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 spaces is less common. Efficient Global Optimization and related methods were extended recently to such spaces, by replacing continuous distance (or similarity) measures with measures suited for the respective problem representations.
This article investigates a larg 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.

Keywords: Efficient Global Optimization, Surrogate Model, Combinatorial Optimization, Permutation, Distance Measure, Genetic Algorithm

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 [14]. 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. This Radial Basis Function Networks (RBFN), Kriging and EGO [23, 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 presented in this paper is therefore to replace the distance measures used in continuous

spaces (e.g., euclidean or similar measures) with distance measures more suited for the given solution representations. Two questions arise in this context:

- Which distance measure is most suited?
- How can this measure be chosen a priori as well as during the optimization procedure, i.e., offline and online, 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. 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 in a Kriging model will be used to solve various problem types and instances. The employed distance measures, Kriging model and EGO will be introduced in Sec. 3. The performance of the various methods will be examined in an experimental study, as outlined in Sec. 4. Observations will be described in Sec. 5 and discussed in Sec. 6. Finally, the paper is summarized and an outlook on future research is given in Sec. 7.

2 Previous Research

Compared to their frequent usage for continuous problem domains, surrogate model driven approaches are relatively unknown in combinatorial or mixed optimization [13]. Regarding data-driven approaches for black-box problems (which are in the focus of this paper), Li et al. [22] proposed Radial Basis Function Network (RBFN) models based on a weighted distance measure, replacing the usual distance measures 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 [2]. Hutter [12] also describes a Kriging model with a Hamming distance based kernel function to handle categorical variables.

Moraglio and Kattan [23] adapted an RBFN to arbitrary distance measures to model arbitrary combinatorial optimization problems. They performed first tests with NK-Landscapes. Their approach has since been applied to Quadratic Assignment Problems (QAP) [25], package-deal negotiation [8], and tree-based problems from Genetic Programming (GP) [24]. GP has also been coupled with RBFNs to evolve better discrete surrogate models [17]. As Moraglio and Kattan [23] indicate, the generalization of RBFN for arbitrary distance is also possible with other models, e.g., with Kriging. This was recently investigated by Zaefferer et al. [34]. Besides a linear modeling approach, Kriging and RBFN were employed in a model-based optimization algorithm, as well as compared to model-free algorithms. Kriging and RBFN may provide an error estimate, thus enabling the calculation of the Expected Improvement of a candidate solution. Hence, Efficient Global Optimization (EGO) was implemented for combinatorial problems. Zaefferer et al. [34] employed Hamming, Swap and Interchange

distance. Kriging-based EGO was shown to outperform all other model-based approaches, as well as all model-free approaches. It was also shown that the choice of distance measure has a very strong influence on optimization performance.

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

3 Methods

3.1 Distance measures

In other context, some research investigated distance measures for permutations, e.g., for the purpose of landscape analysis [29] or diversity preservation [30]. These previous studies illustrate that a large array of distance measures is available.

For the purpose of distance-based modeling for permutation problems, only Hamming and Swap [25] as well as Interchange Distance [34] were used. Although Hamming distance worked very well, several questions remain unanswered. This includes performance of various other measures,or a practical decision method towards selecting a measure when building a model.

In this study, we will analyse 14 different distance measures, as summarized in Table 1. The given runtime complexity refers to the employed implementations. More efficient variants may be available. All distance measures were implemented in `R`[1], which uses efficient C code whenever it is possible. The large runtime of Interchange Distance is due to an unresolved dependency on slower `R` code.

To avoid scaling bias, all distance measures are scaled to yield values from $[0; 1]$. Furthermore, cyclic or reversal independency are not guaranteed: for some problems, a permutation may be left or right shifted (cyclic independency) or even reversed (reversal independency), but will still yield the same function value, e.g., in the Traveling Salesman Problem (TSP). This will not be reflected by the distance measures in this work, but may be of interest for future research. To give further information on the employed measures, we describe their basic features. Since naming of measures in literature varies, this clarification is useful to avoid confusion.

- Levenshtein and edit distance are sometimes used as synonyms. In fact, Levenshtein is only one example of an edit distance. It calculates the minimum number of deletions, insertions, or substitutions required to transform one string (or here: permutation) into another. For an implementation we refer to Wagner and Fischer [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

[1] `R` is a language for statistical computing, see `www.r-project.org`

Table 1. List of investigated distance measures. TODO complexity needs revision. Second column reports runtime complexity. Third column reports median runtime distance calculations for permutation of size $N = 30$, estimated by 1000 evaluations.

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

required to transform one permutation into another. It has also been called precedence distance [29], or Kendall's Tau [18, 30]. It can be computed as described in [30]:

$$\delta_{Swa.}(\pi, \pi') = \sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij} \quad \text{where} \quad z_{ij} = \begin{cases} 1 & \text{if } \pi_i < \pi_j \text{ and } \pi'_i > \pi'_j, \\ 0 & \text{otherwise} \end{cases}.$$

- An interchange operation is the transposition of two arbitrary elements in a permutation. Respectively, the Interchange (also: Cayley) distance is the minimum number of interchanges required to transform one permutation to another [29].
- The longest common subsequence distance counts the largest number of elements that follow each other in both permutations, with interruptions. We use the algorithm provided by Hirschberg [11].
- 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 from [33].
- The R-distance [7, 30] counts the number of times that one element follows another in one permutation, but not in the other. It is the same as the uni-directional adjacency distance measure [27]. 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 [27, 29] counts the number of times two elements are neighbors in one, but not in the other permutation. Unlike R-distance (uni-directional adjacency), the order of the two elements does not matter.
- The Position Distance [29] is the same as Deviation Distance or Spearman's footrule [30], $\delta_{\text{Pos}}(\pi, \pi') = \sum_{k=1}^{n} |i - j| \quad \text{where} \quad \pi_i = \pi'_j = k$.

- The Squared Position distance is Spearman's rank correlation coefficient [30]. 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.
  $\delta_{Ham.}(\pi,\pi') = \sum_{i=1}^{n} a_i$ where $a_i = \begin{cases} 0 & \text{if } \pi_i = \pi'_i, \\ 1 & \text{otherwise} \end{cases}$
  It can also be understood as an edit distance with substitutions only.
- The Euclidean distance is $\delta_{Euc.}(\pi,\pi') = \sqrt{\sum_{i=1}^{n}(\pi_i - \pi'_i)^2}$.
- The Manhattan distance is $\delta_{Man.}(\pi,\pi') = \sum_{i=1}^{n} |\pi_i - \pi'_i|$. It is the same as the A-distance [30, 7].
- The Chebyshev distance is $\delta_{Che.}(\pi,\pi') = \max_{1\le i\le n}(|\pi_i - \pi'_i|)$.
- The Lee Distance [21] can be adapted to permutations with $\delta_{Lee}(\pi,\pi') = \sum_{i=1}^{n} min(|\pi_i - \pi'_i|, n - |\pi_i - \pi'_i|)$.

Other distance measures are possible (and could be evolved with GA [6] or Genetic Programming), but are not considered in this study.

3.2 Kriging for combinatorial optimization

For a detailed description of Kriging, we refer to Forrester et al. [9]. The adaptation to combinatorial or mixed problems was described by Zaefferer et al. [34].

The following notation is adopted from Forrester et al. [9]. Given a set of n solutions $\boldsymbol{X} = \{\mathbf{x}^{(i)}\}_{i=1\ldots n}$ in a k-dimensional continuous search space with observations $\mathbf{y} = \{y^{(i)}\}_{i=1\ldots n}$, Kriging is a method to find an expression for a predicted value at an unknown point by interpreting the observed responses $\mathbf{y}$ as if they are realizations of a stochastic process. The following set of random vectors $\boldsymbol{Y} = \{Y(\mathbf{x}^{(i)})\}_{i=1\ldots n}$ is used to define this stochastic process. The random variables $Y(\cdot)$ are correlated as follows [9]:

$$\operatorname{cor}\left[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)})\right] = \exp\left(-\sum_{j=1}^{k} \theta_j |x_j^{(i)} - x_j^{(l)}|^{p_j}\right). \tag{1}$$

The weights θ_j and the shape parameter p_j have to be estimated. All pairs $\{(i,l)\}$ are collected in the correlation matrix $\boldsymbol{\Psi}$. It is used in the Kriging predictor

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \boldsymbol{\psi}^T \boldsymbol{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \tag{2}$$

where $\hat{y}(\mathbf{x})$ is the predicted function value of a new sample $\mathbf{x}$, $\hat{\mu}$ is the maximum likelihood estimate of the mean and $\boldsymbol{\psi}$ is the vector of correlations between training samples $\boldsymbol{X}$ and the new sample $\mathbf{x}$. The error of the prediction can be estimated with

$$\hat{s}^2(\mathbf{x}) = \hat{\sigma}^2(1 - \boldsymbol{\psi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi}^T), \tag{3}$$

where $\hat{\sigma}^2$ is another parameter of the model to be estimated. The (usually small) contribution of error due to estimation of $\hat{\mu}$ is omitted.

As described by Zaefferer et al. [34], Eq. (1) has to be transformed for combinatorial or mixed spaces:

$$\operatorname{cor}\left[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)})\right] = \exp(-\theta \mathrm{d}(\mathbf{x}^{(i)}, \mathbf{x}^{(l)})^p), \tag{4}$$

where d(·) can be any distance measure for the given problem representation.

Maximum Likelihood Estimation (MLE), which comprehends an optimization procedure, is used to determine the model parameters, i.e., θ, p, $\hat{\sigma}$,and $\hat{\mu}$. MLE requires a matrix inversion (also later in the prediction step, see (2)), which can usually be performed directly or via Cholesky decomposition. In the previous article [34], standard inversion was used. Here, we observed a problem with standard inversion for 4 of the 14 distance measures (Int., Lev., LCSeq, Che.). While all others worked well, these four measures may produce numerical instability, as exemplified in Fig. 1. In case of standard inversion, the depicted

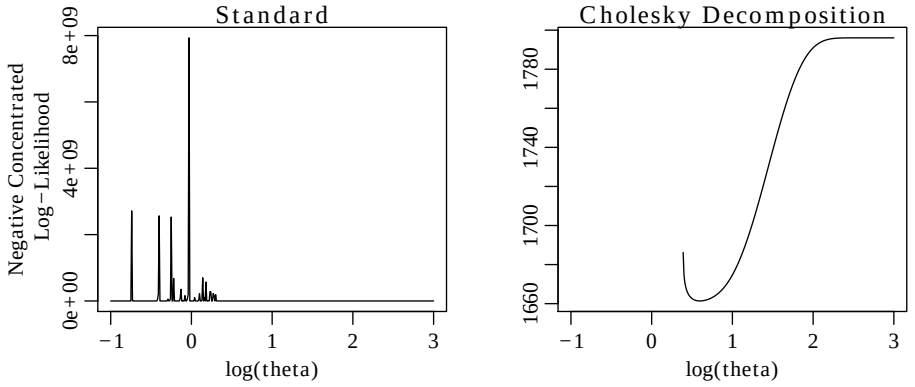

Fig. 1. Negative concentrated log-likelihood plotted versus log θ. Likelihood landscape for a Kriging model based on 100 solutions of the tho30 QAP instance, using Levenshtein distance. Standard matrix inversion is compared to inversion via Cholesky Decomposition. Missing values in the right plot represent non positive semi-definite correlation matrices.

spikes in the likelihood landscape lead to a wrong choice of θ. Hence, Cholesky decomposition is chosen instead. Preliminary experiments showed that this does not change behavior of the other 10 distance measures.

3.3 Choosing a distance measure in Kriging

In standard/continuous Kriging, the distance measure is not actually 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. These choices may or may not be included in the MLE process.

In combinatorial spaces, one may also have such parameters (e.g., the cost for each of the three edit operations in the Levenshtein distance may be seen as parameters). Apart from such special cases, the primary interest is not to determine parameters, but rather to select from a discrete set of measures. We suggest a very simple approach to this problem: perform MLE for each distance

measures separately. Afterwards, the distance measure that yielded best likelihood is chosen for the model. This approach will be referred to as "All" in the comparison of the various single distance measures.

It has to be expected, that this approach can never be as good as any approach based on the best single distance measure. This is due to the fact, that wrong choices will occur, especially while data is still very sparse. An exception would be the case where the underlying optimization problem has some dynamic behavior. Then, one distance measure may be adequate in the beginning of the optimization run, whereas another becomes better suited at the end. This is not expected for the simple benchmark problems examined in this study. Therefore, we expect the performance of choosing distance measure with MLE to be equal to or worse than the best single measure.

3.4 Efficient Global Optimization

The Kriging model introduced in Sec. 3.2 can be employed in an Efficient Global Optimization algorithm. EGO was introduced by Jones et al. [14] and recently extended to combinatorial problems by Zaefferer et al. [34]. In this algorithm, a Kriging model is first build based on an initial set of solutions. Based on this, the Expected Improvement (EI) of a candidate solution is computed with [9]

$$\mathrm{EI}(\mathbf{x}) = (y_{\min} - \hat{y}(\mathbf{x}))\Phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}\phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right),$$

otherwise $\mathrm{EI}(\mathbf{x}) = 0$. Here, $y_{\min}$ is the lowest target function value found so far. EI determines how much improvement can be expected from the candidate solution to be predicted. The solution that maximizes EI is evaluated with the target function. The result is used to update the Kriging model until a termination criterion is fulfilled (here: a fixed budget of target function evaluations).

4 Experimental Setup

4.1 Correlation between distances

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

4.2 Matrix condition

To quickly assess whether all measures yield positive semi-definite correlation matrices, we performed an experimental test. Ten samples were created randomly, while another 90 are created by consecutive interchange mutations. This yields 100 samples of varying distances. This was done for various dimensions ($n = \{5, 6, 7, 8, 9, 10, 20, 50, 100\}$). In case of the smallest instance, the 100 samples represent a very large section of the search space (which has a size of $N! = 120$), whereas the larger instances yield less crowding.

Since θ will 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, 5 different permutation problem types are investigated.

- As in [34], four instances of the Quadratic Assignment Problem (QAP) [4] from the QAPLIB [5] 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 instance of the Flow-shop Scheduling Problem (FSP) [31] are chosen (reC05, reC13, reC19, reC31 [26]) from the OR-Library [3]. Here, the finishing time of the last of n jobs sequenced on m machines is minimized.
- Three TSP instances are chosen from the TSPLIB [28] (bayg29, fri26, gr24). In the TSP, the cost or length of a route through several locations is minimized, where each location has to be visited once.
- Three instances of the Asymetric TSP (ATSP) are generated. For each instance, a distance matrix is created randomly with a uniform distribution. The three instances are of size 10, 20, and 30 (i.e., the number of locations to visit). The main difference to TSP is that the cost (or duration) of traveling between two locations is not independent of direction. Thus, the distance matrix can be asymmetric. The instances are referred to as atsp10, atsp20 and atsp30.
- Finally, four instances of the single-machine total Weighted Tardiness problem (WT) [1] are chosen, also from the OR-Library [3] (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 dimension (i.e. length of the permutation) is given by the number in the instance name. For FSP, the length (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 from industry.

4.4 Local Fitness distance correlation

Fitness Distance Correlation [15] is a popular measure for the analysis of fitness landscapes. Several samples are drawn from the search space. Their fitness values and their distances to the known optimum are determined and the corresponding

correlation (between fitness and distance) is computed. In the context of minimization, small distances should correlate with low fitness values in case of easy problems, i.e., positive correlation. Misleading problems on the other hand may yield negative correlation. Thus, the FDC values may represent an indicator of how difficult a problem is.

We are interested in FDC from a different perspective. Rather than calculate FDC to estimate the difficulty of a landscape based on a given distance measure, it would be interesting to see whether FDC is of any use to identify a suitable distance measure for a given fitness landscape.

To avoid the issue of unknown global optima, one can resort to several ways of replacing the optimum by other reference solutions. All these have their own drawbacks (cf. Kubiak [20]). Kallel and Schoenauer [16] suggested to use the best solution in the set instead of the global optimum to asses the FDC value, thus yielding a Local FDC (LFDC), due to the fact that the reference sample is only a local optimum. In the experiments, LFDC will be calculated based on 20,000 unique, randomly created individuals for each instance.

4.5 Optimization performance: EGO versus GA versus RS

Finally, we compare optimization performance. To that end, the Kriging based EGO, a model-free GA and Random Search (RS) are used to optimize the given problem instances under a strictly limited budget of up to 200 function evaluations. GA and RS are baselines in this comparison. Their main purpose is to identify whether the various EGO variants with different distance measures work or not. That means, any of the 15 EGO variants (14 distance measures, and a combination of all) that does not outperform RS or GA should be disregarded.

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. Basically, the very same GA is used to optimize the surrogate models in the EGO variants. However, due to the assumption that the surrogate model is much cheaper, the budget is increased to 10,000 function evaluations, and the population size is increased to 20. The EGO algorithm will start with an initial set of ten solutions.

For a fair comparison of actual competitors, all these parameters would require tuning. But since the basic GA is rather just a baseline, this is not necessary. Also, the various EGO variants use the exact same settings thus yielding a fair comparison among themselves.

5 Observations

5.1 Correlation, LFDC

5.2 Matrix condition

It could be observed, that for most measures, a positive semi-definite matrix could be determined for each n, but naturally not for each θ. The lower θ is, the

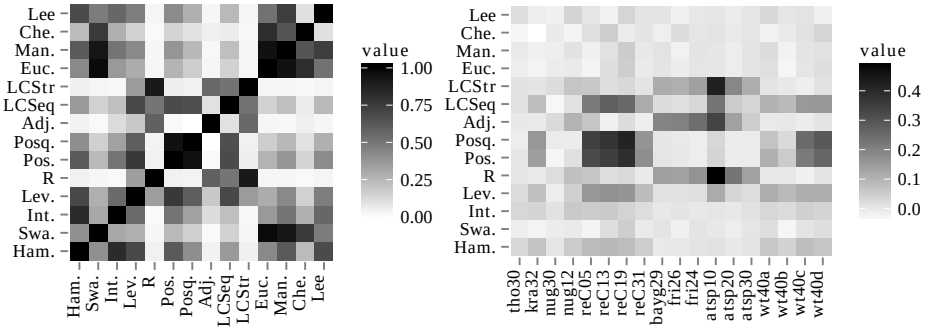

Fig. 2. Heatmap of correlation between distance measures (left). Heatmap with LFDC values (right).

more likely are correlation matrices that are not positive semi-definite. The exception was the adjacency distance measure. Here, for the two smallest problem sizes ($n = 5$, $n = 6$), none of the calculated correlation matrices was positive semi-definite. While these low numbers of N do not occur in any of the benchmarks used in this paper, one should keep in mind to avoid using adjacency distance when the number of samples may represent a large portion of the search space.

5.3 Optimization performance

Figure 3 shows the results of the optimization experiments, comparing the median of the final best values found. The distance measures (e.g. Lee., Ham.) indicate EGO with a Kriging model based on the respective measure.

EGO based on Chebyshev distance (Che.) is not shown, as it is the only distance measure to never outperform random search. Some structure can be discovered, regarding similarity of results for the different instances. Three main groups 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 of a group. These three blocks do coincide with the (more vague) structure that was visible in the LFDC results. At the same time, large LFDC values may not be directly associated with improved optimization performances. For instance, Pos. and Posq. had large LFDC values for the FSP instances, but never rank better than on the fourth position. Posq. Distance has especially bad optimization performance for reC13 and reC19, in spite of showing largest LFDC.

Overall, including all measures in the MLE process (*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. The single best distance measures is Hamming distance, yielding best results in 6 of the 18 test problems, but receiving much lower ranks for some of the other instances. Considering the various problem

types, the single best distance measures are: Ham. for QAP, Lev. for FSP, Adj. for TSP, R for ATSP, Pos. for WT. Random search performs worst, as expected, only the Chebyshev distance based approach has similar performance. The model-free GA is always outperformed by at least 5 EGO variants. Choosing the wrong distance measure may however yield to performance worse than the model-free GA.

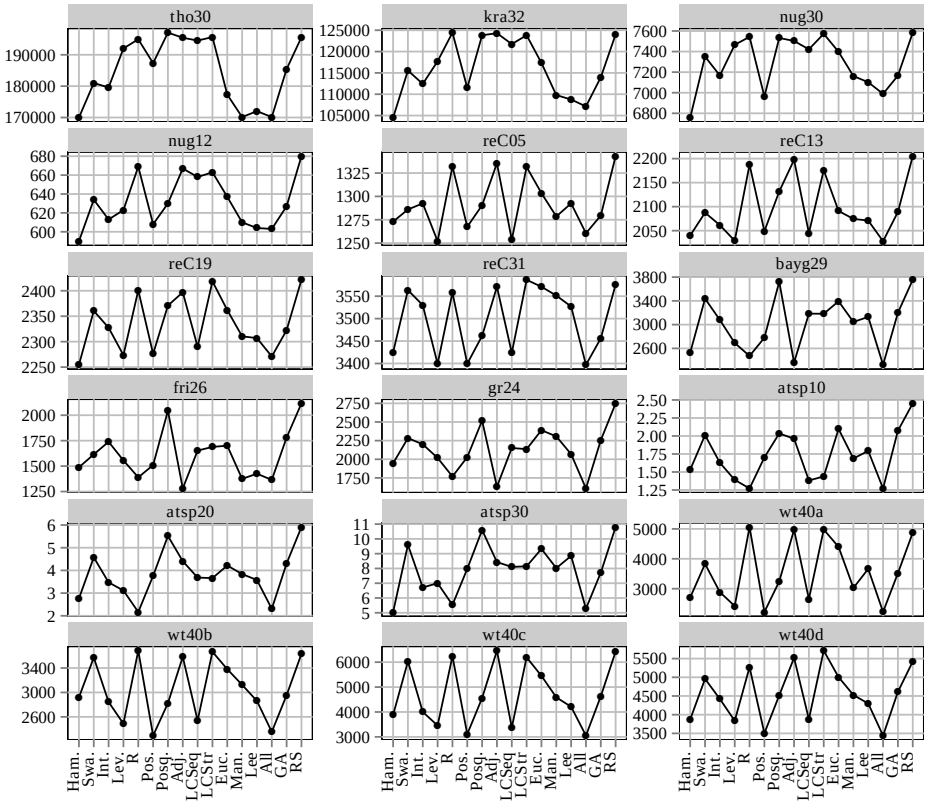

Fig. 3. Optimization performance: Median performance of each method and each problem instance. Smaller values are better.

6 Discussion

While it was shown that LFDC does expose some similar structure as the optimization performance, it is apparently unable to identity an well-suited measure for a specific problem. Thus, one should take great care when using such kind of indicators to select a measure. Even more so, as it was not even tested for

smaller sample sizes. Investing more effort into LFDC does seem unprofitable, considering these results.

Rather, the nice and robust performance of choosing a distance measure during 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. At the same time, the computational effort may make it necessary to restrict the set of distance measures used in this approach. In such cases, Hamming distance should always be in the set. It is the single best performing measure, and the cheapest, too.

7 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 type 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.

Research may also focus on learning distance function for Kriging-based models in combinatorial spaces. Learning of correlation (or kernel) functions with Genetic Programming is not new [10, 19]. Also, distance functions have been evolved with GA [6] in the context of string matching. Combining both ideas to evolve better distance functions 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.

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. T. Bartz-Beielstein, M. de Vegt, K. E. Parsopoulos, and M. N. Vrahatis. Designing particle swarm optimization with regression trees. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–173/04, Universität Dortmund, Germany, Mai 2004.
3. J. E. Beasley. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.
4. R. E. Burkard. Quadratic assignment problems. *European Journal of Operational Research*, 15(3):283 – 289, 1984.
5. R. E. Burkard, S. E. Karisch, and F. Rendl. QAPLIB – a quadratic assignment problem library. *Journal of Global Optimization*, 10(4):391–403, 1997.
6. D. Camacho, R. Huerta, and C. Elkan. An evolutionary hybrid distance for duplicate string matching. Technical report, 2008.
7. 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.

8. S. Fatima and A. Kattan. Evolving optimal agendas for package deal negotiation. In N. Krasnogor et al., editors, *Genetic and Evolutionary Computation Conference (GECCO'11), Proceedings*, pages 505–512. ACM, 2011.
9. A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling*. Wiley, 2008.
10. C. Gagné, M. Schoenauer, M. Sebag, and M. Tomassini. Genetic programming for kernel-based learning with co-evolving subsets selection. In T. Runarsson, H.-G. Beyer, E. Burke, J. Merelo-Guervós, L. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 1008–1017. Springer Berlin Heidelberg, 2006.
11. D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343, 1975.
12. F. Hutter. *Automated configuration of algorithms for solving hard computational problems*. PhD thesis, University of British Columbia, 2009.
13. Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011.
14. 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.
15. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.
16. L. Kallel and M. Schoenauer. Fitness distance correlation for variable length representations. Technical report, Technical Report 363, CMAP, Ecole Polytechnique, 1996.
17. A. Kattan and E. Galvan. Evolving radial basis function networks via gp for estimating fitness values using surrogate models. In *Congress on Evolutionary Computation (CEC'12), Proceedings*, pages 1–7. IEEE, 2012.
18. M. Kendall and J. Gibbons. *Rank correlation methods*. A Charles Griffin Book. E. Arnold, 1990.
19. G. Kronberger and M. Kommenda. Evolution of covariance functions for gaussian process regression using genetic programming. *CoRR*, abs/1305.3794:308–315, 2013.
20. M. Kubiak. *Fitness-Distance Analysis for Adaptation of a Memetic Algorithm to Two Problems of Combinatorial Optimisation*. PhD thesis, Poznan University of Technology, 2009.
21. C. Lee. Some properties of nonbinary error-correcting codes. *Information Theory, IRE Transactions on*, 4(2):77–82, June 1958.
22. 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 (CEC'08), Proceedings*, pages 2764–2771. IEEE, 2008.
23. A. Moraglio and A. Kattan. Geometric generalisation of surrogate model based optimisation to combinatorial spaces. In *Proceedings of the 11th European Conference on Evolutionary Computation in Combinatorial Optimization*, EvoCOP'11, pages 142–154, Berlin, Heidelberg, 2011. Springer-Verlag.
24. A. Moraglio and A. Kattan. Geometric surrogate model based optimisation for genetic programming: Initial experiments. Technical report, University of Birmingham, 2011.

25. 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 (GECCO'11), Companion*, pages 133–134. ACM, 2011.
26. C. R. Reeves. A genetic algorithm for flowshop sequencing. *Computers & operations research*, 22(1):5–13, 1995.
27. C. R. Reeves. Landscapes, operators and heuristic search. *Annals of Operations Research*, 86:473–490, 1999.
28. G. Reinelt. TSPLIB—A traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.
29. T. Schiavinotto and T. Stützle. A review of metrics on permutations for search landscape analysis. *Computers & operations research*, 34(10):3143–3153, 2007.
30. M. Sevaux, K. Sörensen, et al. Permutation distance measures for memetic algorithms with population management. In *Proceedings of 6th Metaheuristics International Conference (MIC'05)*, 2005.
31. E. Taillard. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47(1):65–74, 1990.
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 (GECCO'14), Proceedings*, 2014. Preprint (Accepted).