The final publication is available at Springer via
http://dx.doi.org/10.1007/978-3-319-10762-2_37

Distance Measures for Permutations in Combinatorial Efficient Global Optimization

Martin Zaefferer, Jörg Stork, and Thomas Bartz-Beielstein

Cologne University of Applied Sciences
Faculty for Computer and Engineering Sciences, 51643 Gummersbach, Germany
firstname.lastname@fh-koeln.de

Abstract. For expensive black-box optimization problems, surrogate-model based approaches like Efficient Global Optimization are frequently used in continuous optimization. Their main advantage is the reduction of function evaluations by exploiting cheaper, data-driven models of the actual target function. The utilization of such methods in combinatorial or mixed search spaces is less common. Efficient Global Optimization and related methods were recently extended to such spaces, by replacing continuous distance (or similarity) measures with measures suited for the respective problem representations.
This article investigates a large set of distance measures for their applicability to various permutation problems. The main purpose is to identify, how a distance measure can be chosen, either a-priori or online. In detail, we show that the choice of distance measure can be integrated into the Maximum Likelihood Estimation process of the underlying Kriging model. This approach has robust, good performance, thus providing a very nice tool towards selection of a distance measure.

1 Introduction

One frequent issue in real-world optimization problems are costly objective function evaluations. These may be caused by time-consuming simulations or complex trials and measurements. In continuous optimization, surrogate-model based approaches use cheaper, data-driven models to reduce the number of objective function evaluations, e.g. in the Efficient Global Optimization (EGO) algorithm [16]. In combinatorial optimization, surrogate models received less attention. Recently, approaches from continuous modeling and optimization have been extended to mixed or purely combinatorial problem spaces: Radial Basis Function Networks (RBFN), Kriging, and EGO [22, 34]. A short overview of these previous studies will be given in Sec. 2.

The employed modeling tools base their prediction on measures of similarity or distance between candidate solutions. The core idea of the extension is therefore to replace the distance measures used in continuous spaces (e.g., Euclidean) with distance measures more suited for the given problem representation. Two questions arise in this context: First, which distance measure is most suited? And second, how can this measure be chosen a priori as well as during the optimization procedure for a given problem?

This article tries to provide answers to both questions for an important solution representation type: permutations. The permutation representation is required in a large array of problems [2]. The reader may consider production

processes, which have to be divided into several jobs to be scheduled for one or more machines in order to achieve a timely completion. Here, several distance measures will be used to handle various problem classes and instances. The employed distance measures, Kriging and EGO will be introduced in Sec. 3. Their performance will be examined in an experimental study, as outlined in Sec. 4. Observations will be described and discussed in Sec. 5. Finally, a summary and an outlook on future research are given in Sec. 6.

2 Previous Research

Compared to their frequent use for continuous problem domains, surrogate model driven approaches are relatively unknown in combinatorial or mixed optimization [15]. Voutchkov et al. [31] introduce an expensive optimization problem for signed permutations, concerning weld sequence optimization. Regarding data-driven approaches for black-box problems (which are in the focus of this paper), Li et al. [21] proposed Radial Basis Function Network (RBFN) models based on a weighted distance measure, replacing the usual distance measure employed in RBFN. Their RBFN models were able to model to mixed-integer problems. Mixed problems also occur in algorithm tuning, where continuous, discrete, and categorical parameters may occur. In this context, Random Forest models have been used due to their ability to capture discrete and categorical parameters [4]. Hutter [14] also describes a Kriging model with a Hamming distance based kernel function to handle categorical variables.

Moraglio and Kattan [22] adapted an RBFN to arbitrary distance measures to model arbitrary combinatorial optimization problems. Their approach has also been applied to Quadratic Assignment Problems (QAP) [23]. The same conceptual extension for Kriging was recently investigated by Zaefferer et al. [34]. This allowed to apply the Kriging based Efficient Global Optimization (EGO) algorithm to combinatorial problems. Kriging-based EGO performed very well, in comparison to other model-driven or model-free approaches. Furthermore, the choice of distance measure was shown to have a very strong influence on optimization performance.

In this article, we will focus on Kriging-based EGO only. We will look at a much larger array of permutation problems and distance measures. Our goal is to derive recommendations for the selection of problem-specific distance measures.

3 Methods

3.1 Distance Measures

Previously, distance measures for permutations were investigated, e.g., for the purpose of landscape analysis [27] or diversity preservation [29]. These previous studies illustrate that a large array of distance measures is available. For the purpose of distance-based modeling, only Hamming, Swap and Interchange distance [23, 34] were used. In this study, we will analyze 14 different distance measures, as summarized in Table 1. The given runtime complexity refers to the employed implementations. More efficient variants may be available. To avoid scaling bias, all distance measures are scaled to yield values from $[0; 1]$.

In the following, we describe basic features of these distance measures. Since naming of measures in literature varies, this clarification is useful to avoid confusion.

Table 1. Investigated distance measures. Second column lists runtime complexity. Third column lists median runtime of 1000 evaluations for permutations of length 30.

Name	complexity	runtime [μs]	Abbrev.
Levenshtein	$O(n^2)$	7	Lev
Swap	$O(n^2)$	6	Swa.
Interchange	$O(n^2)$	14	Int.
Longest Common Subsequence	$O(n^2)$	8	LCSeq
Longest Common Substring	$O(n^2)$	8	LCStr
R	$O(n^2)$	5	R
Adjacency	$O(n^2)$	6	Adj.
Position	$O(n^2)$	6	Pos.
Position^2	$O(n^2)$	6	Posq.
Hamming	$O(n)$	2	Ham.
Euclidean	$O(n)$	6	Euc.
Manhattan	$O(n)$	4	Man.
Chebyshev	$O(n)$	3	Che.
Lee	$O(n)$	6	Lee

- Levenshtein and Edit distance are sometimes used as synonyms. In fact, Levenshtein is only one example of an edit distance. It counts the minimum number of deletions, insertions, or substitutions required to transform one string (or here: permutation) into another. For an implementation we refer to [32].
- A swap operation is the transposition of two adjacent elements in a permutation. The Swap distance is defined as the minimum number of swaps required to transform one permutation into another. It has also been called Precedence distance [27], or Kendall's Tau [19, 29]. For permutations of length n it is [29]:
$$\delta_{Swa.}(\pi, \pi') = \sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij} \quad \text{where} \quad z_{ij} = \begin{cases} 1 & \text{if } \pi_i < \pi_j \text{ and } \pi'_i > \pi'_j, \\ 0 & \text{otherwise.} \end{cases}$$
- An interchange operation is the transposition of two arbitrary elements. Respectively, the Interchange (also: Cayley) distance is the minimum number of interchanges required to transform one permutation to another [27].
- The longest common subsequence distance counts the largest number of elements that follow each other in both permutations, with interruptions. We use the algorithm described in [13].
- The longest common-substring distance counts the largest number of elements that follow each other in both permutations, without interruption, i.e., all elements are adjacent. We use the implementation in [33].
- The R-distance [9, 29] counts the number of times that one element follows another in one permutation, but not in the other. It is identical with the unidirectional adjacency distance [25]. It is computed by
$$\delta_R(\pi, \pi') = \sum_{i=1}^{n-1} y_i \quad \text{where} \quad y_i = \begin{cases} 1 & \text{if } \exists j : \pi_i = \pi'_j \text{ and } \pi_{i+1} = \pi'_{j+1}, \\ 0 & \text{otherwise.} \end{cases}$$
- The (bi-directional) adjacency distance [25, 27] counts the number of times two elements are neighbors in one, but not in the other permutation. Unlike R-distance (uni-directional), the order of the two elements does not matter.
- The Position distance [27] is identical with the Deviation distance or Spearman's footrule [29],
$$\delta_{\text{Pos}}(\pi, \pi') = \sum_{k=1}^{n} |i - j| \quad \text{where} \quad \pi_i = \pi'_j = k \ .$$

- The Squared Position distance is Spearman's rank correlation coefficient [29]. In contrast to the Position distance, the term $|i-j|$ is replaced by $(i-j)^2$
- The Hamming distance or Exact Match distance simply counts the number of unequal elements in two permutations, i.e.,

$$\delta_{Ham.}(\pi,\pi') = \sum_{i=1}^{n} a_i \quad \text{where} \quad a_i = \begin{cases} 0 & \text{if } \pi_i = \pi'_i, \\ 1 & \text{otherwise.} \end{cases}$$

- The Euclidean distance is

$$\delta_{Euc.}(\pi,\pi') = \sqrt{\sum_{i=1}^{n} (\pi_i - \pi'_i)^2} \ .$$

- The Manhattan distance (A-Distance [29, 9]) is

$$\delta_{Man.}(\pi,\pi') = \sum_{i=1}^{n} |\pi_i - \pi'_i| \ .$$

- The Chebyshev distance is

$$\delta_{Che.}(\pi,\pi') = \max_{1 \le i \le n} (|\pi_i - \pi'_i|) \ .$$

- The Lee distance [20] can be adapted to permutations with

$$\delta_{Lee}(\pi,\pi') = \sum_{i=1}^{n} min(|\pi_i - \pi'_i|, n - |\pi_i - \pi'_i|) \ .$$

The reversal distance (number of reversals required to transform one permutation to another) was not used, even though it is especially promising for the Traveling Salesperson Problem (TSP). Calculating the reversal distance for unsigned permutations is NP-hard [10].

3.2 Kriging for Combinatorial Optimization

Kriging is a very flexible predictor, that models the correlation between samples, assuming that they are derived from a Gaussian process. Kriging also provides an estimate of uncertainty of its own prediction. For a detailed description of Kriging, we refer to Forrester et al. [11]. The adaptation to combinatorial or mixed problems was described by Zaefferer et al. [34].

Training a Kriging model requires the set of m solutions $\boldsymbol{X} = \{\mathbf{x}^{(i)}\}_{i=1\ldots m}$ with observations $\mathbf{y} = \{y^{(i)}\}_{i=1\ldots m}$. The predicted mean of a new candidate solution $\mathbf{x}$ is referred to as $\hat{y}(\mathbf{x})$, the estimated error of that prediction is $\hat{s}(\mathbf{x})$. The Kriging model has the parameters θ, p, $\hat{\sigma}$,and $\hat{\mu}$. Maximum Likelihood Estimation (MLE) is used to determine these parameters and requires a matrix inversion. In [34], standard inversion was used. We observed a problem with standard inversion for 4 out of the 14 distance measures (Int., Lev., LCSeq, Che.). While all others worked well, these four measures may produce numerical instability. Hence, the more stable inversion via Cholesky decomposition is used, which requires a positive semi-definite correlation matrix.

3.3 Choosing a Distance Measure in Kriging

In standard Kriging, the distance measure is not fixed. Rather, it can partially be understood as a parametrized distance measure. E.g., it may resemble Euclidean ($p = 2$) or Manhattan ($p = 1$) distance.

The choice of distance measure can also be understood as a (categorical) parameter of the model. Hence, we suggest to perform MLE for each distance measure separately. Afterwards, the distance measure with maximum likelihood is chosen for the model. This procedure repeats every time the model is build, i.e., in each iteration of a single EGO run. In the experimental study this will be referred to as *"All"*.

A wrong decision may occur, especially while data is still very sparse. Therefore, we expect the performance of choosing a distance measure with MLE to be equal to or worse than the best single measure. An exception would be the case where the underlying optimization problem has a dynamic behavior. Then, different measures may be preferable in different phases of the optimization run.

3.4 Efficient Global Optimization

EGO was introduced by Jones et al. [16]. In this algorithm, a Kriging model is first build based on an initial set of solutions. If the uncertainty $\hat{s}(\mathbf{x}) > 0$, one can compute the Expected Improvement (EI) of a candidate solution, otherwise $\mathrm{EI}(\mathbf{x}) = 0$. EI determines how much improvement can be expected from the candidate solution, and thus balances exploitation vs. exploration. The solution that maximizes EI is evaluated with the target function. The result is used to update the Kriging model. This is repeated until a termination criterion (e.g., function evaluation budget) is fulfilled.

4 Experimental Setup

4.1 Correlation Between Distances

As a first step, correlation between the 14 different distance measures is investigated. Distances between all permutations of length $n = 7$ are computed (i.e., 5040 distance values for each measure), and the correlation is calculated.

4.2 Matrix Condition

To quickly assess whether all measures yield positive semi-definite correlation matrices (as required for MLE), we performed an experimental test. Ten solutions were created randomly, while another 90 were created by consecutive interchange mutations. This yields 100 solutions of varying distances. This was done for various permutation lengths ($n = \{5, 6, 7, 8, 9, 10, 20, 50, 100\}$). In case of the smallest instance, the 100 solutions represent a very large section of the search space (which has a size of $n! = 120$). Larger instances are less crowded. Since θ will also influence the correlation matrix condition, it was varied from 10^{-10} to 10^{10}. For each distance measure, each dimension n, and each θ the correlation matrix is computed and its condition checked.

4.3 Benchmark Problems

For all further experiments, five permutation problem classes are used.

- As in [34], four instances of the Quadratic Assignment Problem (QAP) [6] from the QAPLIB [7] are chosen (`nug30`, `nug12`, `tho30` and `kra32`). In the QAP n facilities have to be assigned to n locations. Assignment cost is minimized, based on flow between facilities and distance between locations.
- Four instances of the Flow-shop Scheduling Problem (FSP) [30] are chosen (reC05, reC13, reC19, reC31 [24]) from the OR-Library [5]. Here, the finishing time of the last of n jobs sequenced on m machines is minimized.
- Three TSP instances are chosen from the TSPLIB [26] (bayg29, fri26, gr24). In the TSP, the cost or length of a route through several locations is minimized. Each location has to be visited once.

- Three instances of the Asymmetric TSP (ATSP) are generated (atsp10, atsp20, atsp30). For each instance, a distance matrix is created randomly with a uniform distribution. The three instances are of size 10, 20, and 30. In contrast to TSP, the cost of traveling between two locations depends on direction.
- Finally, four instances of the single-machine total Weighted Tardiness problem (WT) [1] are chosen, also from the OR-Library [5] (the first four of length 40, i.e., wt40a, wt40b, wt40c, wt40d). Here, n jobs are sequenced on one machine that can handle one job at a time. The tardiness of a schedule for all jobs, weighted by a set of n given weights is minimized. It depends on the given processing times and due dates of each job.

For QAP, TSP, ATSP and WT the length of the permutation n is given by the number in the instance name. For FSP, n is 20, 20, 30 and 50 for reC05, reC13, reC19 and reC31 respectively.

We use this benchmark set under the artificial assumption of costly target function evaluation. While some of these problems have actual real world relevance (e.g., based on real world data), none may be considered expensive. This allows for a more in-depth study, providing first results, which of course should be validated with actually expensive problems in future studies.

4.4 Local Fitness Distance Correlation

Fitness Distance Correlation [17] is a measure for the analysis of fitness landscapes. It measures correlation between fitness value and distance to the known global optimum. When the optimum is unknown, it can be replaced by the best solution in the set, thus yielding the Local FDC (LFDC) [18]. Here, LFDC will be calculated based on 20,000 unique, randomly created individuals for each instance. When minimizing an easy problem, positive correlation is expected to occur. Thus, the LFDC values may represent an indicator of problem difficulty. We are interested in LFDC from a different perspective. It is investigated, whether LFDC can be used to identify a suitable distance measure for a given problem. LFDC may be unsuited towards this end, as previous studies already showed that FDC may be misleading for certain problems [3].

4.5 Optimization Performance

Finally, we compare optimization performance. To that end, EGO, a model-free Genetic Algorithm (GA), Random Search (RS) and a simple 2-opt local search are employed with a strictly limited budget of 200 function evaluations. GA, RS and 2-opt are baselines in this comparison. Their main purpose is to identify whether the various EGO variants work at all.

The GA used in the comparison will use cycle crossover and the mutation operator is an interchange of arbitrary elements. Furthermore the algorithm will use a population size of ten, crossover rate 0.5, mutation rate $1/n$, tournament selection with tournament size two and tournament probability 0.9. The EGO algorithm will start with an initial set of ten solutions. Kriging parameter p is set to one, while the others are determined with MLE. Internally, EGO will perform optimization of the (assumed to be cheap) surrogate model. Hence, the same GA is used with 10,000 model evaluations, and a population size of 20.

For a fair comparison of actual competitors, most of the mentioned parameters would require tuning. Since the basic GA is just a baseline, this is not necessary. The EGO variants use identical settings thus yielding a fair comparison among themselves.

5 Observations and Discussion

5.1 Correlation and LFDC

Correlation between distance measures and LFDC values are depicted in Fig. 1. Measures that correlate also have comparable LFDC (e.g., Pos. and Posq.). The LFDC for QAP, TSP & ATSP as well as FSP & WT have similar structure.

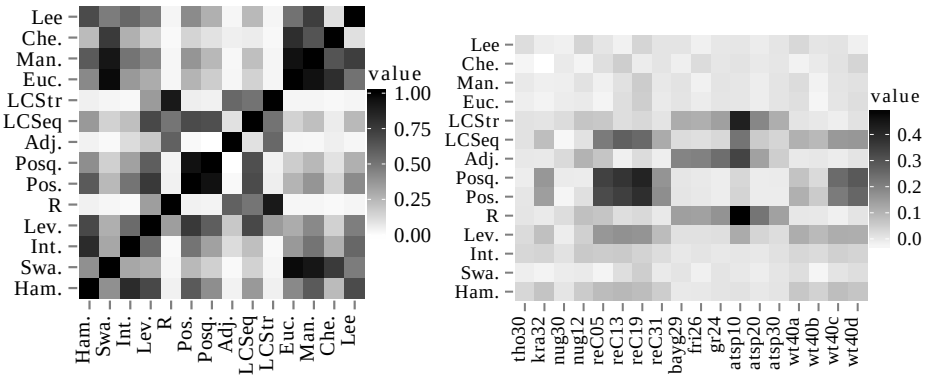

Fig. 1. Heatmaps of distance measure correlation (left), and LFDC values (right).

5.2 Matrix Condition

For most distance measures, a positive semi-definite matrix could be determined for each n. Only Adjacency distance with $n = 5$ and $n = 6$ did not yield any positive semi-definite matrices. Hence, one should avoid using Adjacency distance when the training samples represent a large portion of the search space.

5.3 Optimization Performance

Figure 2 shows the results of the optimization experiments. Each EGO variant is referred to by the name of the employed distance measure. Results of 2-opt are not shown, for the sake of brevity. 2-opt usually ranks worse than GA and only outperforms GA for the three TSP instances. Still, it can not compete with the model-based approaches. Chebyshev distance (Che.) and RS are consistently outperformed by the GA and hence not included in the plot.

Three main groups with similar structure can be identified: first, the QAP instances, second, the TSP and ATSP instances, and third, the WT and FSP instances. Members of each group have a similar pattern, although the best performing method may not be identical for all members. These three blocks do coincide with the structure that is visible in the LFDC results. LFDC may identify the best measure (see e.g., R-Distance and atsp10) or may fail completely (e.g., Posq. and reC19). LFDC is apparently unsuited to identify a proper distance measure.

Overall, the model-free GA is always outperformed by at least 5 EGO variants. Choosing the wrong distance measure may however lead to performance worse than the model-free GA. Choosing a distance measure with MLE (*All*) never ranks worse than 3rd best, making it the most robust method in this test bed. *All* ranks first place in 7 of 18 instances. It seems that the underlying problem does profit from a dynamic choice of distance measure. This behavior may be related to dynamic behavior observed for the choice of mutation operators through self-adaption [28]. Many distance measures can be related to

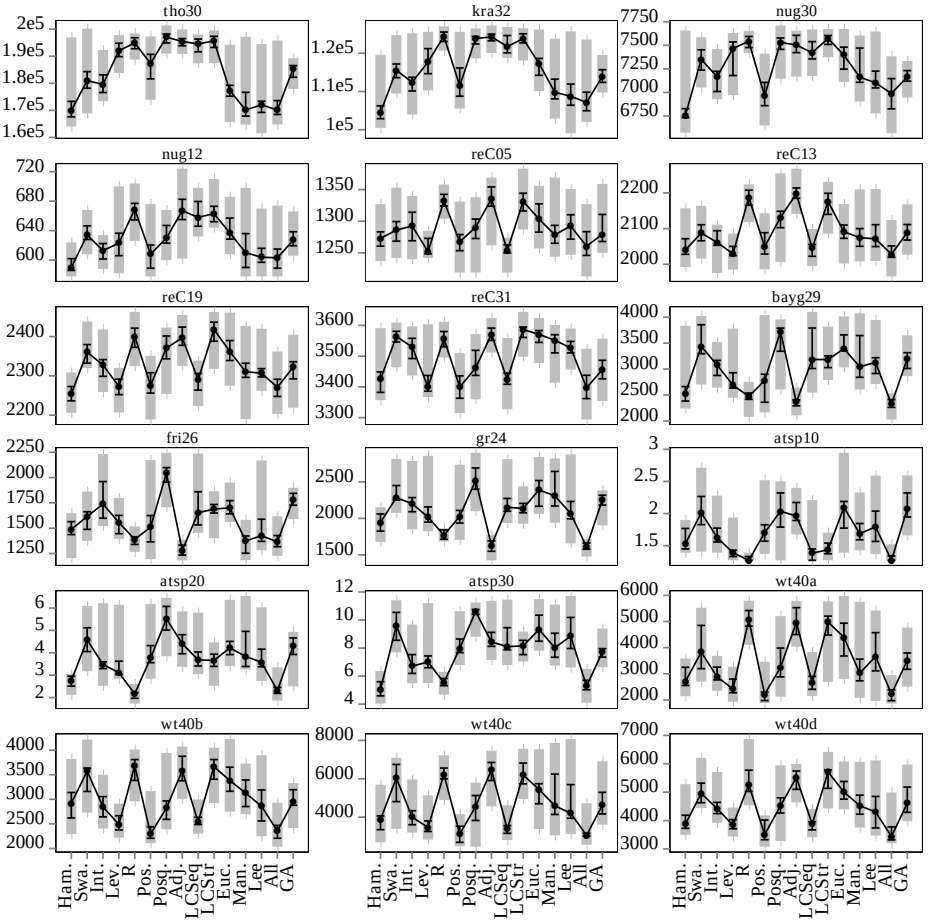

Fig. 2. Optimization performance: Dots are median, black bars are interquartile range, thick grey bars are range from minimum to maximum. Smaller values are better.

specific mutation operators. The single best distance measure is Hamming distance, yielding best results in 6 of the 18 test problems, but receiving lower ranks for other instances. For each problem class, the single best distance measures are: Ham. for QAP, Lev. for FSP, Adj. for TSP, R for ATSP, Pos. for WT. While the scheduling problems rather reflect the importance of a relative order, TSP or ATSP are more concerned with adjacency of neighboring cities. Hence, it makes sense that a bi-directional adjacency measure is used for TSP, while uni-directional adjacency (R-Distance) is used for the ATSP instances. In ATSP, direction matters, whereas in TSP it does not.

The nice and robust performance of choosing a distance measure with MLE makes for a promising result. Here, the only issue is to carefully avoid numerical problems, i.e., to use matrix inversion via Cholesky decomposition. Should the increased computational effort necessitate a smaller set of distance measures, Hamming distance should always be included, due to good performance and lowest cost.

6 Summary and Outlook

This work investigated the suitability of various distance measures in surrogate modeling for the optimization of several permutation problems. It was shown, that each problem class or instance may require a different distance measure. Correlation between distance and fitness values (LFDC) proved to be a poor way of selecting a distance measure for a given problem class or instance. On the other hand, integrating the selection of a measure into the MLE process of a Kriging model proved to be a very well performing and robust approach.

Further research may focus on learning distance measures for Kriging-based models in combinatorial spaces. Learning of correlation functions with Genetic Programming is not new [12]. Also, distance measures have been evolved with GA [8] in the context of string matching. Combining both ideas to evolve better measures for distance-based models may thus be an interesting path to follow. If interpretable distance measures evolve, this may also give interesting insight into the underlying problems.

Acknowledgments This work has been kindly supported by the Federal Ministry of Education and Research (BMBF) under the grant CIMO (FKZ 17002X11).

References

1. T. Abdul-Razaq, C. Potts, and L. V. Wassenhove. A survey of algorithms for the single machine total weighted tardiness scheduling problem. *Discrete Applied Mathematics*, 26(2–3):235 – 253, 1990.
2. A. Allahverdi, C. Ng, T. E. Cheng, and M. Y. Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032, 2008.
3. L. Altenberg. Fitness distance correlation analysis: An instructive counterexample. In T. Bäck, editor, *ICGA*, pages 57–64. Morgan Kaufmann, 1997.
4. T. Bartz-Beielstein, M. de Vegt, K. E. Parsopoulos, and M. N. Vrahatis. Designing particle swarm optimization with regression trees. Technical Report CI–173/04, Universität Dortmund, Mai 2004.
5. J. E. Beasley. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.
6. R. E. Burkard. Quadratic assignment problems. *European Journal of Operational Research*, 15(3):283 – 289, 1984.
7. R. E. Burkard, S. E. Karisch, and F. Rendl. QAPLIB – a quadratic assignment problem library. *Journal of Global Optimization*, 10(4):391–403, 1997.
8. D. Camacho, R. Huerta, and C. Elkan. An evolutionary hybrid distance for duplicate string matching. Technical report, Universidad Autonoma de Madrid, 2008.
9. V. Campos, M. Laguna, and R. Martí. Context-independent scatter and tabu search for permutation problems. *INFORMS Journal on Computing*, 17(1):111–122, 2005.
10. A. Caprara. Sorting by reversals is difficult. In *Proceedings of the First Annual International Conference on Computational Molecular Biology*, RECOMB '97, pages 75–83, New York, NY, USA, 1997. ACM.
11. A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling*. Wiley, 2008.
12. C. Gagné, M. Schoenauer, M. Sebag, and M. Tomassini. Genetic programming for kernel-based learning with co-evolving subsets selection. In *Parallel Problem Solving from Nature*, volume 4193 of *Lecture Notes in Computer Science*, pages 1008–1017. Springer Berlin Heidelberg, 2006.

13. D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343, 1975.
14. F. Hutter. *Automated configuration of algorithms for solving hard computational problems*. PhD thesis, University of British Columbia, 2009.
15. Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011.
16. D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
17. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann Publishers Inc., 1995.
18. L. Kallel and M. Schoenauer. Fitness distance correlation for variable length representations. Technical report, Ecole Polytechnique, 1996.
19. M. Kendall and J. Gibbons. *Rank correlation methods*. A Charles Griffin Book. E. Arnold, 1990.
20. C. Lee. Some properties of nonbinary error-correcting codes. *Information Theory, IRE Transactions on*, 4(2):77–82, June 1958.
21. R. Li, M. T. M. Emmerich, J. Eggermont, E. G. P. Bovenkamp, T. Back, J. Dijkstra, and J. Reiber. Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. In *Congress on Evolutionary Computation*, pages 2764–2771. IEEE, 2008.
22. A. Moraglio and A. Kattan. Geometric generalisation of surrogate model based optimisation to combinatorial spaces. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 142–154. Springer-Verlag, 2011.
23. A. Moraglio, Y.-H. Kim, and Y. Yoon. Geometric surrogate-based optimisation for permutation-based problems. In N. Krasnogor et al., editors, *Genetic and Evolutionary Computation Conference*, pages 133–134. ACM, 2011.
24. C. R. Reeves. A genetic algorithm for flowshop sequencing. *Computers & operations research*, 22(1):5–13, 1995.
25. C. R. Reeves. Landscapes, operators and heuristic search. *Annals of Operations Research*, 86:473–490, 1999.
26. G. Reinelt. TSPLIB—A traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.
27. T. Schiavinotto and T. Stützle. A review of metrics on permutations for search landscape analysis. *Computers & operations research*, 34(10):3143–3153, 2007.
28. M. Serpell and J. E. Smith. Self-adaptation of mutation operator and probability for permutation representations in genetic algorithms. *Evolutionary Computation*, 18(3):491–514, 2010.
29. M. Sevaux and K. Sörensen. Permutation distance measures for memetic algorithms with population management. In *Metaheuristics International Conference*, pages 832–838, 2005.
30. E. Taillard. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47(1):65–74, 1990.
31. I. Voutchkov, A. Keane, A. Bhaskar, and T. M. Olsen. Weld sequence optimization: The use of surrogate models for solving sequential combinatorial problems. *Computer Methods in Applied Mechanics and Engineering*, 194(30-33):3535–3551, Aug 2005.
32. R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173, 1974.
33. Wikipedia. Longest common substring problem — wikipedia, the free encyclopedia, 2014. [Online; accessed 26-March-2014].
34. M. Zaefferer, J. Stork, M. Friese, A. Fischbach, B. Naujoks, and T. Bartz-Beielstein. Efficient global optimization for combinatorial problems. In *Genetic and Evolutionary Computation Conference*, 2014. Preprint (Accepted).