Efficient Global Optimization for Combinatorial Problems

Martin Zaefferer, Jörg Stork, Martina Friese,
Andreas Fischbach, Boris Naujoks, Thomas Bartz-Beielstein
Cologne University of Applied Sciences
51643 Gummersbach, Germany
[firstname].[lastname]@fh-koeln.de

ABSTRACT

Real-world optimization problems may require time consuming and expensive measurements or simulations. Recently, the application of surrogate model-based approaches was extended from continuous to combinatorial spaces. This extension is based on the utilization of suitable distance measures such as Hamming or Swap Distance. In this work, such an extension is implemented for Kriging (Gaussian Process) models. Kriging provides a measure of uncertainty when determining predictions. This can be harnessed to calculate the Expected Improvement (EI) of a candidate solution. In continuous optimization, EI is used in the Efficient Global Optimization (EGO) approach to balance exploitation and exploration for expensive optimization problems. Employing the extended Kriging model, we show for the first time that EGO can successfully be applied to combinatorial optimization problems. We describe necessary adaptations and arising issues as well as experimental results on several test problems. All surrogate models are optimized with a Genetic Algorithm (GA). To yield a comprehensive comparison, EGO and Kriging are compared to an earlier suggested Radial Basis Function Network, a linear modeling approach, as well as model-free optimization with random search and GA. EGO clearly outperforms the competing approaches on most of the tested problem instances.

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: Optimization—*Global Optimization*; G.2.1 [**Discrete Mathematics**]: Combinatorics—*Combinatorial Algorithms*

General Terms

Algorithms, Experimentation

Keywords

Genetic Algorithm, Surrogate Model-Based Optimization, Efficient Global Optimization, Kriging, Gaussian Processes, Distance Measure

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598282.

1. INTRODUCTION

Surrogate models are well established tools for cost reduction of time consuming or expensive simulation and optimization runs in the continuous domain. Surrogate models, as employed in methods like Efficient Global Optimization (EGO) [15] or Sequential Parameter Optimization (SPO) [3] offer several advantages. They reduce the number of function evaluations, give information on the shape of the optimized landscape and estimate effects and interactions between parameters.

Combinatorial optimization problems arise in many real-world settings. Examples for expensive combinatorial problems are scheduling or sequencing problems that depend on time consuming simulations (e.g., [30]). The application of surrogate-models to combinatorial problems is scarce. It is desirable to extend successful approaches from continuous optimization to combinatorial spaces.

The generalization of distance-based models to combinatorial spaces was earlier proposed by Moraglio and Kattan [20]. The core idea is to replace continuous distance measures (e.g., Euclidean) with distance (or similarity) measures native to a combinatorial problems representation, e.g., Hamming Distance (HD) or various edit distances. Relevant previous studies and similar work are reviewed in Sec. 2.

Moraglio and Kattan [20] use Radial Basis Function Networks (RBFN) in their studies, but also suggest to employ more complex models like Kriging. Kriging may provide an error estimate for each prediction, which can be exploited to calculate the Expected Improvement (EI) of a candidate solution. Thus, EGO can be applied to combinatorial optimization problems.

The following topics will be addressed in this paper:

1. extension of Kriging to combinatorial problems,
2. determination of the optimization performance of a Kriging-supported Genetic Algorithm (GA), and
3. investigation of the applicability of EGO to combinatorial optimization problems.

Methods to deal with these topics are introduced in Sec. 3. Numerical tests, which are performed on standard test problems, are described in Sec. 4. Experimental results are presented in Sec. 5 and discussed in Sec. 6. Finally, Sec. 7 gives a summary and an outlook.

2. PREVIOUS RESEARCH

2.1 Surrogate Models in Optimization

Generally speaking, a surrogate model $\widehat{\mathcal{M}}$ is a (coarse grained or cheap) model that replaces a (fine grained or ex-

pensive) model $\mathcal{M}$ with higher complexity. The reader may consider a Computational Fluid Dynamics (CFD) model that replaces a real-world problem. This CFD model itself can be replaced by a simplified analytical model. In the former case, CFD models are considered as surrogates, whereas in the latter, they are considered as fine grained models that are replaced by a surrogate.

In this paper, however, the term surrogate is restricted to data-driven models. They replace the simulation model $\mathcal{M}$, which is given by a function f, see Algorithm 1. Here, it is assumed that function evaluations clearly dominate the time consumption (or cost), i.e., most time is spent in line two and six of Algorithm 1. Stopping criteria can be a given budget of function evaluations, a specified time limit or a fitness value to be reached. The set A_p contains all underlying parameters of Algorithm 1, e.g., number of initial solutions, type and parameterization of the search strategy (line 5) or the type of surrogate model.

Algorithm 1: Surrogate model-based optimization

Input: Function f, stopping criteria, parameter set A_p
Output: Best solution found y^*, final model $\widehat{\mathcal{M}}^*$
1 Create initial solutions (randomly or with design of experiment);
2 Evaluate solutions with f;
3 **while** *Stopping criteria not reached* **do**
4 Build/update $\widehat{\mathcal{M}}$;
5 Find best solution(s) predicted by $\widehat{\mathcal{M}}$;
6 Evaluate solution(s) with f;
7 **end**

Surrogate models can be Artificial Neural Networks, Linear Models, Kriging, Multivariate Adaptive Regression Splines, Random Forests, and many more. Jones et al. [15] introduced EGO, which uses the predicted mean and variance provided by a Kriging surrogate to compute the Expected Improvement (EI) of a candidate solution. Without loss of generality, we will consider the case of minimization. Following the notation in [10], we consider an expensive function f and a related surrogate of f, denoted $\hat{f}$. Gaussian process models allow the determination of the mean squared error $\hat{s}^2(\mathbf{x})$ as described in [26]. Let y^* denote the best found function evaluation so far, $\Phi(\cdot)$ and $\phi(\cdot)$ denote the cumulative distribution function and probability density function, respectively. If $\hat{s}(\mathbf{x}) > 0$, then the expected improvement can be determined as

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

otherwise $\mathrm{EI}(\mathbf{x}) = 0$. EI determines how much improvement can be expected from the candidate solution to be predicted. Thus, EGO uses EI instead of the mean prediction to determine a promising candidate solution in line 5 of Algorithm 1. Besides saving evaluations of the expensive function f, this approach also provides an infill criterion (i.e., EI) that balances exploitation versus exploration. For a more detailed summary on surrogate model-based numerical optimization we refer to the overview by Jin et al. [13]. A statistical framework for model-based optimization is provided by the Sequential Parameter Optimization [3].

2.2 Surrogate Models in Combinatorial Optimization

Combinatorial surrogate models are a relatively new research topic [14]. Data-driven surrogate models were used in combination with GA or Ant Colony Optimization (ACO), however, mostly for optimization of continuous vectors, where classical models are applicable. Methods like Estimation of Distribution Algorithms or ACO can also be understood to use models [32], e.g., the Bayesian network model in the Bayesian Optimization Algorithm [23] or the pheromone model in ACO.

Another branch of combinatorial surrogate-model applications comprehends solvers for specific problem representations and specific applications. Voutchkov et al. [30] optimize a welding sequence. To represent the combinatorial problem, a signed permutation is used. The surrogate model replaces an expensive Finite Element (FE) model by estimating the influence of each individual element in the sequence, based on the observations made in previously tested sequences. Their surrogate model uses not only the resulting fitness values. It also exploits intermediate results that reflect impact of individual elements, depending on their position in the sequence. Exploiting such intermediate results will give this model a clear advantage over the more simple, fitness-value driven approaches. On the other hand, the applicability of this model is restricted to this specific setup and cannot be transferred to other application areas.

Fonseca et al. [9] defined Similarity-Based Models (SBM) as models that keep a memory of solutions and estimate the performance of new samples by comparing them to that memory. Fonseca et al. list Fitness Inheritance [29], Fitness Imitation [18, 13] and k-Nearest Neighbor (k-NN) [2] as examples. They test a GA supported by a k-NN model on a set of numerical, continuous test functions. Bernardino et al. [5] perform similar tests with Artificial Immune Systems. In both cases Hamming and Euclidean Distance are used as measures of similarity, showing that this approach does not depend on a specific problem representation. In this study, we do not employ the SBM variants suggested by Fonseca et al. [9], because the proposed models are not suited to predict a new optimum. For example, the k-NN model would never predict that a candidate solution has better performance than the best known solution. The k-NN model may be useful in a model management algorithm [9, 5]. However, it is not useful in a framework as outlined in Algorithm 1.

More suited towards our goal are the approaches of Li et al. [19] and Moraglio and Kattan [20]. They use distance-based models, which are able to predict promising, new solutions. Section 2.3 will review related results. Also, we introduce a model similar to k-NN but more suited to the given purpose in Sec. 3.2.

2.3 Applying Continuous Surrogates in Combinatorial Search Spaces

Li et al. [19] proposed RBFN models for optimization in non-Euclidean spaces by replacing the employed distance measure. Their RBFN models were applied to mixed-integer problems, using a mixed integer evolution strategy. Another approach to a mixed problem is taken by Hutter [12], who describes a Kriging model based on a weighted Hamming distance to model categorical variables for algorithm tuning. In a very similar way, Moraglio and Kattan [20] suggested a

generalization of distance-based models from continuous to combinatorial spaces.

The core idea is to employ distance measures, which are inherent to the combinatorial problem representation (e.g., Edit Distance). Such problem representations can be binary strings (e.g., binary knapsack problem, NK-Landscapes) permutations (e.g., assignment and scheduling problems) trees (e.g., symbolic regression) or any non-standard combinatorial problem representation.

Moraglio and Kattan [20] demonstrated this with a RBFN adapted to arbitrary distance measures to solve instances of NK-Landscapes (NKL). This RBFN-based approach has also been applied to the Quadratic Assignment Problem (QAP) [22], package-deal negotiation [8] and tree-based problems from Genetic Programming (GP) [21]. GP has also been coupled with the RBFN-based approach to evolve better discrete surrogate models [16]. All those works employ some form of RBFN based on arbitrary distance measures, thus adapted to combinatorial spaces. As Moraglio and Kattan [20] indicate, this can also be done with other models, e.g., with Kriging.

The key issue is to replace the Euclidean (RBFN) or per-variable (Kriging) distances by distance measures, which directly work for the inherent problem representation. Depending on the model type under consideration, other changes may become necessary. For instance, in the context of arbitrary distance measures, there is no guarantee that a given distance matrix will be invertible, as required by RBFN. Therefore, Moraglio and Kattan [20] suggested to replace the matrix inversion with the pseudoinverse. This issue is revisited in the following description of the Kriging model employed in the herein described work.

3. METHODS

3.1 Kriging for Combinatorial Problems

Kriging is a method for interpolation and regression based on Gaussian process modeling. The following notation is adopted from Forrester et al. [10]. 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 [10]:

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

Equation (1) defines a non-Euclidean distance measure, which uses a weighted per-element distance. The weights θ_j and the shape parameter p_j have to be estimated. The matrix that collects correlations of all pairs $\{(i,l)\}$ is called the correlation matrix $\boldsymbol{\Psi}$. It is used in the Kriging predictor

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

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

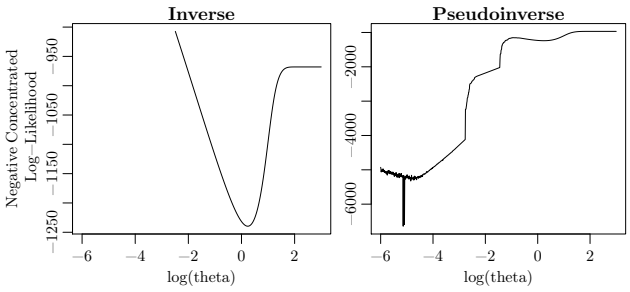

Figure 1: Negative concentrated log-likelihood plotted versus $\log(\theta)$. Likelihood landscape for a Kriging model, based on 200 randomly sampled solutions of a $N = 10$, $K = 2$ NKL-Landscape. Standard matrix inversion is compared to pseudoinverse. Values not plotted in the left plot represent (close to) singular correlation matrices. Both plots show a (local) optimum at $\log(\theta) \approx 0.24$.

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

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

The width parameter $\boldsymbol{\theta}$ determines how far the influence of each sample point $\mathbf{x}$ spreads. If the correlation structure differs in different directions of the search space, fitting different θ_j values for each direction of the search space is desirable. This is the so-called anisotropic case. Isotropic models are better suited for combinatorial search spaces, because direction is a vague concept for combinatorial optimization problems. Therefore, Eq. (1) is transformed to become isotropic, i.e., with scalar θ and p, i.e.:

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

where $\mathrm{d}(\cdot)$ can be any distance measure for the given problem representation. Now, the samples $\mathbf{x}$ are not restricted to continuous values and may consist of various types, e.g., binary strings, permutations, or trees.

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. A non degenerated or positive-semidefinite matrix is required for this inversion. Moraglio and Kattan [20] state that it can not be guaranteed that such matrices will still be invertible if they are based on an arbitrary distance measure. Therefore, they replace the matrix inversion in their RBFN model with the pseudoinverse.

In case of Kriging, this can introduce a new global optimum into the MLE landscape. Selecting θ based on such an optimum may lead to a poor prediction at new sample points. This situation is exemplified in Fig. 1. Here, a Kriging model is built, based on 200 random samples from a NK-Landscape ($N = 10$, $K = 2$), see Sec. 4.1. The Hamming distance (HD) metric is employed as a distance measure. The likelihood is calculated once with standard matrix inversion, once with the pseudoinverse. The figure shows the dependency of the estimated likelihood value on the θ parameter. Clearly, when θ becomes too small, the correlation matrix will become close to singular. Hence, no values are plotted for this region (Fig. 1, *left*). In practice, a penalty function is used to handle this problem. The pseudoinverse generates extremely good likelihood values in this region.

While both plots have an optimum at approx. 0.24, it is a local one when using pseudoinverse. Thus, it is disregarded during MLE with pseudoinverse. Instead, very small θ values would be chosen. As mentioned earlier, θ will control how far the influence of each observation will spread in the search space. With very small θ, all observations affect the whole search space equally. Here, this leads to very bad predictive performance. On the other hand, the local optimum ($\theta \approx 0.24$) results in a surrogate model $\widehat{\mathcal{M}}$ that is highly correlated with the expensive model $\mathcal{M}$ at unseen sample points. This example shows that standard matrix inversion works well with HD, while pseudoinverse does not. This is not surprising, as the HD metric was successfully applied in related contexts (e.g., [24, 25]).

Of course, there may be distance measures where the situation is more complicated. If a distance measure does not yield valid (i.e., non-degenerate or even positive-semidefinite) correlation matrices, two solutions are possible. First, MLE may be replaced with a cross-validation approach. This may increase the computational burden. Second, the correlation function (4) or the distance measure may be adapted to guarantee that the correlation matrix is valid. As a first step, this paper only relies on the assumption that all employed distance measures are valid.

3.2 A Linear Model

As a simple base-line comparison, a Linear, distance-based Model (LM) is introduced. To predict the quality of a new solution, all existing samples are sorted according to their distance to the new sample. The average of the observations at the smallest and second smallest distance can be used to estimate a linear trend, i.e., slope and intercept. The intercept is the prediction for the utility value of the new sample. This approach may be compared to a k-NN model with $k = 2$, with the difference that the prediction is not based on the mean of the two nearest neighbors, but rather on a linear trend estimated from these. Thus, it may suggest new optimal solutions, which k-NN can not.

3.3 Radial Basis Function Network

The employed RBFN model is based on the description by Moraglio and Kattan [20], with the predictor,

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{K} w_i \exp\left(-\beta \mathrm{d}(\mathbf{x}, c^{(i)})^2\right)$$

where K is the number of centers $\mathbf{c}$ (here: all samples in $\boldsymbol{X}$ are centers), $\mathrm{d}(\cdot)$ is an arbitrary distance measure and w_0 is the mean of all observations. The weights $\mathbf{w}$ are determined with the pseudoinverse, $\mathbf{w} = \boldsymbol{G}^{+}(\boldsymbol{y} - \mathbf{1}w_0)$. Here, the matrix $\boldsymbol{G}$ has the elements $g_{ij} = \exp\left(-\beta \mathrm{d}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})^2\right)$. Furthermore, D denotes the maximum distance and $\beta = 1/2D^2$. The RBFN will also be used in an EGO framework (hence called EGOR). To that end, we can get a rough error estimate in a similar way as done with Kriging, i.e., from Eq. (3), where $\boldsymbol{\Psi}$ is replaced by $\boldsymbol{G}$ and $\hat{\sigma}$ is replaced by the standard deviation of the observations.

4. EXPERIMENTAL SETUP

4.1 Test Problems

The experiments in this paper are based on those in two previous studies [20, 22]. All experiments are performed using the free software environment for statistical computation, `R`. Function evaluations are assumed to be expensive, dominating the cost of the optimization process. This assumption is made to justify the large computational overhead of surrogate model training and exploitation. Hence, strictly limited budgets are imposed during the experiments.

NK-Landscapes (NKL), as proposed by Kauffman [17], are fitness landscapes based on binary strings. The fitness of a string is the sum of fitness contributions of N string elements, each impacted by K other elements. The fitness of a binary string is therefore given by (cf. [1])

$$f(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} g_i(x_i; x_{i1}, ...x_{iK}), \tag{5}$$

where x_i is the i-th bit of the string $\mathbf{x}$, and $x_{i1}, ...x_{iK}$ are the bits that influence the contribution of x_i. A cyclic order is used, i.e., x_1 follows x_N. For each string element, a function g_i assigns a real-valued weight to each possible combination of the element and its neighbors, typically sampled uniformly from $[0; 1]$. This results into N lookup tables of $2^{(K+1)}$ values. In this paper, the K neighbors that impact the contribution of the i-th element x_i are given by the sequence $(x_{i+1}, ..., x_{i+K})$.

The **Quadratic Assignment Problem** (QAP) [6] describes a permutation problem, where N facilities have to be assigned to N locations. Assignment cost is minimized, based on flow between facilities (a) and distance between locations (b). The optimization problem is to find an optimal permutation π of length N from the set of all permutations Π_N, that is

$$\min_{\pi \in \Pi_N} \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} b_{\pi(i),\pi(j)}. \tag{6}$$

The instances `nug30`, `tho30` and `kra32` from the QAP Library (QAPLIB) [7] were chosen in [22]. Here, instance (`nug12`) was added to incorporate smaller search spaces.

Unimodal (UNI) problems were suggested by Moraglio et al. [22] as simple and transparent test cases. Here, the fitness of a permutation is its distance to the fixed reference permutation $\pi = 1, 2, 3, ..., 30$. Both HD and Swap Distance (SD) are used, each forming a different UNI instance (`unih30`, `unis30`).

Because the look-up tables are randomly generated, our results differ from previous results presented in [20, 22]. Also, the exact distribution of the K neighbors may differ. The UNI problems depend on the chosen reference permutation. The QAP instances should be identical, thus yielding the most comparable results. All problems are of a low or moderate sizs, e.g., the permutations are no longer than 32 elements. Of course, larger problems may be handled as well. Still, larger problems would require much more data to build reasonable models. In practice, even small problem instances may be hard to solve when evaluation becomes expensive, e.g., Voutchkov et al. [30] only consider signed permutations of length six. Hence, we chose rather small or medium sized problem instances.

4.2 Surrogate Models

Three surrogate models are employed in the experiments, RBFN, Kriging, and LM. No tuning of model parameters is performed. EGO will be performed with Kriging or RBFN (EGOR). The Kriging implementation is based on the origi-

nal Matlab code by Forrester et al. [10], as reimplemented in the SPOT R package. It is adapted to combinatorial spaces as described in Sec. 3.1. The parameter p is fixed at a value of one, all others are estimated with MLE. RBFN and LM are implemented in R, according to their description in Sec. 3.2 and 3.3.

4.3 Optimization Algorithms

Two model-free optimization algorithms are employed, Random Search (RS) and GA. RS will only be employed to optimize the test functions, namely QAP, NKL, and UNI. This provides a reference performance which should be beaten by any more sophisticated algorithm. The GA will be used on all problems and with all surrogate models.

In all cases, the crossover rate is 0.5, the mutation rate $\frac{1}{N}$. Tournament selection is performed with a tournament size of two and a probability 0.9. For NKL, bitwise mutation and uniform crossover are used. In case of the permutation problems (UNI, QAP) interchange mutation (i.e., interchange of arbitrary elements) and cycle crossover are chosen.

When the NKL instances are optimized directly, the population size is N. When a NKL surrogate model is optimized, population size is $10N$. For QAP and UNI, the population size is set to 10 (direct) and 20 (surrogate), respectively. The number of function evaluations for NKL is N^2 (direct) and $100N^2$ (surrogate), respectively, and the number of function evaluations for UNI and QAP is set to 100 (direct) and $10,000$ (surrogate), respectively.

The GA is combined with five different model-based approaches (RBFN, LM, Kriging, EGO with Kriging and EGOR with RBFN). For RBFN, LM and Kriging, the exploration strategy used by Moraglio and Kattan [20] is employed: a random solution is selected for evaluation, if the model does not predict a value better than the best known solution. The EGO variants do not need this explicit exploitation mechanism, because they use a natural way to balance between exploration and exploitation.

To enable a more coherent experimental setup, the use of a memetic GA with a two-opt step to optimize the surrogate model for QAP and UNI (cf. [22]) is omitted in our setup. Preliminary experiments indicated that additional local search does not improve the results significantly. This may be due to the accuracy of the surrogate models. Since their prediction is not perfectly exact, exhaustive local search may be unprofitable. Also note, that duplicates are avoided, with respect to the restricted budgets.

Of course, the rather simple GA is a potentially weak competitor, the RS even more so. Future work may replace the model-free GA with a more potent, state-of-the-art competitor. Still, a state-of-the-art approach may be hard to find for the case of strictly limited budgets.

4.4 Distance Measures

Several distance measures are discussed in the literature. Schiavinotto and Stützle [27] compared several distance metrics for search landscape analysis, concerning permutations. A different set is reviewed by Sevaux et al. [28] for their usefulness in a diversification strategy of a memetic algorithm. As shown for QAP, even phenotype information can be used for calculating distances [4]. For the purpose of measuring distance between binary strings or permutations, we consider the following distance measures.

Hamming Distance (HD) The number of unequal elements between two strings $\mathbf{x}$ and $\mathbf{y}$, i.e.,

$$HD(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} a_i \quad \text{where} \quad a_i = \begin{cases} 0 & \text{if } x_i = y_i, \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

The Hamming distance fulfills the conditions of a metric on the vector space of the words of length n. It can quickly determined, especially for binary strings and is therefore used in the NKL experiments. It is expected that the HD works well when modeling NKL data, as it very much resembles the distance measure usually used in RBFN or Kriging for continuous problems. In detail, $\sum_{i=1}^{n} |x_i - y_i|$ yields values identical to Eq. (7) if the two binary strings $\mathbf{x}$ and $\mathbf{y}$ are interpreted as numerical vectors containing zeros or ones. Equation (7) can be used to measure the distance between arbitrary strings of equal length. Hence, it can also be applied to permutations. There, two other distance measures are employed in the experiments.

Swap Distance (SD) A swap operation is the interchange in position of two adjacent elements in a permutation. SD counts the minimal number of swaps necessary to transform one permutation into another. For the calculation of this measure, we use the algorithm as described by Schiavinotto and Stützle [27]. Moraglio et al. [22] report that SD performed poorly. This may be due to the fact, that SD only concerns adjacent elements, while interchanging two arbitrary elements may be a more reasonable smallest step. This gave the inspiration to add an additional distance measure to our portfolio.

Interchange Distance (ID) An interchange operation is the interchange in position of two arbitrary elements in a permutation. ID counts the minimal number of interchanges required to transform one permutation into another. Schiavinotto and Stützle [27] provide an algorithm to calculate this measure, which is employed in this work. They also performed experiments to measure correlation between different distance measures and report a high correlation between HD and ID, whereas the correlations between HD and SD, as well as between SD and ID are low. Based on this information, ID is an interesting candidate in our portfolio. It has to be noted that HD has the lowest computational complexity. In case of equal performance, HD should be selected.

5. RESULTS

Experimental results are visualized by boxplots in Fig. 2, 4 and 5. The inner box indicates the 25 and 75 % quartiles and the bold line is the median. Circles are outliers, and the outmost lines specify the range of values exluding these outliers. For the two instances with $K = 2$, EGO finds the optimum in all of the 20 runs. EGOR (with RBFN) performs similarly well, but does not find the optimum in every run on the $K = 2, N = 25$ instance. In General, EGO outperforms EGOR. For $N = 25$, $K = 5$, only EGO and EGOR ever solve the problem, but not in all runs (EGO: 7 of 20, EGOR: 1 of 20). EGO is clearly best on all NKL instances but the instance with $N = 10$, $K = 5$. To analyze this behavior further, the runs with $N = 10$, $K = 5$ are extended to 300 function evaluations. The corresponding results are visualized in Fig. 3. Here, no decision can be made for small budgets. EGO performs better for budgets > 100. The model-free GA is clearly outperformed on the NKL instances by RBFN and EGO. LM fails to outperform

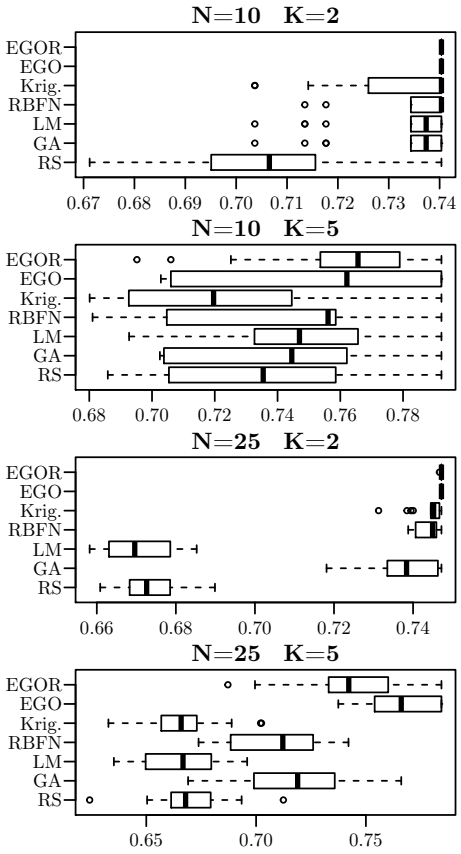

Figure 2: Boxplot of NKL results. All modeling approaches employ HD only. Larger values are better.

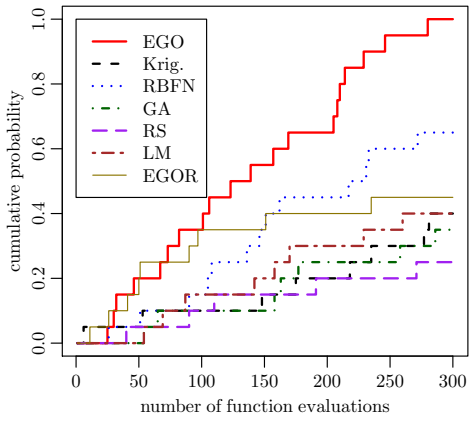

Figure 3: Empirical runtime distribution plot for the NKL instance ($N = 10, K = 5$). Y-axis shows the fraction of the 20 runs which reached the optimum.

the basic RS for the two larger NKL instances. Optimization with Kriging mean predictions often performs worse than RBFN or LM, with the exception of the $N = 25$, $K = 2$ NKL instance.

EGO is best on all QAP instances, but only with HD. Other distance measures may or may not lead to performance that is worse than the model-free GA, which often ranks second best to EGO. On the QAP instances, SD is sometimes better than ID, sometimes vice-versa.

For `unis30` the GA outperforms only RS and LM, as well as Kriging with HD. For `unih30`, GA is outperformed by EGO, EGOR and RBFN with HD. HD seems to work best on all tested permutation problems, with exception of `unis30`. Optimization with Kriging mean predictions performs better than RBFN for `unis30`. With EI, on the other hand, Kriging (EGO) performs nearly always better than RBFN (EGOR).

Overall, EGO clearly performs best, often (but not always) followed by RBFN or EGOR. Of the model-based approaches without EI, RBFN is best, although it is outperformed by the model-free GA several times. None of the permutation problems are ever solved with the given budgets, although EGO comes very close for `unis30` and `nug12`.

6. DISCUSSION

The surrogate model-based approaches work very well for NKL with $K = 2$. Results in general were much better than for the permutation problem experiments. Two reasons for this excellent performance can be given. First, the search space of the tested NKL instances is much smaller than that of the permutation problems. E.g., the largest NKL ($n = 25$) has $2^n \approx 3.36 \times 10^7$ possible combinations, whereas the smallest QAP problem (n=12) has $n! \approx 4.79 \times 10^8$. Second, the choice of distance measure (HD) is not only natural to this problem, but is a measure which is similar to the distance measures used in continuous domains.

Relative to RS, performance of most algorithms decreased with larger K. This has to be expected to some extent, since such landscapes are clearly more rugged and difficult. That is, the larger K is, the more fitness contributions change when one single bit is flipped. Clearly, this does not only make the optimization more difficult, but also the modeling, since the correlation between neighboring solutions decreases.

For $N = 10$ and $K = 5$, the results showed large variances and the worst EGO performance. Additionally, the GA performance was not significantly better than RS. This can be attributed to the more rugged fitness landscape. The high variance in the performance of all approaches also suggests that more function values are required, which can be supported by earlier work on NKL. The dynamic programming algorithm proposed by Weinberger [31] solves the NKL in $O(2^K N)$ steps. That is, it needs 40 ($N = 10, K = 2$), 320 ($N = 10, K = 5$), 100 ($N = 25, K = 2$) or 800 ($N = 25, K = 5$) steps. As can be seen, all but the $N = 10$ and $K = 5$ instance received budgets rather close to these numbers, which may be a reason for the observed performance. Longer runs with 300 evaluations revealed that after sufficient evaluations, EGO would outperform the other approaches and reliably solve the instance.

For the permutation problems, EGO is the best working optimization approach, and HD the most suitable distance measure. It is interesting to observe that SD sometimes outperforms ID, although ID was reported to have larger correlation with HD, which worked best. This result should be investigated further. In contrast to results from Moraglio et al. [22], the model-free GA often outperforms the RBFN-supported GA on the permutation problems. The performance of the RBFN model seems to be very similar to the performance reported earlier. Thus, the difference may be in the choice of settings for the basic, model-free GA, which performs better than reported in the earlier study. E.g., the choice of using tournament over truncation selection may be

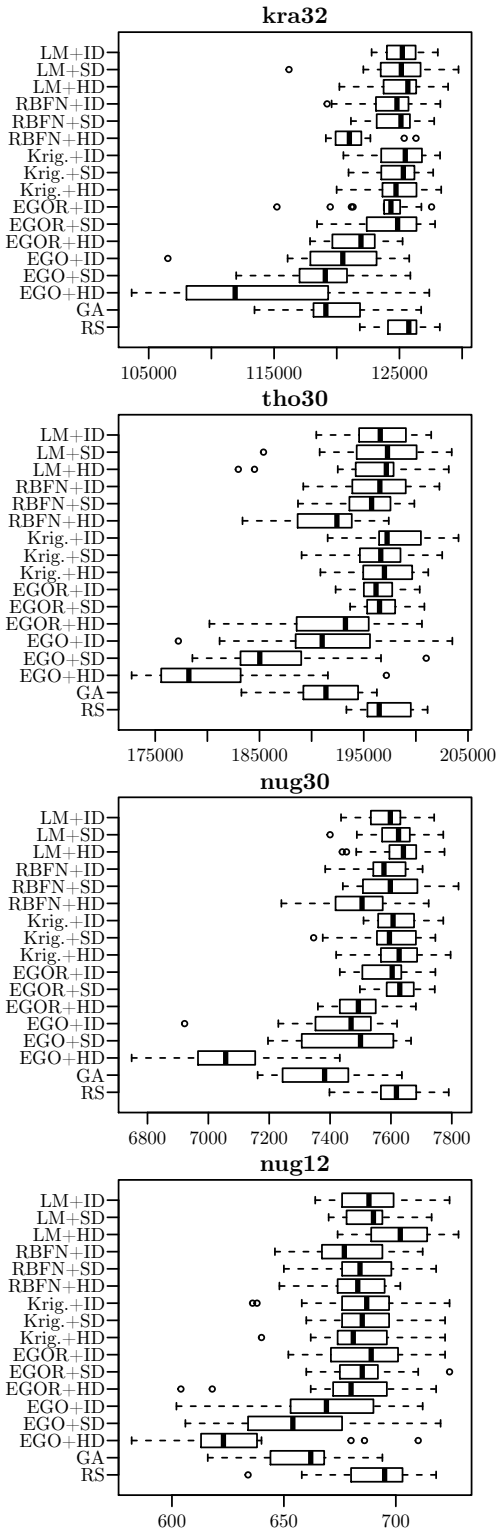

Figure 4: Boxplot of QAP results. Smaller values are better.

the reason. Or else, there may be differences due to the mutation operator, as interchange mutation was used instead of swap mutation. This result stresses the need for a future study on tuning of the applied approaches.

A further difference to earlier results [22] is the performance observed for the unimodal functions. Here, the best

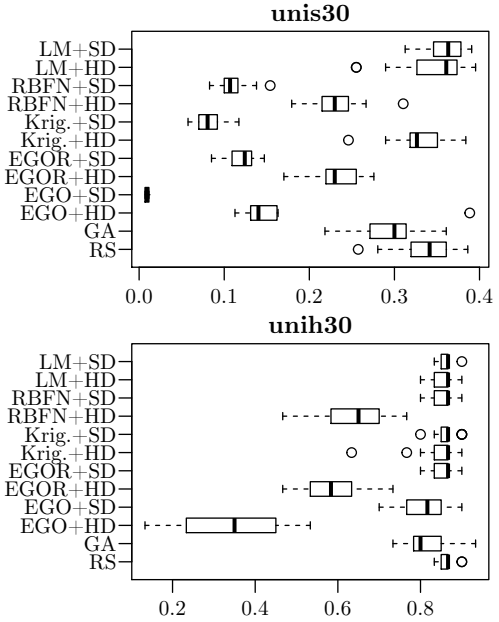

Figure 5: Boxplot of results for the artificial, unimodal test instances. Smaller values are better.

working distance measure is always that which is used in the instance itself. That is, HD works best with `unih30` and SD works best with `unis30`. The bad overall performance on `unih30` may be caused by the larger number of non-unique fitness values. Our results indicate that HD is clearly the best metric for the given problems. But the results on `unis30` show that the choice of distance measure can be affected by the problem type. This warrants further research with problem instances of larger variety. For other permutation problems, completely different measures may be preferable.

7. SUMMARY AND OUTLOOK

We demonstrated that EGO can be successfully applied to combinatorial optimization problems and that this Kriging-based approach was able to outperform a model-free GA. However, EGO does not make approaches like GA superfluous, because GAs are very useful for the proposed extension of EGO to combinatorial spaces. Searching for the solution with largest EI takes usually place in a multi-modal landscape. EI depends on the variance estimate which is zero at known samples. In between samples, this will often create local optima. Finding the global optimum solution of the EI landscape requires a surrogate-optimizer that is able to escape such local optima. Stochastic, population-based methods like GA are most suitable for this purpose.

Besides EGO and GA, model-based searches with Kriging (without EI), RBFN and LM were included in the comparison. All model-based approaches were outperformed by EGO, in some cases even by the model-free GA. It was also observed that RBFN does hold on very well to the basic Kriging model, as long as they are not used in EGO. Kriging outperformed the more simple RBFN when it was employed in EGO. The exploration/exploitation balancing of EGO in combination with the more powerful Kriging predictor and a well chosen distance measure seem to make the difference. As other model types do not provide an error estimate, heuristics may be examined in future research. E.g.,

variance estimated from distances to known samples, or by using some form of cross-validation.

Furthermore, the permutation distance measures were revealed to have a strong impact on the results. With one exception, Hamming Distance worked best for all problem instances. For other problems, the situation may be different. At the same time, the number of possible distance measures is much larger than the set used in this work. This issue will again occur, when tree-based representations or more exotic representations are concerned. Hence, the question of choosing a distance measure will remain an important issue. Distance measures were previously used for several tasks, e.g., diversity preservation in GAs or fitness landscape analysis. On the other hand, considering expensive problems yields different limitations than those encountered in earlier studies. For example, measures previously disregarded because of their complexity [28] may be of use in contexts where overall time consumption is dominated by the expensive target function. The interaction of distance measure, the correlation functions, and other parameters of the model are of interest for further investigation. The following topics will also be subject of our future research:

- performing a detailed study on parameter tuning of all compared approaches, to guarantee a fair comparison and understanding interactions of parameters
- implementing Co-Kriging [11] for combinatorial spaces, to include cheaply available data into the optimization process.

Acknowledgments.

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

8. REFERENCES

[1] L. Altenberg. *NK Fitness Landscapes*, chapter B2.7.2, pages B2.7:5–B2.7:10. IOP Publishing Ltd and Oxford University Press, 1997.

[2] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, Aug 1992.

[3] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuß. Sequential parameter optimization. In B. McKay et al., editors, *Congress on Evolutionary Computation (CEC'05), Proceedings*, pages 773–780. IEEE, 2005.

[4] A. Beham, E. Pitzer, and M. Affenzeller. A new metric to measure distances between solutions to the quadratic assignment problem. In *International Symposium on Logistics and Industrial Informatics (LINDI'11), Proceedings*, pages 45–50. IEEE, 2011.

[5] H. S. Bernardino, L. G. Fonseca, and H. J. C. Barbosa. *Surrogate-Assisted Artificial Immune Systems for Expensive Optimization Problems*, chapter 10, pages 179–198. InTech, Oct 2009.

[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] 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] L. Fonseca, H. Barbosa, and A. Lemonge. A similarity-based surrogate model for expensive evolutionary optimization with fixed budget of simulations. In *Congress on Evolutionary Computation*, pages 867–874. IEEE, 2009.

[10] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling*. Wiley, 2008.

[11] A. I. Forrester, A. Sóbester, and A. J. Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269, Dec 2007.

[12] F. Hutter. *Automated configuration of algorithms for solving hard computational problems*. PhD thesis, University of British Columbia, 2009.

[13] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.

[14] Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011.

[15] 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.

[16] 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.

[17] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, 1993.

[18] H.-S. Kim and S.-B. Cho. An efficient genetic algorithm with less fitness evaluation by clustering. In *Congress on Evolutionary Computation (CEC'01), Proceedings*, volume 2, pages 887–894. IEEE, 2001.

[19] R. Li, M. T. M. Emmerich, J. Eggermont, E. G. P. Bovenkamp, T. Bäck, 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.

[20] 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.

[21] A. Moraglio and A. Kattan. Geometric surrogate model based optimisation for genetic programming: Initial experiments. Technical report, University of Birmingham, 2011.

[22] 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.

[23] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz. BOA: The bayesian optimization algorithm. In W. Banzhaf et al., editors, *Genetic and Evolutionary Computation Conference*, pages 525–532. Morgan Kaufmann, 1999.

[24] J. C. Platt, C. J. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a gaussian process prior for automatically generating music playlists. In *NIPS*, pages 1425–1432, 2001.

[25] P. A. Romero, A. Krause, and F. H. Arnold. Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201, 2013.

[26] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[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. Sevaux, K. Sörensen, et al. Permutation distance measures for memetic algorithms with population management. In *Metaheuristics International Conference*, 2005.

[29] R. E. Smith, B. A. Dike, and S. A. Stegmann. Fitness inheritance in genetic algorithms. In *Proceedings of the 1995 ACM Symposium on Applied Computing*, SAC '95, pages 345–350, New York, NY, USA, 1995. ACM.

[30] 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.

[31] E. D. Weinberger. Np completeness of kauffman's n-k model, a tuneable rugged fitness landscape. Working Papers 96-02-003, Santa Fe Institute, Feb. 1996.

[32] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo. Model-based search for combinatorial optimization: A critical survey. *Annals of Operations Research*, 131(1-4):373–395, 2004.