

Experiments on Metaheuristics: Methodological Overview and Open Issues

Marco Chiarandini¹, Luís Paquete^{2,3}, Mike Preuss⁴, and Enda Ridge⁵

¹ Department of Mathematics and Computer Science,
University of Southern Denmark, Odense, Denmark

`marco@imada.sdu.dk`

² Faculty of Economics

³ CSI – Centre for Intelligent Systems
University of Algarve, Faro, Portugal

`lpaquete@ualg.pt`

⁴ Chair of Algorithm Engineering
University of Dortmund, Dortmund, Germany

`mike.preuss@uni-dortmund.de`

⁵ Department of Computer Science
The University of York, York, U.K.

`eridge@cs.york.ac.uk`

Abstract. Metaheuristics are a wide class of solution methods that have been successfully applied to many optimization problems. The assessment of these methods is commonly based on experimental analysis but the lack of a methodology in these analyses limits the scientific value of their results. In this paper we formalize different scenarios for the analysis and comparison of metaheuristics by experimentation. For each scenario we give pointers to the existing statistical methodology for carrying out a sound analysis. Finally, we provide a set of open issues and further research directions.

1 Introduction

Metaheuristics are generated by the assemblage of search methods such as construction heuristics, local search and more general guidance criteria to solve a specific problem. Despite the lack of theoretical foundation, their simplicity has attracted many researchers and practitioners. Many results in the literature indicate that metaheuristics are the state-of-the-art techniques for problems for which there is no efficient algorithm. However, for many problems, metaheuristics do not always reach an optimal solution, even for long computation times. In addition, it is often impossible to obtain an analytical prediction of either the solution achievable within a given computation time or the time taken to find a solution of a given quality. The assessment of these conflicting performance measures is critical for the evaluation of metaheuristics [2] and their application to real problems. Given the lack of theoretical guidelines and the stochastic nature of most metaheuristics, such performance assessments are best carried out by experimentation.

Measures of performance such as the *solution-cost* and *run-time* can be seen as random variables. The field of *statistics* therefore provides the appropriate basis for supporting the research on metaheuristics. Statistics offers the advantage of providing i) a systematic framework (the design of experiments) for the collection and evaluation of data, thus maximizing the objectivity and replicability of experiments; ii) a mathematical foundation (the statistical analysis) that supplies a probabilistic measure of events on the basis of inference from the empirical data. Moreover, the use of statistical tools encourages and supports a methodical approach to experimentation. Several possible use of statistical tools in the study of algorithms and heuristics have been well illustrated in [10]. Nevertheless, in the field of metaheuristics, we still note some reluctance in conducting well designed experiments.

In this paper, we review different scenarios in the assessment of metaheuristics and outline the statistically-oriented methodologies for their analysis. In doing this, we extend the cases discussed in [10] to other cases which are typical of the studies on metaheuristics.

At the highest level, we distinguish two models of analysis:

1. the *univariate model*, in which either solution-cost or run-time is taken into account;
2. the *multivariate model*, in which both solution-cost and run-time are of interest.

The first model has long been the more typical in assessing metaheuristics. There is a considerable amount of literature on the methods for its analysis. The second model, although providing a deeper insight into the analysis, has not yet been accurately and explicitly addressed, probably due to its higher degree of complexity. In this case, some of the methods may be adapted from *multivariate statistics*, which is a well-developed field [1]. *Random set theory* might also provide the building blocks for a methodological approach [17,15]. However, these links have not yet been thoroughly explored. Moreover, there remain cases for which it is not yet clear whether mathematically founded methods of analysis have been developed at all. Our goal in this paper is to formalize the scenarios of metaheuristic analysis and put them in the context of existing statistical methods, while pointing out where there is need for further research and interdisciplinary development.

2 The Univariate Model

In this case, the researcher or practitioner is interested in either solution-cost (e.g., in a minimization problem) or run-time⁶ as performance measure. If the concern is solution-cost, we assume that equal computational resources are allocated to the different algorithms in the study (*fairness principle* [27]). If the concern is run-time, time is measured when a solution with desired properties is found.

⁶ We assume that run-time corresponds to the number of operations performed based on some cost model that is related to the CPU time, or simply wall clock time.

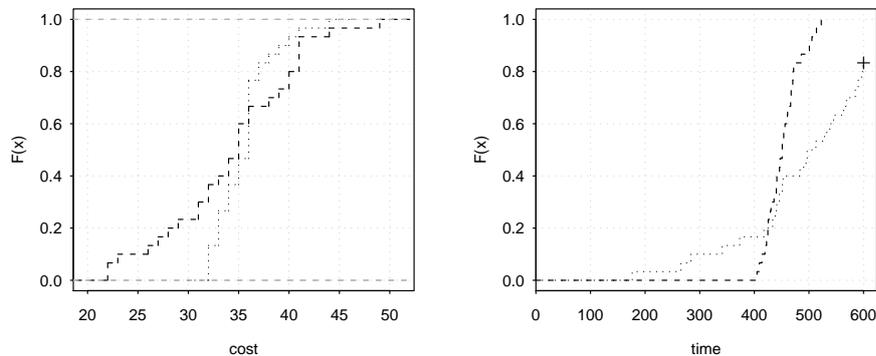


Fig. 1. ECDFs of two metaheuristics on the graph coloring problem. On the left we measure the solution cost in terms of number of colors which are to be minimized, on the right we measure the computational time to find a solution with the minimal number of colors. A dashed line and a dotted line are used to distinguish between different algorithms.

Characterization The performance measure X (solution-cost or run-time) of a metaheuristic on a single instance can be described by its *probability distribution* $p(x) = \Pr[X = x]$ or equivalently by its *cumulative distribution function*⁷

$$F(x) = \Pr[X \leq x] = \sum_{x_i \leq x} p(x_i). \quad (1)$$

Alternatively, if the probability distribution is known, few *parameters*, e.g., the mean and variance, may be enough to represent it.

In experiments on metaheuristics, we observe data X_1, \dots, X_n sampled from the distributions above. It is then possible to derive the *empirical cumulative distribution function* (ECDF) as follows

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I(X_i \leq x) \quad (2)$$

where $I(\cdot)$ denotes the indicator function. Note that the formula is general and holds both for *uncensored* as well as for *censored data*. In experiments on metaheuristics which consider run-time, sample data may be censored if a time limit (imposed for practical reasons) is reached before a local optimum or a solution with certain properties is found.

In Figure 1 we show examples of ECDFs for metaheuristics on the graph coloring problem. On the left two metaheuristics are run 30 times on the same

⁷ We restrict the notation to the discrete case, which is often the case of the existent experimental research.

instance and the depicted ECDFs represent the frequency with which each algorithm attains coloring at least as good as the colors indicated on the x -axis. On the right two metaheuristics are left running 600 seconds on the same instance recording the time when the chromatic number is found. In this case, the ECDFs represent the frequency with which a solution is found within the time indicated. The fact that one ECDF is truncated (i.e., it does not reach 1) indicates that the algorithm was not able to solve the instance within the given time.

Usually, the performance assessment of a metaheuristic is carried out on a representative sample of a *class of instances*. In this context, a performance measure X of a metaheuristic that is applied to a class of instances Π can be described by the following probability distribution [6,23]

$$p(x) = \sum_{\pi \in \Pi} p(x|\pi)p(\pi). \quad (3)$$

In practice, instances in the class may have different probabilities to appear and, hence, the term $p(\pi)$ has an influence in the analysis. If instances are instead equally likely to appear, the term $p(\pi)$ is a constant and may be neglected. The way to deal with $p(\pi)$ is in the instance sampling process. Here we assume for the sake of simplicity that instances are sampled from the class Π with equal probability.

Summary measures for sampled data are divided into measures of location, such as the sample mean and q -quantiles, and measures of dispersion, such as the sample variance and the standard variation. Clearly, summary measures tend to hide part of the information contained in the sample data. Often histograms, boxplots and ECDF plots are used to provide a more complete view of the data.

Few computational studies focus on the characterisation of metaheuristic performance, i.e., the nature of the distribution $p(x)$. On the side of run-time, some links have been explored with a branch of statistics called *survival analysis* that deals with *time-to-event* models. It was shown that ECDFs of run-time obtained by high-performance metaheuristics are often close to being exponentially distributed [19]. On the side of solution-cost, some research has used models from *extreme value theory* to support the conclusion that ECDFs are well approximated by Weibull distributions [25].

Analysis Most of the literature on metaheuristics has focused on experimental comparisons. In this case, the use of descriptive statistics, such as the sample mean and the standard deviation, is not sufficient. Inferential statistics must also be used to check that the sampled data are enough to claim that the differences observed can be generalized to the population distributions. Statistical tests are used to make these statements objective by checking whether a standard level of confidence is present in the data. If the test does not allow to reject the absence of differences, the researcher must either collect more data in order to increase the power of the test and detect also small differences or he must stop if differences become irrelevant in practical terms.

There are two kinds of statistical inferential tests: parametric and non-parametric. The majority of parametric tests are based on the central assumption of *normally distributed* data. The studies mentioned above on the nature of $p(x)$ for both solution-cost and run-time seem to indicate that this assumption is not met in the analysis of metaheuristics, in which case, distributions are unknown. However, this does not rule out the use of parametric statistics. It is well known that some tests, like those based on the F -ratio in ANOVA, are very robust to deviations from the normality assumption, especially for large data sets. Other techniques such as the transformation of data (logarithm, inverse and square root) can help data to meet the normality assumption. Non-parametric tests, such as rank-based tests [32] and permutation tests [26], remove this assumption. Yet, these methods are less developed and less powerful than the parametric tests.

The starting case discussed in any text book of statistics is the *two sample case*. In experiments on metaheuristics it is rarely the case that the comparison involves only two alternatives, but this starting example serves us to clarify a few basic concepts, in particular, the distinction between *unreplicated case* and *replicated case*. In the unreplicated case, two metaheuristics are run once on n instances. Since metaheuristics are randomized, it is recommended to adopt a basic variance reduction technique and run both algorithms with the same random seed on the same instance. Single instances may be treated as levels of a *blocking factor* [27]. In the two sample case, this leads to the use of tests in their *matched pairs* form. The possible tests are, in the parametric case, the *t-test*⁸ and, in the non-parametric case, the *binomial test* (if ties are not possible) and the *Wilcoxon signed rank test* [32]. Alternatively, permutation tests may be used to generate the distribution of the test statistic from permutations of the sampled data [26].

In the replicated case, two metaheuristics are run r times on n instances. If the experiments are conducted by blocking both on instances and on random seed then the same tests used in the previous unreplicated case can be applied. Alternatively, other specific tests are available such as the parametric *two-way ANOVA* (with blocking) or the non-parametric *Kruskal-Wallis rank sum test* [32]. However, if the overall total number of experiments is fixed, the unreplicated design is preferable [6].

More accurate tests compare the ECDFs of the two algorithms. The *Kolmogorov-Smirnov* (KS) test statistic considers the maximal difference between the two ECDF curves, and derives the distribution of this statistic by permutation methods [12]. This test is able to identify more general differences than location differences (mean or median). In particular, the test can also be used to determine whether there exists *statistical dominance* between the two curves [12].

Note that in the absence of statistical dominance and equal variance, every test above must be used with caution because it is not trivial to decide which

⁸ More specifically, the t-test for differences in means should be used under the assumptions of variances unknown and not equal (some statistical software refer to this as the Welch form of the t-test).

distribution we may prefer. Consider, for example, the two metaheuristics whose ECDF is represented in Figure 1, left. Studying only means we would choose the algorithm represented by the dashed line but the variance of its performance is clearly larger than the other algorithm represented by the dotted line and we might prefer for our application an algorithm whose performances are more certainly described. Moreover, some additional care must be taken in the presence of censored data. In this case, the statistical tests mentioned above are no longer appropriate for testing and the suitability of *bootstrap* methods [11] or tests used in survival analysis [18] might need to be investigated.

Dealing with metaheuristics, we are however often faced with more complex designs than these simple ones. We need indeed to compare several possible configurations arising from the combination of different factors (i.e., metaheuristic components and metaheuristic parameters). By applying any of the aforementioned tests c times, the effective level of confidence (the error committed in rejecting the hypothesis that the two algorithms have equal performance) of the overall test procedure becomes $\alpha_{EX} = 1 - (1 - \alpha)^c$ [20]. For example, setting α for each single test at the usual value of 0.05 and in presence of 3 comparisons, $c = 3$, then α_{EX} would grow to 0.14. A common procedure to control α_{EX} is to perform first an analysis of variance to identify whether there is at least one factor that exhibits significant difference and then to proceed to *post-hoc* multiple comparisons by adjusting (or not) the α value.

In the parametric case, the analysis of variance is carried out with the well-known ANOVA in its one- or multiple-way forms, depending on the presence of one or more algorithmic factors under analysis [24]. The appropriate methods for *post-hoc* analysis are methods for *all-pairwise comparisons* or *multiple comparisons* with the best [20]. In the latter case the *Tukey honest significant difference method* is appropriate. Alternatively, it is possible to use an *all-pairwise t-test* (also referred to as Fisher least significant difference method) with an adjustment of the α value. Among the adjustment methods, the basic Bonferroni's method is quite conservative while the Holm's procedure exhibits higher statistical power and should be therefore preferred. In the non-parametric case, the *Friedman rank-based test* and extensions thereof allow to determine differences in a single factor with blocking scenario, although the adjustment issue is more controversial in this context [12,32]. More complex scenarios with multiple factors of interest must be reconducted under the single factor scenarios. Extensions of the KS test to study differences among several ECDFs are also available. In particular we mention the *Birnbaum-Hall test* [12].

2.1 Advanced Topics

Recently, more advanced topics of statistics have been used to analyse and compare metaheuristics. These are regression trees, Design of Experiments (DOE) and sequential testing through fine-tuning algorithms.

Regression trees Regression trees are simple hierarchical models for grouping the available samples of a system according to the most important variable

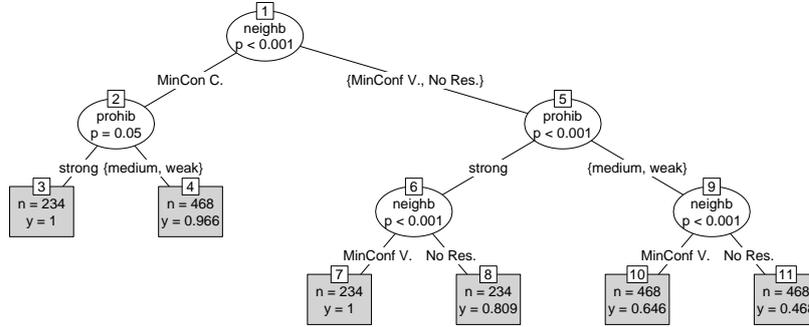


Fig. 2. An example of regression tree analysis for Tabu Search components on the graph coloring problem. The importance of the effect of factors on the performance is recognizable from the level in the tree where the relative branching occurs. In this analysis the two numerical parameters do not yield any branching meaning that their effect is negligible.

value ranges. This is especially helpful if quantitative and qualitative variables are contained in the system, as it is often the case if metaheuristic parameters are considered. Both are matched onto decision tree splits within the same binary tree. The importance of a variable or value range of a variable directly corresponds to the level of the nodes containing it in its decision criterion. A weakness of this technique is that variable interactions are not considered in the linear models that determine the splits. The sample set is partitioned into axis parallel rectangles in the coordinate system of the variables. *CART, classification and regression trees*, a standard method, dates back to [7].

An example of the output of this analysis is given in Figure 2. The data are extracted by an experiment designed to assess the contribution of components in a Tabu Search metaheuristic for the graph coloring problem. Factors of the study were: three strategies to restrict the neighborhood, three prohibition mechanisms in the definition of a tabu move and two numerical parameters for the definition of the tabu length. All these Tabu Search instantiations were run once with the same time limit on 30 uniform graphs of the same size. In the figure, the performance measure y corresponds to the number of colors normalized among the instances. The lower this measure is, the better the performance of the algorithms that belong to the root-leaf path is. The number of data n left after the partition is also reported. On the nodes, the p-value of the test that determines branching is also reported.

Designs and techniques for parameter screening and tuning The deployment of a metaheuristic usually involves setting the values of a large number

of parameters, whose interactions are usually unknown. This difficulty is further exacerbated by the possible interactions of some of these parameters with some characteristics of the problem instance. The challenge then is to: i) determine which algorithm parameters and problem characteristics have an effect on the responses; ii) model the relationship between the most important algorithm parameters, problem characteristics and responses; and iii) optimise the responses based on this relationship.

However, parameter settings are often chosen in an *ad-hoc* manner or quoted from the literature without any rigorous examination of their suitability. Simple factorial designs are inappropriate. They require testing all levels of all factors⁹ with one another. For a tuning problem of 10 parameters and 2 problem characteristics, tested at the minimum of 2 levels each, this would require a prohibitive $2^{12} = 4096$ design points. *Fractional factorial designs* (FFD) offer a manageable alternative, which uses some subset of a factorial design's runs. This subset can be chosen so that main effects and some lower order interactions can still be determined but higher order interactions are *aliased* with one another. The assumption in using a fractional factorial design is that higher order interactions are likely to be of little consequence and so their aliasing can safely be ignored. Recent publications [28,29] illustrate the screening and tuning of metaheuristics with Design of Experiments techniques [24] such as *desirability functions* for multi-response optimisation and *overlay plots* for response robustness.

Sequential testing and racing One issue in all the tests described above is how many replicates are needed in order to distinguish differences. Increasing the number of replicates for each configuration decreases the outcome's sensitivity. At the same time, it increases the effort needed to adapt the parameters of the metaheuristic to the treated problem. It has long been common to solve this meta-optimization problem by means of a one factor at a time approach. However, several tuning methods based on the idea of *sequential testing* have emerged recently. We present two of these.

Racing algorithms, e.g., the F-Race [6], select the best out of a finite number of configurations (continuous parameters may be discretized) by running them several times and deleting the inferior ones by means of statistical tests as soon as significance arise. In each iteration, all remaining configurations are run on one out of a possibly infinite number of problem instances. The advantage is the reduction of the overall number of experiments to determine a single best configuration.

Sequential parameter optimization (SPO) [3] combines an underlying regression model and a stochastic process as correlation model utilized in DACE [30] with a simple variance reduction technique and the expected improvement heuristic [21]. During an SPO run, the number of replicates is subsequently incremented for the most successful configurations to reduce error probabilities. Next to a best

⁹ Here the term *factor* covers both metaheuristic tuning parameters and problem instance characteristics.

configuration, the internal model of SPO may also be used for algorithm analysis purposes (parameter interactions, etc.). A crossover of the two approaches, extending SPO by means of automatic hypothesis testing as used in the F-Race, is suggested in [4].

3 The Multivariate Model

In the analysis of metaheuristics for optimization problems the univariate case may be oversimplified. A thorough understanding of the performance of a metaheuristic should include indeed both solution-cost and run-time. In this case, the analysis falls into the scope of *Multivariate statistics*. We distinguish two specific scenarios under the multivariate model that may be of interest for the researcher:

- *Scenario 1*: study of solution cost and run-time *when* a certain termination criterion is reached, that is, the metaheuristic terminates *naturally*; therefore, each run of a metaheuristic is represented by a *point* in the plane *solution-cost* \times *run-time*;
- *Scenario 2*: study of solution cost and run-time *during* the run of the algorithm *until* a certain termination condition is reached; hence, each run of an algorithm is characterized by a *set of points* in the plane *solution-cost* \times *run-time*..

3.1 Scenario 1

A typical example under this scenario is the study of construction heuristics. They terminate when a complete solution has been produced. The interest is in determining which heuristic returned the best solution *and* was the fastest.

Characterization Let $\mathbf{X} \in \mathbb{R}^2$ denote now the bivariate performance measure of solution-cost and run-time. Its distribution function is defined by

$$F(\mathbf{x}) = \Pr[\mathbf{X} \leq \mathbf{x}]$$

where \leq denotes the weak component-wise order in \mathbb{R}^2 . $F(\mathbf{x})$ gives the probability that a metaheuristic finds a given solution cost within a given run-time. The estimation of this probability is obtained from a collection of n points $\mathbf{X}_1, \dots, \mathbf{X}_n$ of solution-cost and run-time at the end of n independent runs. The corresponding ECDF is then defined as:

$$F_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n I(\mathbf{X}_i \leq \mathbf{x}).$$

Note that the ECDF can take also points derived from *intersections* of point coordinates [17]. Algorithms for computing these ECDFs are described in [5,14].

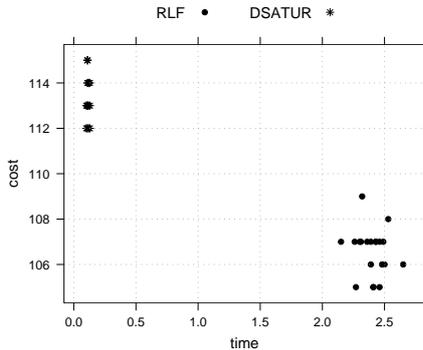


Fig. 3. Two sets of points in the space solution-cost and run-time obtained by running two construction heuristics for graph coloring on 20 instances.

Analysis In multivariate analysis the interest is either about the “external” structure of data (i.e., the configuration or interpoint distances of swarms of points in the Euclidean space \mathbb{R}^2 of solution-cost and run-time) or about the “internal” structure of the variables (i.e., how much correlated solution-cost and run-time are). In our specific case, the interest is mainly in the “external” structure which, in the two algorithms case, corresponds to comparing two samples of points by examining the center of gravity of the two swarms. Note that when the points represent results on different instances it may be necessary to apply some transformation to data in order to make their comparison meaningful.

This situation is represented in Figure 3 where we plot the points in the space solution-cost and run-time attained by two construction heuristics for the graph coloring problem, namely DSATUR [8] and RLF [22]. Each point represents one run of the construction heuristic on one instance and indicates the quality of the solution returned and the computational time. The data are collected by running both algorithms on a set of 20 instances of similar characteristics yielding 20 bivariate observations per algorithm.

In parametric statistics, a test to compare the bivariate means is the *Hotelling’s T^2 test* [1]. In the case of comparisons with more than two algorithms, the *multivariate analysis of variance* (MANOVA) [32] can be used to guarantee the overall confidence level before proceeding to the pairwise comparisons. In the non-parametric alternative, extensions of permutation tests based on the same Hotelling’s T^2 test statistic have been proposed in [16,26]. Note that the notion of rank ordering that underlies univariate non-parametric statistics does not readily extend into several dimensions.

As in the previous cases, one could also look more closely at the distribution of solution-cost and run-time, hence considering the corresponding ECDFs. Statistical tests based on a supremum test statistic similar to the Kolmogorov-Smirnov test for the two-sample case or the Birnbaum-Hall test for the multi-sample case can be applied to compare these distributions. However, the distribution of the

test statistics is not known in advance and, therefore, one has to implement permutation tests.

3.2 Scenario 2

In this scenario, the researcher is interested in the distribution of the solution cost over the time in which the metaheuristic was running. The analysis becomes more complex because, as described previously, it has to focus on *sets* of points of solution-cost and run-time collected during possibly multiple runs of the metaheuristics.

Characterization As suggested in [17,19,33], the distribution of solution-cost and run-time is seen as a *random set of bidimensional points* which is obtained during the run. Therefore, topics of *random set theory* seem appropriate for the analysis of metaheuristics under this scenario [17,15]. If only the improvements with respect to solution cost are recorded, the run of a metaheuristic can then be described as a set of *non-dominated* points of solution-cost and run-time [17]. Let $\mathcal{X} = \{\mathbf{X}_j \in \mathbb{R}^2, j = 1, \dots, m\}$ be a random set of m points of solution-cost, run-time where each element \mathbf{X}_j is non-dominated with respect to the other elements in \mathcal{X} . The new cumulative distribution function of solution-cost and run-time is then denoted by

$$F(\mathbf{x}) = \Pr[\mathcal{X} \preceq \mathbf{x}] \quad (4)$$

where $\mathcal{X} \preceq \mathbf{x}$ means that $\mathbf{X}_1 \leq \mathbf{x} \vee \dots \vee \mathbf{X}_m \leq \mathbf{x}$ [17]. The estimation of this probability is obtained from a collection of points of solution-costs and run-times whenever there is an improvement on the solution-cost during each of the n independent runs. The corresponding ECDF is then defined as follows:

$$F_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n I(\mathcal{X}_i \preceq \mathbf{x}) \quad (5)$$

where $\mathcal{X}_1, \dots, \mathcal{X}_n$ are n sets of non-dominated points of solution-cost and run-time obtained in n independent runs. Note that Eq. (4) corresponds to the *attainment function* (AF) and Eq. (5) to the *empirical AF* [17].¹⁰

The top plots of Figure 4 show several quantiles of the EAFs for the performance of a Novelty algorithm (left plot) and a Tabu Search algorithm (right plot) in 10 runs on instance `flat1000_60_0` of the graph coloring problem (see [9] for details). The bottom plot shows the median EAF, that is, the set of points whose probability of being attained in one single run is 50%. The plots clearly indicate that the specific Tabu Search algorithm performs better up to 500 seconds, whereas the Novelty algorithm performs better afterwards.

Stützle [33] proposed a similar perspective to analyze a metaheuristic by the ECDF of run-time for chosen bounds on the solution cost based on certain ratios from the known optimum (or lower bounds); the resulting distributions are called *qualified run-time distributions functions*.

¹⁰ Code for computing these ECDFs is available at www.tik.ee.ethz.ch/pisa/.

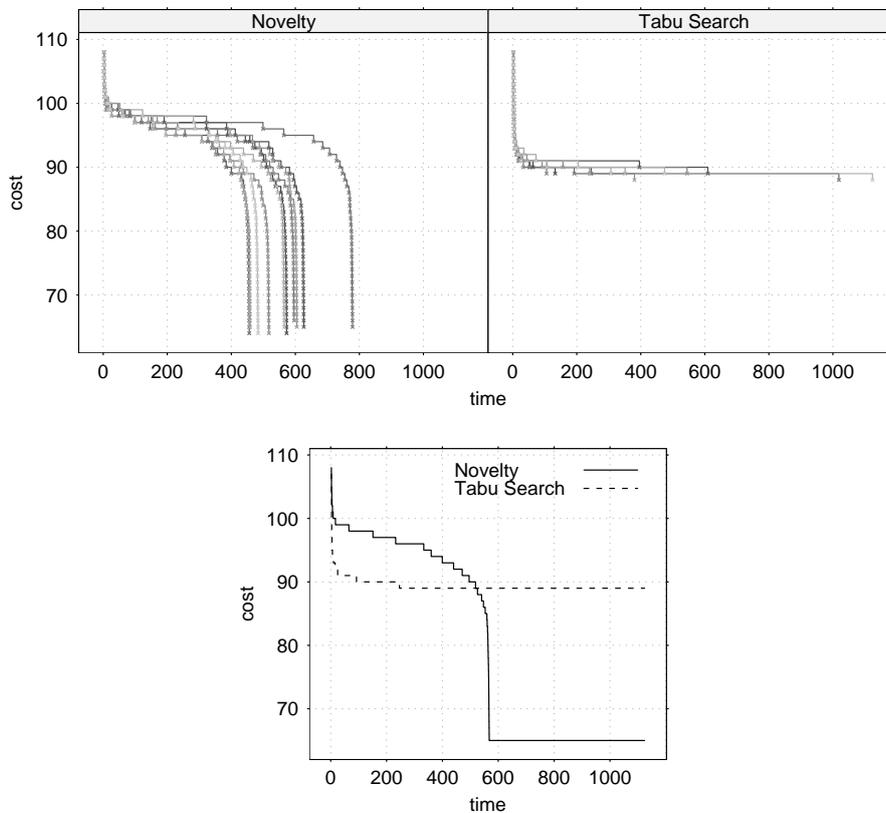


Fig. 4. Plots of EAFs for the performance of a Novelty and a Tabu Search algorithm for an instance of the graph coloring problem. See text for more details.

Analysis When comparing metaheuristics in this scenario, the best metaheuristic is the one that produces a set of points of solution-cost and run-time that dominates all the other sets of points of solution-cost and run-time associated with the other metaheuristics. It is difficult to find such metaheuristic in practice. There are those that converge to reasonably good solution quality very quickly, and those that can reach high quality solutions only after a large amount of time, following a slower convergence rate. The goal should be to find which metaheuristic performs better with respect to different intervals of computation time.

Very little research has been undertaken on this topic. We mention the work of Taillard [34] who suggested the use of statistical tests to compare solution costs of algorithms during the run. After collecting points of solution-cost and run-time associated to multiple runs of several metaheuristics, a Mann-Whitney test is

conducted for comparing the solution costs obtained by different metaheuristics each time an improvement was observed in any of the runs. The significance level of each test must be updated to take into account the multiple comparisons performed.

A different approach would be to use statistical tests analogous to Kolmogorov-Smirnov and Birnbaum-Hall tests with permutation arguments for testing the inequality of ECDFs. This has been done in a different context [31].

Finally, note that Eq. (4) cannot fully characterize metaheuristic performance since it does not take into account the dependence between points in a set [15]. Second-order moments that are described in [15] would be more informative. They permit investigating the probability that two points of solution-cost and run-time are attained simultaneously in a run.

4 Concluding Remarks and Open Issues

Examining the literature, we have the impression that the assessment of metaheuristics must be improved in order to produce results that are more scientific. In particular, a methodological approach must be followed. In this paper, we presented a framework for doing this, extending previous work to the multivariate case. This has received little attention to date. Some issues still remain.

- In the univariate models, the parametric assumptions of both normality and equal variance seem to be violated. Simulation studies that show the robustness of parametric models in the metaheuristics field would help to gain confidence on the reliability of results from these models.
- Some approaches exist in statistics for the analysis of Scenario 2 in the multivariate model. In particular, a study of the suitability of *repeated measurements methods* [1,26] should be undertaken.
- This latter scenario can be interpreted as a multiobjective problem, as noted in [17]. Therefore, unary performance measures that are used for the multiobjective case, such as the hypervolume and the ϵ -indicator, can be used as well. Although they have been shown to have some drawbacks [35], their use in the context of the multivariate analysis here suggested is worth investigating.
- Probably the most crucial issue in algorithm analysis is the possibility to generalize results to at least a class of instances. How to do this above all in the multivariate case is not well understood. Indeed, in both the multivariate scenarios considered, difficulties arise when aggregating data from different instances.
- Contrary to what has been done in the univariate model, advanced methods mentioned in Section 2.1 for the multivariate case have not yet been applied. A first attempt to extend racing algorithms to scenario 1 is given in [13]. The application of advanced designs, sequential testing procedures and regression trees in this context requires further development.

Acknowledgment The authors are grateful to the participants of the Workshop on Empirical Methods for the Analysis of Algorithms (EMAA) for their discussion on the main topic of this paper.

References

1. T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, 2003.
2. R. S. Barr, B. L. Golden, J. P. Kelly, M. G. Resende, and W. R. Stewart. Designing and Reporting on Computational Experiments with Heuristic Methods. *Journal of Heuristics*, 1:9–32, 1995.
3. T. Bartz-Beielstein. *Experimental Research in Evolutionary Computation – The New Experimentalism*. Springer Verlag, 2006.
4. T. Bartz-Beielstein and M. Preuss. Considerations of budget allocation for sequential parameter optimization (spo). In L. Paquete et al., editor, *Workshop on Empirical Methods for the Analysis of Algorithms, Proceedings*, pages 35–40, Reykjavik, Iceland, 2006.
5. J. L. Bentley. Multidimensional divide-and-conquer. *Communications of the ACM*, 23(4):214–229, 1980.
6. M. Birattari. *The Problem of Tuning Metaheuristics, as seen from a Machine Learning Perspective*. PhD thesis, Université Libre de Bruxelles, Belgium, 2004.
7. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Monterey CA, 1984.
8. D. Bréaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.
9. M. Chiarandini. *Stochastic local search methods for highly constrained combinatorial optimisation problems*. PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany, 2005. Forthcoming.
10. M. Coffin and M. J. Saltzman. Statistical analysis of computational tests of algorithms and heuristics. *INFORMS Journal on Computing*, 12(1):24–44, 2000.
11. P. R. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, 1995.
12. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, 1980.
13. M. L. den Besten. *Simple Metaheuristics for Scheduling: An empirical investigation into the application of iterated local search to deterministic scheduling problems with tardiness penalties*. PhD thesis, Darmstadt University of Technology, Darmstadt, Germany, October 2004.
14. C. M. Fonseca. Output-sensitive computation of the multivariate ECDF and related problems. In *Proceedings of COMPSTAT 2002*, page 30, 2002.
15. C. M. Fonseca, V. Grunert da Fonseca, and L. Paquete. Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In C. C. Coello et al., editors, *Evolutionary Multi-criterion Optimization (EMO 2005)*, LNCS 3410, pages 250–264. Springer Verlag, 2005.
16. P. I. Good. *Permutation Tests: A practical guide to resampling methods for testing hypothesis*. Springer Verlag, 2000.
17. V. Grunert da Fonseca, C. M. Fonseca, and A. Hall. Inferential performance assessment of stochastic optimizers and the attainment function. In *Evolutionary Multi-criterion Optimization (EMO 2001)*, pages 213–225, 2001.
18. D. P. Harrington and T. R. Fleming. A class of rank test procedures for censored survival data. *Biometrika*, 69:553–566, 1982.

19. H. Hoos. *Stochastic Local Search – Methods, Models, Applications*. PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany, 1998.
20. J. Hsu. *Multiple Comparisons - Theory and Methods*. Chapman & Hall/CRC, 1996.
21. D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
22. F. T. Leighton. A graph coloring algorithm for large scheduling problems. *Journal of Research of the National Bureau of Standards*, 84(6):489–506, 1979.
23. C. C. McGeoch. Towards an experimental method for algorithm simulation. *INFORMS Journal on Computing*, 8(1):1–15, 1996.
24. D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2005.
25. I. M. Ovacik, S. Rajagopalan, and R. Uzsoy. Integrating interval estimates of global optima and local search methods for combinatorial optimization problems. *Journal of Heuristics*, 6(4):481–500, 2000.
26. F. Pesarin. *Multivariate Permutation Tests – With Applications in Biostatistics*. John Wiley & Sons, 2001.
27. R. L. Rardin and R. Uzsoy. Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics*, 7(3):261–304, 2001.
28. E. Ridge and D. Kudenko. Analyzing Heuristic Performance with Response Surface Models: Prediction, Optimization and Robustness. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2007.
29. E. Ridge and D. Kudenko. Screening the Parameters Affecting Heuristic Performance. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2007.
30. S. N. Lophaven and H. B. Nielsen and J. Søndergaard. DACE - A MATLAB kriging toolbox. Technical Report IMM-TR-2002-12, IMM – Informatics and Mathematical Modelling, Technical University of Denmark, Kgs. Lyngby, Denmark, 2002.
31. K. J. Shaw, C. M. Fonseca, A. L. Nortcliffe, M. Thompson, J. Love, and P. J. Fleming. Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, volume 1, pages 34–75, 1999.
32. D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall, 2000.
33. T. Stützle. *Local Search Algorithms for Combinatorial Problems - Analysis, Algorithms, and New Applications*. PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany, 1998.
34. É. D. Taillard. Comparing non-deterministic iterative methods. In *Proceedings of the Third Metaheuristics International Conference*, pages 273–276, Porto, 2001.
35. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.