered to as hypervolume (see Zitzler and Thiele, 1998). The algorithm used in this paper to calculate the hypervolume for a Pareto set has been proposed by Fonseca et al. (2006). An example of a hyperarea, a special case of the hypervolume in two dimensions, is given in Fig. 1. The larger the hypervolume, the better the Pareto set.

The hypervolume metric is interesting because it is a unary metric which is sensitive both to the overall advancement of the non-dominated set and to the distribution of individual points across the set. The placement of the reference point r is critical and determines the sense and the magnitude of the hypervolume. Problems may appear if objectives have dissimilar scales or if some objectives are not bounded.

[1] A solution *weakly* dominates another if it is as least as good with respect to all objectives and better with respect to at least one objective.

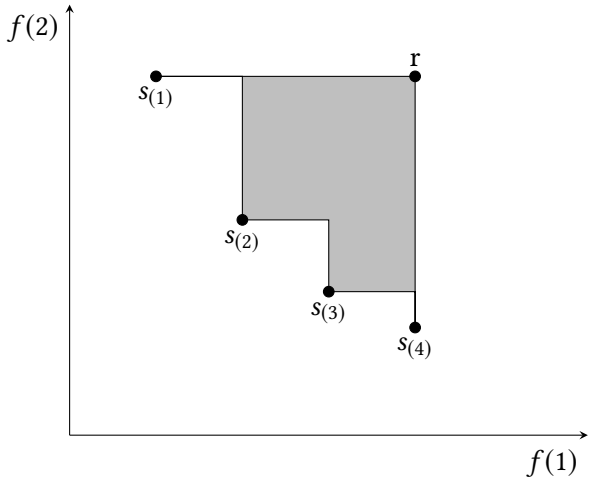

Figure 1: The hypervolume indicator in a two-objective case

3.3 Spacing

To evaluate the distribution of the solution vectors in the retrieved Pareto sets, spacing (SP) is used. It was introduced by Schott (1995). Schott's metric measures the diversity of a non-dominated set with a relative distance measure between consecutive solutions in the obtained non-dominated set S. It is calculated as

$$SP(S) = \sqrt{\frac{1}{|S-1|}\sum_{i=1}^{|S|}(d_i - \bar{d})^2} \tag{2}$$

$$d_i = \min_{s_k \in S \wedge s_k \neq s_i} \sum_{m=1}^{M} |f_m(s_i) - f_m(s_k)| \tag{3}$$

where m is the number of objective functions and $\bar{d}$ is the average of d_i. The distance measure is the minimum value of the sum of the absolute difference in objective function values between solution s_i and any other solution s_k in the obtained non-dominated set. In Okabe et al. (2003) it is pointed out that this measure should be used with some caution, as a large gap between solution vectors can bias the results. A value of zero means that all solution vectors are equally spaced. Schott's metric should be as small as possible.

4 The Promethee method

The Promethee method, introduced by Brans, Vincke, et al. (1986), is used in this paper to create a ranking of the alternative combinations of settings for the algorithmic parameters. Promethee is an outranking method in multiple-criteria decision analysis (MCDA), that originally was proposed in two forms. Promethee I, where a partial ranking of alternatives is created, and Promethee II, where a complete ranking is given as output. Two main parameters can be set by the decision maker to influence the behaviour of the ranking mechanism. For each criterion i, a weight π_i which is a measure of relative importance of that criterion, can be given. Next to the weight, a preference function for each criterion needs to be chosen. Six basic types of generalised criteria have been defined by Brans, Vincke, et al. (1986). For nearly all basic types of generalised criteria, additional parameters need to be defined such as an indifference zone and a preference zone, for which the decision maker is indifferent or has a strict preference for one solution over the other.

4.1 Promethee optional choices

The three metrics used in our study correspond to the criteria in the Promethee method. Criteria may have different weights corresponding to their importance, but we have chosen to use equal weights. Varying these weights can favour one index over the others, but that is out of the scope of this paper, and left to the decision maker. For all three metrics the third criterion proposed in Brans, Vincke, et al. (1986) has been selected. It is called the *criterion with linear preference* and requires one additional parameter p.

The preference function of action a with regard to b, $P(a, b)$, is a function of the difference between the objective values for two alternatives a and b, $f(a)$ and $f(b)$, so that we can write

$$P(a, b) = \mathcal{P}(f(a) - f(b)). \tag{4}$$

It has to be a non-decreasing function, equal to zero for negative values of $d = f(a) - f(d)$. In order to give a batter view on the indifference area, Brans and Mareschal (2005) introduce a function $H(d)$ which is directly related to the preference function P.

$$H(d) = \begin{cases} P(a, b) & \text{if } d \geq 0 \\ P(b, a) & \text{if } d \leq 0 \end{cases} \tag{5}$$

When d is defined as $d = f(a) - f(b)$, Eq. (6) defines the type 3 criterion, which is called the criterion

with linear preference. As long as d is lower than a value p, the preference of the decision maker increases linearly with d. If d becomes greater than p, we have a strict preference situation.

$$H(d) = \begin{cases} |d|/p & \text{if } |d| \leq p \\ 1 & \text{if } p < |d| \end{cases} \tag{6}$$

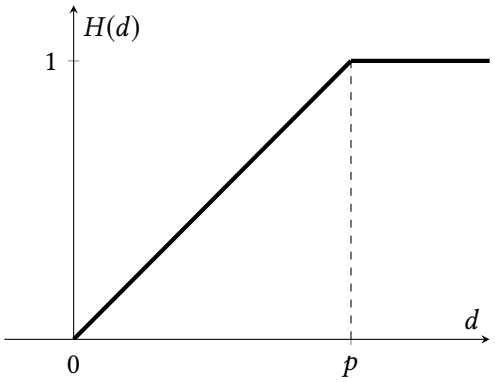

Figure 2: Linear preference and indifference area

4.2 Promethee algorithm

Following equation Eq. (4) a preference function value $H(d)$ of alternative a over alternative b is calculated for each criterion i and denoted as $P_i(a, b)$. The multicriteria preference index Π is defined as the weighted average over the preference functions P_i as defined in Eq. (6). The multi-criteria preference index Π is defined as the weighted average of the preference functions P_i as described in Eq. (7), where the weight π_i is a measure of the relative importance of criterion f_i. The Promethee rankings are obtained by means of the concepts of *flows*: an entering flow and a leaving flow. Both the entering flow, ϕ^-, and the leaving flow, ϕ^+, and their resulting net flow, ϕ, are calculated.

The entering flow is the sum of all multi-criteria preference indices (Π), of one alternative over all other alternatives in set $\mathcal{A}$. For more information on the the calculation of the preference index and the positive and negative flows, the reader is referred to Brans and Mareschal (2005) and Brans, Vincke, et al. (1986).

$$\Pi(a, b) = \frac{\sum_{i=1}^{k} \pi_i P_i(a, b)}{\sum_{i=1}^{k} \pi_i} \tag{7}$$

$$\phi^-(a) = \sum_{b \in \mathcal{A}} \Pi(b, a) \tag{8}$$

$$\phi^+(a) = \sum_{b \in \mathcal{A}} \Pi(a, b) \tag{9}$$

$$\phi(a) = \phi^+(a) - \phi^-(a) \tag{10}$$

5 The Promethee GDSS Procedure

Sections 3 and 4 offer a methodology to evaluate Pareto fronts based on a single instance. However, it is usually important that an algorithm performs well (and hence has a good set of parameters) for a wide variety of instances, rather than a single one. In general no parameter setting is expected to dominate all others for all instances, in which case one is generally interested in a compromise set, that offers a reasonable performance for many instances. This problem is similar to what is called a *group decision problem*. Such a problem occurs in real-life when different decision makers are involved in taking a decision, such as different departments in a company, different countries in an international organization, or different stakeholders in a project. To support this type of decision problem, supporting systems have been developed, for example to help with structuring the idea(s), the generation of alternative options, or the voting procedures. In the context of the Promethee method, Macharis et al. (1998) have developed a multi-criteria procedure for group decision support. This procedure describes a multi-step approach in which decision makers are in a room with computer support and a facilitator. While corresponding to our problem, we face a simpler problem as the decision makers in our problem do not set their own criteria weights

For each instance that is examined in this paper, the metaheuristic proposed in J. Janssens et al. (2015) is applied for the combinations of five settings for the tabu list parameter and fifteen options for the move order parameter, as explained in Section 7. The Pareto optimal sets, retrieved from the algorithm, are compared on three performance indices by the Promethee method. This gives us a list of net flows for each parameter combination for each instance.

The Promethee Group Decision Support System Procedure (see Brans and Mareschal, 2005) is used as a mechanism to retrieve an overall ranking. This approach is intended to be used when one has to deal with multiple decision makers, but could be utilised to find a global "best" parameter setting as well.

Three phases are described in this technique, for which phase I and phase II are the generation of the alternatives and criteria, and the individual evaluation by each decision maker (using Promethee) respectively.

The third phase entails the global evaluation which makes use of the individual evaluations, here the

net flows for each instance, as criteria. Each of them is assigned a weight, which in this paper is set equal to 1 for all instances. After that, the PROMETHEE method is applied to this matrix of alternatives, using the type 3 generalised criterion with strict preference $p = 2$.

6 The vehicle routing problem for courier companies

The approach is illustrated by means of a vehicle routing problem for courier companies. The geographical region for delivery is divided into small zones, called microzones. A microzone corresponds to a bin or container in which parcels can be dropped at arrival at the depot. To ensure stability in the planning, the preferred assignment of microzones to vehicles is fixed in a tactical plan. When an estimate of the workload in each microzone is available, the microzones are reassigned to the available vehicles, with a preference for the vehicle they are assigned to, in the tactical plan. The optimization problem, as it appears for this application, is formulated as a tri-objective vehicle routing problem. The three objectives include: (1) minimise the total transportation cost; (2) minimise the deviation from the tactical plan; and (3) minimise the workload imbalance. An effective metaheuristic for generating a Pareto front with non-dominated solutions has been formulated in J. Janssens et al. (2015). The approximation to the real but unknown Pareto front depends on the parameters which guide the heuristic.

The model in J. Janssens et al. (2015) assumes that a tactical plan is given, with known work times for the microzones and known microzone-vehicle allocations. A set of realistic test instances is generated. The test instances comprise: zones that are assigned to vehicles in a tactical plan, an initial routing for the tactical plan, a limit on the amount of work a vehicle (driver) can perform, the amount of work in the microzones and the coordinates of the microzones in the distribution zone.

The instances used in this paper, which can be found on `http://antor.uantwerpen.be/ZVRP`. For an overview of the naming syntax for the instances and the range of values from which the instance properties are drawn, the reader is referred to J. Janssens et al. (2015). The instance characteristics that are investigated include the number of microzones, the number of clusters and the deviation from the average workload. Ríos-Mercado (2016) have developed a metaheuristic (GRASP) to solve a commercial districting problem, which means to partition a set of basic units into a pre-specified number of districts in order to minimize a measure of territory dispersion. It refers to a real application for a beverage distribution company.

7 Algorithmic parameter values

The algorithm in the case study constructs a routing plan for the multi-objective VRP by iteratively applying a multi-neighborhood tabu search heuristic to a weighted sum of the three objective functions. By varying the weights of the weighted sum, the tabu search heuristic is forced to explore different areas of the search space. Fundamental decisions on this type of heuristics include a decision on the neighborhood moves to be used and in which sequence, as well as the size of the tabu list.

The experiment investigates two parameters of the algorithm: the tabu list size and move order/move inclusion (related to neighborhood search). The *tabu list size* ranges from a size of zero elements to eight elements. The parameter does not directly change the size in physical memory, but rather the number of iterations that each element remains in the tabu list for. At every iteration, one element is placed in the list and will remain there for a number of iterations equal to the tabu list size. This directly relates to the number of elements that is present in the tabu list.

The second parameter determines the *move order* to be used in the variable neighborhood tabu search. This move order defines the sequence of moves to be explored. The decision is made to have three consecutive moves at maximum, to keep the possible number of combinations to a manageable size. The move order does not necessarily comprise every move defined in the paper. All permutations without repetition of moves are examined. Each move is executed until no further improvement is found. This means that scheduling the same move twice in a row is not beneficial. The choice has been made to exclude any repetition of the moves, to reduce the number of combinations, even though executing some move x followed by move y followed by move x might yield a better result than only executing move x followed by move y. If all moves are included, six possible combinations exist. If only two moves are executed (one move is excluded), there are six more combinations. If only one move is used, three combinations are possible. This gives a total of 15 possible move orders, in comparison to the 81 ($4^3 - 1 + 4^2 - 1 + 4^1 - 1$) combinations, without restriction on repeated moves. An overview of the move orders tested in the experiment is shown in Table 1.

8 Analysis of the results

A full factorial experiment has been set up and executed, in which the algorithm described in J. Janssens et al. (2015) is run for every possible parameter combination (nine values of the tabu list size and 15 values of the move sequences as described in Section 7) on all instances, multiple times.

For each of the resulting Pareto fronts, the three performance indices are calculated. The reference

Table 1: Move order

Move order ID	Order of moves
0	2-opt – swap – relocate
1	2-opt – relocate – swap
2	swap – 2-opt – relocate
3	swap – relocate – 2-opt
4	relocate – 2-opt – swap
5	relocate – swap – 2-opt
6	2-opt – swap
7	2-opt – relocate
8	swap – 2-opt
9	swap – relocate
10	relocate – 2-opt
11	relocate – swap
12	2-opt
13	swap
14	relocate

point for the hypervolume and the pairwise coverage is calculated for each instance separately. This is done to avoid any bias based on the instance characteristics. E.g., if one instance features microzones that are located in close proximity, solutions for such an instance will naturally have longer driving distances than those for an instance in which microzones are spread out in a broad area. In such a situation, a common reference point for the hypervolume and the pairwise coverage makes little sense.

On the resulting performance indices, the PROMETHEE method is applied on a per instance per run (of the full factorial experiment) basis. The net flows retrieved from PROMETHEE are analysed statistically with ANOVA to identify which parameters or instance characteristics have a significant impact on the resulting net flows.

The statistical analysis starts by fitting a model with up to third order interactions, which returns a model with a reasonable fit (adjusted $R^2 = 0.925$), but the high number of large (> 0.10) p-values makes it clear that the model has many insignificant terms. Starting from this model, a stepwise regression was performed based on the Akaike Information Criterion (AIC) to eliminate unnecessary terms. By a combination of stepwise regression and the removal of remaining terms with a p-value larger than 0.05, a model with an intercept, nine significant effect terms and six non-significant effect terms with an adjusted $R^2 = 0.928$ was obtained.

The main effects that are not significant, but are part of a significant interaction effect were left in the model. The decision was made to only analyse main effects and second order interaction effects as no third-order interaction effects were statistically significant. The main effects ‘move order’ and ‘tabu list

size' and second order interaction effects 'number of microzones:move order', 'number of clusters:move order', 'number of clusters:tabu list size' and 'workload deviation:move order' are further investigated. The distribution of the residuals is shown in Fig. 3. As the points in this residual plot are randomly dispersed around the horizontal axis, the regression model is appropriate for the data.

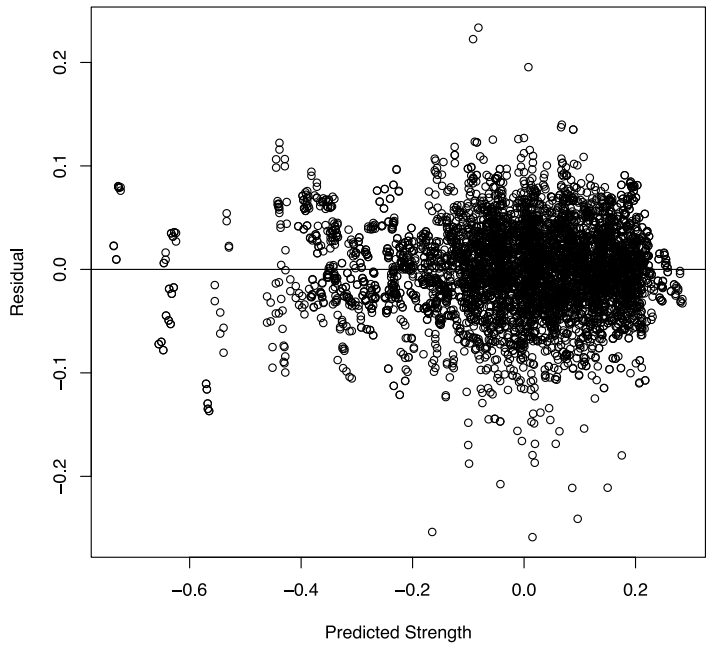

Figure 3: Residual plot for the fitted model

The Tukey HSD (Honest Significant Difference) test for the parameter tabu list size, which is plotted in Fig. 4, reveals that there is a significant difference between tabu list size equal to eight and every other tabu list size, and between tabu list size six and zero (the 95% confidence interval does not contain 0). If the box plot of the tabu list size is examined, verification can be made that the mean net flow for tabu list size eight is slightly lower than the the other mean values. The plots of the interaction effect between tabu list size and number of clusters is shown in Fig. 5 and Fig. 6.

The main effect 'move order' is also significant. Fig. 7 demonstrates that there are large differences in the net flows obtained using the different move orders.

The move order has a significant impact in several interaction effects. The same three move orders are being ranked the highest consistently. From the Tukey HSD test, it can be concluded that no significant difference exists between the mean net flow for these move orders. Move orders 4, 5 and 11 take the highest ranking places for workload deviation, number of clusters and number of microzones (see Fig. 8, 9 and 10)

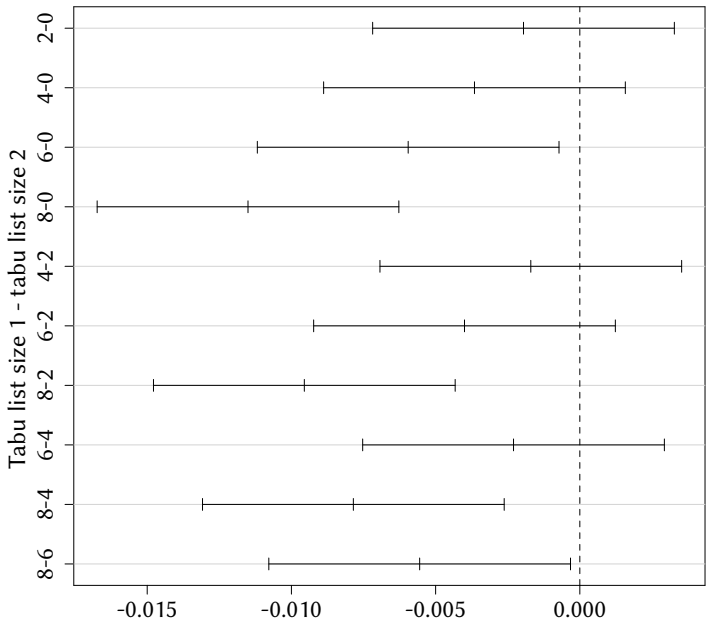

Figure 4: Tukey HSD 95% confidence intervals on the difference in net flows for different tabu list sizes

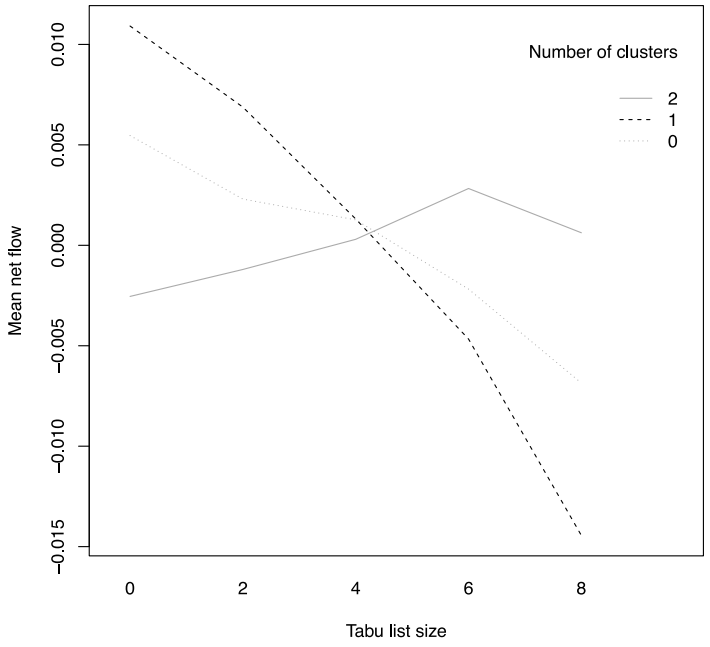

Figure 5: Interaction plot of number of clusters and tabu list size

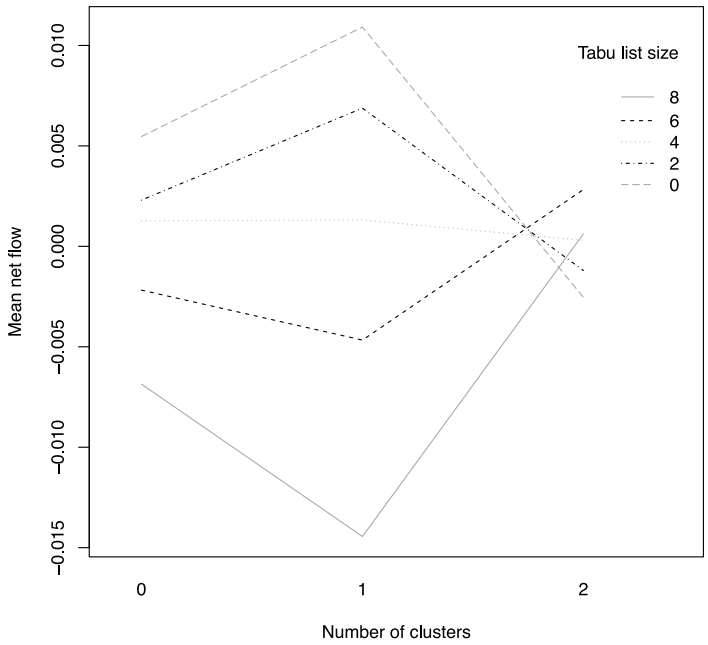

Figure 6: Interaction plot of tabu list size and number of clusters

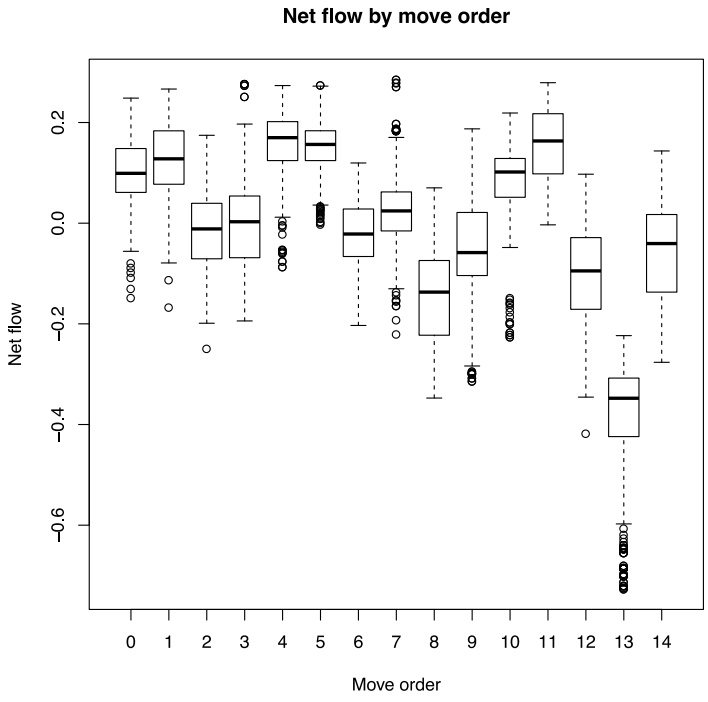

Figure 7: Box plot of move order

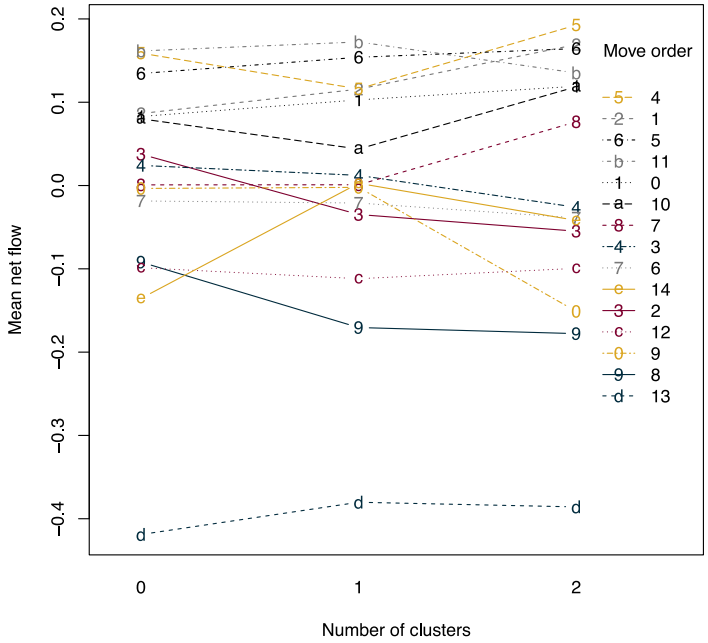

Figure 8: Interaction plot of number of clusters and move order

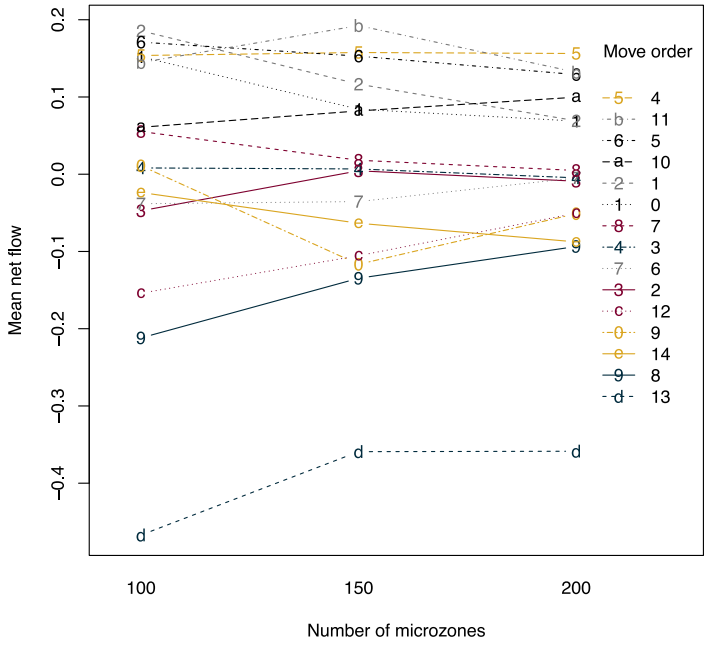

Figure 9: Interaction plot of number of microzones and move order

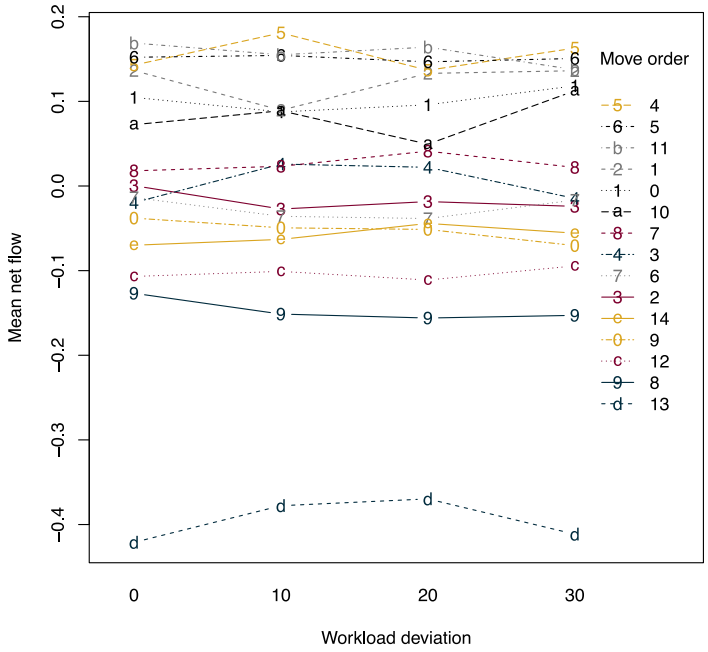

Figure 10: Interaction plot of workload deviation and move order

Furthermore, move order 1 always ranks in the top 5. The four move orders (1, 4, 5, and 11) seem to have in common that the relocate move is directly or indirectly followed by the swap move. The presence and order of the 2-opt move seems to be of less importance. Further experiments should indicate if this effect is reproducible, and what is causing the 2-opt move to be of less importance to the quality of the Pareto front. The poorest performing move order seems to be number 13, which uses only the swap move. No other clear trend seems to be present in the interaction plots.

The PROMETHEE GDSS procedure was also run on the PROMETHEE net flows. The top 20 ranked alternatives are shown in Table 2. The results found by the GDSS procedure are consistent with our analysis with ANOVA. The highest ranking move orders are 1,4,5 and 11, and tabu list sizes 0,2,4 and 6 have a slight edge on tabu list size 8. Of course this is a general ranking, and specific instance properties might give a different ranking as described above.

9 Conclusions

In this paper, we have presented a methodology to study the effects of the algorithmic parameters of a multi-objective algorithm on the quality of the produced Pareto fronts. Our methodology is based on the PROMETHEE method, a well-known multi-criteria decision making method, and allows to identify

Table 2: Top 20 ranking net flows for the GDSS procedure

Move order	Tabu list size	PROMETHEE net flow
11	2	0.0828
4	2	0.0821
4	0	0.0818
11	4	0.0809
4	4	0.0802
5	0	0.0799
11	0	0.0798
11	6	0.0788
4	6	0.0785
5	2	0.0785
5	4	0.0765
5	6	0.0758
11	8	0.0738
4	8	0.0723
5	8	0.0718
1	8	0.0641
1	6	0.0633
1	0	0.063
1	4	0.0624
1	2	0.061

which parameter settings and which instance characteristics influence the quality of the Pareto fronts produced by a multi-objective algorithm. Our method works by running a statistical experiment that applies the multi-objective algorithm on different instances and with different parameter levels to obtain a set of Pareto fronts. Performance indices for the resulting Pareto fronts are calculated and passed to the PROMETHEE method to retrieve a ranking per instance and per run of the experiment. This ranking is than analysed with ANOVA to determine which parameters or instance characteristics have a statistically significant effect, and what the possible interactions are. Additionally,the PROMETHEE Group Decision Support System Procedure can be used to determine a global best parameter setting across all instances.

As an example, we have analysed the performance of the multi-objective algorithm proposed in J. Janssens et al. (2015). The result of applying our methodology shows that the move order has an important impact on the ranking of the Pareto fronts in the PROMETHEE method. This indicates that the move order is largely responsible for the quality of the Pareto front found by the algorithm. Four instances of move order where consistently ranked in the top 5 of highest mean net flow for the characteristics of the test instances. This indicates that even if the specifications of the instances change, these move orders seem to deliver high quality solutions and Pareto fronts. The swap move alone

performs poorly in terms of Pareto front quality, and is ranked very low. No further important effects could be found by visual inspection of the interaction plots for move order and instance characteristics. Secondly, setting the tabu list size to 8 has a negative impact on the quality of the Pareto front as it results in a lower mean net flow than the rest of the tabu list sizes.

The methodology proposed in this paper can be applied to any multi-objective algorithm, and should allow for a more systematic way of setting the parameters of such an algorithm than is commonly done. The analysis of our methodology when applied to different multi-objective algorithms for different problems is therefore the main line of future research.

Acknowledgements

This research is supported by the Interuniversity Attraction Poles (IAP) Programme initiated by the Belgian Science Policy Office (COMEX Project).

References

Adenso-Diaz, B. and M. Laguna (2006). "Fine-tuning of algorithms using fractional experimental designs and local search". In: *Operations Research* 54.1, pp. 99–114.

Balaprakash, P., M. Birattari, and T. Stützle (2007). "Improvement strategies for the F-Race algorithm: Sampling design and iterative refinement". In: *Hybrid Metaheuristics*. Springer, pp. 108–122.

Bartz-Beielstein, T., K. E. Parsopoulos, and M. N. Vrahatis (2004). "Analysis of Particle Swarm Optimization Using Computational Statistics". In: *Proceedings of the International Conference Numerical Analysis and Applied Mathematics (ICNAAM)*. Ed. by T. E. Simos and C. Tsitouras. Weinheim, Germany: Wiley-VCH, pp. 34–37.

Bartz-Beielstein, T. and M. Preuss (2006). "Considerations of Budget Allocation for Sequential Parameter Optimization (SPO)". In: *Proceedings of Workshop on Empirical Methods for the Analysis of Algorithms*. Ed. by L. Paquete, M. Chiarandini, and D. Basso, pp. 35–40.

Birattari, M., Z. Yuan, P. Balaprakash, and T. Stützle (2010). "F-Race and iterated F-Race: An overview". In: *Experimental methods for the analysis of optimization algorithms*. Springer, pp. 311–336.

Bosman, P. and D. Thierens (2003). "A balance between proximity and diversity in multiobjective evolutionary algorithms". In: *IEEE Transactions on Evolutionary Computation* 7.2, pp. 174–188.

Brans, J.-P. and B. Mareschal (2005). "Promethee Methods". In: *Multiple criteria decision analysis: state of the art surveys*. Vol. 78. New York, NY: Springer Science & Business Media, pp. 163–186.

Brans, J.-P., P. Vincke, and B. Mareschal (1986). "How to select and how to rank projects: The Promethee method". In: *European Journal of Operational Research* 24.2, pp. 228–238.

Das, I. and J. Dennis (1997). "A closer look at the drawbacks of minimizing weighted sums of objectives for Pareto set generation in multi-criteria optimization problems". In: *Structural Optimization* 14.1, pp. 63–69.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Vol. 16. Wiley Interscience Series in Systems and Optimization. John Wiley & Sons.

Eiben, A. and S. Smit (2011a). "Evolutionary algorithm parameters and methods to tune them". In: *Autonomous search*. Springer, pp. 15–36.

– (2011b). "Parameter tuning for configuring and analyzing evolutionary algorithms". In: *Swarm and Evolutionary Computation* 1.1, pp. 19–31.

Esquivel, S. C., S. Ferrero, and R. H. Gallard (2002). "Parameter settings and representations in Pareto-based optimization for job shop scheduling". In: *Cybernetics &Systems* 33.6, pp. 559–578.

Fonseca, C. M., L. Paquete, and M. López-Ibáñez (2006). "An improved dimension-sweep algorithm for the hypervolume indicator". In: *IEEE Congress on Evolutionary Computation, 2006. CEC 2006*. IEEE, pp. 1157–1163.

Hooker, J. N. (1995). "Testing heuristics: We have it all wrong". In: *Journal of Heuristics* 1.1, pp. 33–42.

Janssens, G. K. and J. M. Pangilinan (2010). "Multiple Criteria Performance Analysis of Non-dominated Sets Obtained by Multi-objective Evolutionary Algorithms for Optimisation". In: *Artificial Intelligence Applications and Innovations: Proceedings of the 6th International Federation for Information Processing WG 12.5 International Conference, AIAI 2010, Larnaca, Cyprus, October 6-7, 2010*. Berlin, Heidelberg: Springer, pp. 94–103.

Janssens, J., J. Van den Bergh, K. Sörensen, and D. Cattrysse (2015). "Multi-objective microzone-based vehicle routing for courier companies: From tactical to operational planning". In: *European Journal of Operational Research* 242.1, pp. 222–231.

López-Ibáñez, M., J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle (2016). "The irace package: Iterated racing for automatic algorithm configuration". In: *Operations Research Perspectives* 3, pp. 43–58.

López-Ibáñez, M. and T. Stützle (2010). "Automatic configuration of multi-objective ACO algorithms". In: *Swarm Intelligence*. Springer, pp. 95–106.

Macharis, C., J.-P. Brans, and B. Mareschal (1998). "The GDSS promethee procedure". In: *Journal of decision systems* 7.4, pp. 283–307.

Maron, O. and A. W. Moore (1997). "The racing algorithm: Model selection for lazy learners". In: *Lazy learning*. Springer, pp. 193–225.

Nannen, V. and A. Eiben (2007). "Relevance Estimation and Value Calibration of Evolutionary Algorithm Parameters." In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. IJCAI'07. Hyderabad, India: Morgan Kaufmann Publishers Inc., pp. 975–980.

Okabe, T., Y. Jin, and B. Sendhoff (2003). "A critical survey of performance indices for multi-objective optimisation". In: *The 2003 Congress on Evolutionary Computation, CEC '03*. Vol. 2. IEEE, pp. 878–885.

Ríos-Mercado, R. Z. (2016). "Assessing a Metaheuristic for Large-Scale Commercial Districting". In: *Cybernetics and Systems* 47.4, pp. 321–338.

Schott, J. R. (1995). *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*. Tech. rep. DTIC Document.

Zitzler, E. (1999). "Evolutionary algorithms for multiobjective optimization: Methods and applications". PhD thesis. Zürich, Switzerland: Eidgenössische Technische Hochschule Zürich (ETH).

Zitzler, E., M. Laumanns, and S. Bleuler (2004). "A tutorial on evolutionary multiobjective optimization". In: *Metaheuristics for multiobjective optimisation*. Ed. by X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt. Vol. 535. Lecture Notes in Economics and Mathematical Systems. Springer, pp. 3–37.

Zitzler, E. and L. Thiele (1998). "Multiobjective optimization using evolutionary algorithms — A comparative case study". In: *Parallel Problem Solving from Nature — PPSN V*. Ed. by A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel. Vol. 1498. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 292–301.