A Model-Based Framework for Black-Box Problem Comparison Using Gaussian Processes

Sobia Saleem[1], Marcus Gallagher[1(✉)], and Ian Wood[2]

[1] School of Information Technology and Electrical Engineering, University of Queensland, Brisbane 4702, Australia
{s.saleem,marcusg}@uq.edu.au
[2] School of Mathematics and Physics, University of Queensland, Brisbane 4702, Australia
i.wood1@uq.edu.au

Abstract. An important challenge in black-box optimization is to be able to understand the relative performance of different algorithms on problem instances. This has motivated research in exploratory landscape analysis and algorithm selection, leading to a number of frameworks for analysis. However, these procedures often involve significant assumptions, or rely on information not typically available. In this paper we propose a new, model-based framework for the characterization of black-box optimization problems using Gaussian Process regression. The framework allows problem instances to be compared in a relatively simple way. The model-based approach also allows us to assess the goodness of fit and Gaussian Processes lead to an efficient means of model comparison. The implementation of the framework is described and validated on several test sets.

1 Introduction

A continuous black box optimization problem is defined as:

$$\min f(\mathbf{x}),\ \mathbf{x} \in S \subseteq I\!R^n \tag{1}$$

where $f()$ is the objective or fitness function and S is the feasible search space. It is assumed that the form of $f()$ is unknown but can be evaluated at any feasible candidate solution. Many real world problems can be formulated in this way and metaheuristic algorithms are specifically developed for this class of problems. The performance of an algorithm instance on a problem instance depends on how well the assumptions made by the algorithm suit the structure of the problem fitness landscape. Exploratory/fitness landscape analysis [10] aims to develop landscape features for understanding black-box problems, based on a sample of candidate solutions. A variety of different features have been proposed to measure different properties of problem landscapes [14]. If we are able to effectively characterize and compare problem instances, it should enable a

A. Auger et al. (Eds.): PPSN 2018, LNCS 11102, pp. 284–295, 2018.
https://doi.org/10.1007/978-3-319-99259-4_23

better understanding of algorithms and facilitate automated algorithm selection and configuration.

Evaluating the effectiveness of exploratory landscape analysis features and their role in the algorithm selection techniques is a nontrivial experimental challenge. While a number of procedures have been explored in the literature, they can be complex, involve the calculation of a set of landscape features and multiple stages of analysis that may require information not generally available for new problems.

In this paper we propose a model-based framework for continuous black-box problem comparison using Gaussian process (GP) regression. The advantages of this framework are that a flexible model is used, the accuracy of which can be measured, together with an appropriate way of comparing problems via their GP models. We describe our specific implementation of the model-based framework and evaluate the approach on a number of pre-designed test problem sets.

The paper is organized as follows: Sect. 2 summarizes existing frameworks for problem characterization and their limitations. Our proposed model-based framework for problem comparison is described in Sect. 3, highlighting its main elements. In Sect. 4 we describe the experiments, including the problem sets used based on controlled transformations. The experimental results are presented and discussed in Sect. 5 followed by the concluding Sect. 6.

2 Existing Frameworks for Problem Characterization and Algorithm Selection

An early framework for algorithm selection based on problem features was proposed by Rice [20]. This framework is based on extracting problem characteristics $c \in C$ for the given problem $f \in F$ and selecting an algorithm $\alpha \in A$ such that the output (e.g. performance) $y \in Y$ is maximized. The relationship between Rice's framework and more recent research in landscape analysis, meta-learning models and algorithm portfolios is discussed in [14].

Many different features have been proposed in the literature for problem characterization. But capturing and summarizing the structure of an arbitrary landscape is a difficult task [9]. Most features make very strong modelling assumptions (e.g. the R^2 coefficient for a linear or quadratic model of the landscape) or only use part of the information available in the sample (e.g. the sample skewness or kurtosis of the f values in the sample). Using a set of features in combination is a possibility [22], but the features are heterogeneous, making it difficult to select and utilize features in a principled way [16].

A regression model based on landscape features and algorithm (CMA-ES) hyperparameters is built in [13] to predict algorithm performance for a given problem. A framework to analyze the performance of algorithms using problem features is given in [15]. It uses a set of nine selected features and applies principal component analysis to reduce the feature space to two dimensions. An algorithm footprint is estimated on the feature space to relate which feature

Framework 1. Model-based continuous problem comparison

Given: set of problem instances, sample size N, regression model.
for all problems **do**
  Draw a sample of size N from the search space S.
  Evaluate f over the sample.
  Fit a regression model using the sample and f values.
  Evaluate the goodness of fit of the model.
end for
Calculate pairwise (dis)similarities between models.
Output: Problem similarity values.
Results Analysis: Dimensionality reduction or other techniques.

values correspond to particular algorithm performance. Another framework suggested in [11] uses algorithm rankings from the BBOB competition [5] to predict the best algorithm for a set of benchmark problems. The results are related to problem features to find some rules about the algorithm problem relationship. This approach requires a carefully chosen set of test functions and performance measures of the list of algorithms selected.

Most existing techniques that use problem features for algorithm selection or performance prediction are retrospective, using information not generally available for new problem instances, such as reported algorithm performance data or labelled categories of problems. Fundamentally, a way of comparing a set of problems in terms of their relative distances to each other might be simpler and yet still allow us to better understand the problem-algorithm mapping. In the next Section we describe a framework aimed at this, using GP regression models.

3 A Model-Based Framework for Problem Comparison

The framework proposed here essentially involves fitting a regression model to a sample of candidate solutions and fitness function values for a set of black-box optimization problem instances. A goodness of fit or error measure is then used to evaluate the regression model. Our framework compares problems by a comparison of the regression models built for each problem. This provides a set of pairwise distance or similarity values which characterize a problem set in terms of their relative distances or similarities. Algorithm selection could then follow based on the assumption that an algorithm that is effective for a given problem instance is likely to also be effective for nearby problem instances. Calculating features embeds a problem set in a somewhat arbitrary space, where selecting a suitable similarity measure may be difficult. Assuming the sample contains some information about the important landscape structure, a clear advantage of using a flexible regression model is that increasing the sample size will typically result in an improved model fit (i.e. a better representation of the problem landscape) whereas in the case of estimating problem features, increasing the sample size simply makes the feature estimate more robust.

3.1 Problem Comparison Using Gaussian Processes

In principle any regression model could be used in the above framework, however GPs are particularly well-suited to this task, as discussed below. Briefly, a GP is defined as a collection of random variables such that their joint distribution is a multivariate Gaussian, $\mathcal{N}(\mu, \Sigma)$ [19]. A GP is completely specified by its mean ($\mathbf{E}[f(\mathbf{x})]$) and covariance ($k(\mathbf{x}_i, \mathbf{x}_j)$) functions. This defines a prior distribution over the function space. Given a set of training data consisting of input vectors, $\mathbf{x}$ and target values $\mathbf{y}$, the posterior predictive distribution of the GP at a test point, $\mathbf{x}_*$, is Gaussian with mean and variance given by:

$$\bar{f}_* = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y} \tag{2}$$

$$\text{Var}(f_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_* \tag{3}$$

where K is the $(N \times N)$ covariance matrix between all pairs of points in the training set, $\mathbf{k}_*$ is a vector of covariance values between the test point and the training set, I is the identity matrix and σ_n^2 is an additive noise parameter (see below).

In general, calculating a distance or difference between regression models can be a complex task. However given two GPs, the distance calculation becomes a difference between two multivariate Gaussian distributions. The difference between two continuous probability density functions, $p(\mathbf{x})$ and $q(\mathbf{x})$, is commonly measured using the Kullback Leibler (KL) divergence:

$$d_{KL}(p||q) = \int_{-\infty}^{\infty} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} dx \tag{4}$$

If we have two multivariate Gaussian distributions, $\mathcal{N}_a(\mu_a, K_a)$ and $\mathcal{N}_b(\mu_b, K_b)$, then the KL divergence can be written in closed form:

$$d_{KL}(\mathcal{N}_a||\mathcal{N}_b) = \frac{1}{2} \log |K_b K_a^{-1}| + \frac{1}{2} K_b^{-1}((\mu_a - \mu_b)(\mu_a - \mu_b)^T + K_a - K_b) \tag{5}$$

We use the Jeffreys divergence [7], d_J which is a symmetric version of the KL divergence:

$$d_J(\mathcal{N}_a||\mathcal{N}_b) = \frac{1}{2}(d_{KL}(\mathcal{N}_a||\mathcal{N}_b) + d_{KL}(\mathcal{N}_b||\mathcal{N}_a)) \tag{6}$$

This gives us an efficient and direct way to calculate the difference between two problems via the GP models of their landscapes. Similar problems are expected to have relatively small divergence values and large divergence values will imply that the problems are quite different from each other.

3.2 Gaussian Process Implementation Details

Building a GP regression model requires the selection of mean and covariance functions. The mean function is assumed to be zero (the sample data can be

centred by subtracting the sample mean prior to fitting the model). There are many possible covariance functions that can be used to characterize the degree of similarity between data points in the input space [18]. The squared exponential covariance function is the most common choice [19]. The squared exponential covariance function has the form:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(\frac{1}{2l^2}||\mathbf{x}_i - \mathbf{x}_j||_2\right) + \sigma_n^2 \delta_{ij} \tag{7}$$

where δ_{ij} is the Kronecker delta function. The characteristic length scale l is indicative of the smoothness of the function. This hyperparameter captures the distance needed to move in any direction for the function values to become uncorrelated. σ_f^2 is the signal variance. In addition, the noise variance σ_n^2 is a hyperparameter of the GP which specifies the trade-off between the strength of the prior and fitting observed data [19]. In this paper we use the squared exponential covariance function as a spherical model across all dimensions of our problem sample data (i.e. a single l parameter is used). The optimization of these hyperparameters is important for fitting a GP. We use a standard approach to this: conjugate gradient is used to maximize the log-likelihood [19]. The hyperparameter optimization problem is not convex, so we use trial and error to set hyperparameters in the ranges indicated by the problem data. Many values worked well; for the results presented here the initialization used was $l = 0.5$, $\sigma_f^2 = 1000$ and $\sigma_n^2 = 0.1$.

In case of very smooth functions, the correlation between the observations is very high. This results in very similar rows/columns in the covariance matrix which can lead to making it poorly conditioned. This is an important issue in the implementation of GP's. The inversion of the covariance matrix is done using the Cholesky decomposition as it increases the tolerance towards the conditioning problem [17].

3.3 Related Work

Surrogate models have been widely used in optimization, particularly for problems where evaluating f is expensive. The model is used in place of the actual objective function. In using surrogate models it is important to balance the number of samples used (which should be minimum) with the improvement in approximation [3]. GPs as well as other models such as Random Forests and Support Vector Machines have been used [1]. Bayesian optimization algorithms use GPs to find a better solution using a minimum number of function evaluations [4,17]. A survey on the use of surrogate models in evolutionary computation is given in [6]. A framework based on using GP to find the promising individuals in the PSO population during the search is presented in [21].

As far as we are aware, there has only been one recent paper where surrogate regression models have been directly used to characterize problem instances [2]. The authors' main focus is on reducing the sample size required for feature-based algorithm selection. Therefore, a GP model is built from a small sample

as a surrogate, with further sampling carried out on the surrogate. In contrast, we calculate the difference between models fitted to these samples rather than computing features based on the surrogate model.

Finally, the Jeffrey's divergence has previously been used to compare problems in the context of length-scale feature analysis [12]. Length scale distributions were obtained using kernel density estimation and the divergence values calculated numerically over the sample. As shown above, the GP model-based framework requires no density estimation and the divergence values are calculated in closed-form.

4 Experimental Methodology

In this section we validate our framework on several test problems. We have generated 4 sets of 11 problem instances by gradually transforming a standard test problem into a different problem in a controlled way. The transformations determine a possible intuitive ordering of the problems, which we then try and recover in the black-box scenario, using the GP model-based framework. Existing problem sets such as the BBOB functions [5] do not have such an ordering, making it less straightforward to evaluate our results. The problem transformations are:

- Sphere to Rastrigin: the amplitude of the periodic term in the Rastrigin function is increased in 10 equal steps.
- Rastrigin to Flat: piece-wise linear combination, flat region grows from the center and expands equally in each intermediate problem.
- Sphere to Ellipse: increase in condition number across problem dimensions.
- Linear to Sphere: piece-wise linear combination, linear slope function is replaced by sphere function starting from the center.

Figure 1 illustrates the transformations in 1-D. We used 2-D and 5-D versions of the problems, with $S = [-5, 5]^n$. The sample size used was $N = 1000$ for the 2-D problems and $N = 2500$ for the 5-D problems. To visualize the results, dimensionality reduction techniques based on the similarity matrices can be utilized. We applied the t-Stochastic Neighbor Embedding (t-SNE) algorithm [8] as well as heatmaps and dendrograms to visualize the problem comparisons. t-SNE is a state of the art dimensionality reduction technique in machine learning. The algorithm is based on similar principles to our framework in that it calculates the KL divergence between a distribution of distances in a high dimensional data space and a lower dimensional mapping.

5 Results and Discussion

To measure the goodness of fit of the GP model, we have used the normalized mean square error (NMSE) for the test set. The NMSE calculates the deviation between the GP model predicted ($f_*(\mathbf{x}_i)$) and actual fitness values at sample points. It is defined as:

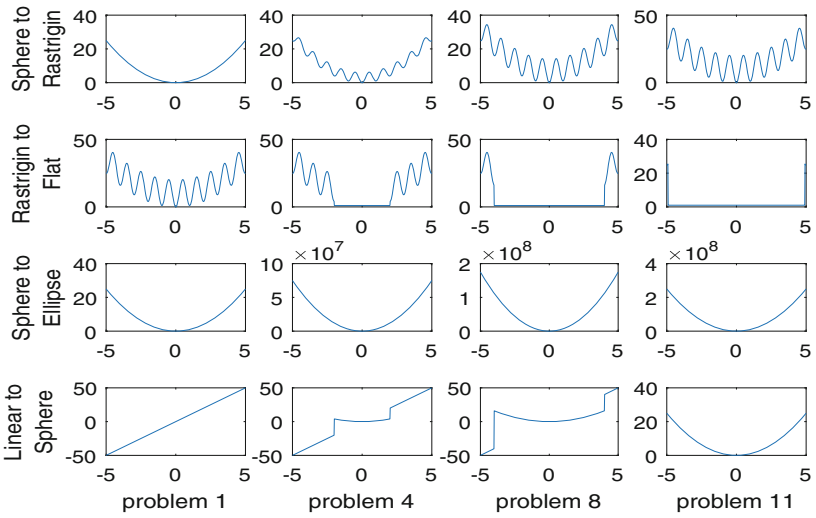

Fig. 1. Problem transformations used to generate the test problem sets. Shown are 1-D versions of problems 1,4,8 and 11 in each set.

$$NMSE = \frac{1}{N} \sum_{i} \frac{(f_*(\mathbf{x}_i) - f(\mathbf{x}_i))^2}{\overline{f_*()} \cdot \overline{f()}} \tag{8}$$

where $\overline{f()}$ is the sample mean of the function values. The NMSE values for the models of all the problems in each transformation in 2D are shown in Table 1. The NMSE values for 5D problem models are shown in Table 2.

Table 1. The estimated NMSE values in 2D.

Prob ID	Sphere2Ellipse	Rastrigin2Flat	Sphere2Rastrigin	Linear2Sphere
1	8.59E-06	0.001599	9.44E-06	3.66E-06
2	1.20E-08	0.0067396	0.00019719	3.72E-06
3	3.91E-09	0.014997	0.0002686	0.001875
4	9.51E-10	0.01169	0.00026819	0.0034078
5	5.38E-10	0.031776	0.00052105	0.0076807
6	7.59E-10	0.077764	0.00028147	0.014644
7	3.50E-10	0.085257	0.00071756	0.07977
8	3.17E-10	0.31847	0.00066255	0.092218
9	2.21E-10	4.1164	0.00019552	0.015076
10	2.36E-10	2.7546	0.00029032	0.11195
11	2.89E-10	0.002448	0.00036012	8.79E-06

Table 2. The estimated NMSE values in 5D.

Prob ID	Sphere2Ellipse	Rastrigin2Flat	Sphere2Rastrigin	Linear2Sphere
1	1.20E-06	0.030001	1.63E-06	2.97E-06
2	1.15E-08	0.030058	0.0013048	2.45E-06
3	4.76E-09	0.032689	0.004254	3.53E-06
4	3.05E-09	0.038068	0.0080946	0.0014831
5	9.84E-10	0.057157	0.011884	0.0264
6	8.69E-10	0.11271	0.01589	0.1312
7	4.85E-10	0.40765	0.019802	0.23115
8	2.03E-10	2.5472	0.023586	0.22068
9	1.56E-10	7.4724	0.027164	0.014305
10	3.71E-11	0.79306	0.030465	1.47E-06
11	1.70E-10	0.0024149	0.033622	1.54E-06

The t-SNE plots for the Sphere to Ellipse transformations in 2D and 5D are shown in Fig. 2. The results show that the order of the problems in the transformation is very strongly recovered from our framework (i.e. problem i tends to be closest to $i-1$ and/or $i+1$). The model error values for all the problems in this set are very small, indicating that the GP is an accurate model of these landscapes. The dimensionality reduction attempts to preserve the pairwise distances between the problem instances, so the orientation of the points in the plot is not meaningful.

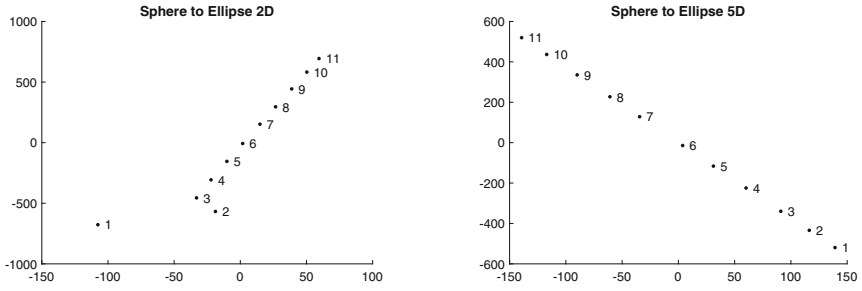

Fig. 2. t-SNE visualisations of the Sphere to Ellipse problem set. Left: 2D, Right: 5D

Figure 3 shows the t-SNE plot for the Rastrigin to Flat problem set in 2D and 5D. The visualisation shows that the problem order from the transformation is strongly recovered. This problem transformation is rather complex as it combines multiple local minima and a perfectly flat surface. The error values for the models (Tables 1 and 2) show relatively high values for problems 8,9 and 10, suggesting that the model lacks some accuracy at modelling the function. This

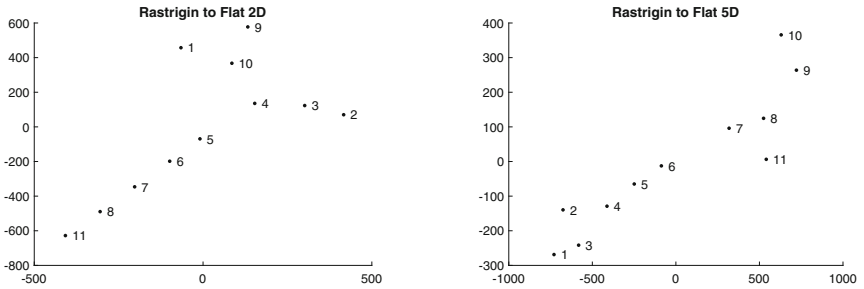

Fig. 3. t-SNE visualisations of the Rastrigin to Flat problem set. Left: 2D, Right: 5D

may partly explain the distribution of problems in Fig. 3 (e.g. problems 9 and 10 in the 2D set are well separated from 8 and 11). No model can be expected to provide a highly accurate model of arbitrary fitness landscapes based on a modest sample of data. A strength of our approach is that rather than simply accepting calculated feature values, the error of the model gives us a measure of how reliable our results are. For the Sphere to Rastrigin transformations, Fig. 4(a) shows excellent recovery of the problem ordering in 2D. The Sphere function (problem 1) appears somewhat separate, perhaps because it is the only unimodal problem in the set. In Fig. 4(b) the trend is not as visually obvious across the entire problem set, however most problem instances have, as their nearest neighbours, the neighbouring problems in the transformation (e.g. 8 is closest to 9 and 10, 7 is closest to 6 and 5). The model error values for this problem transformation, (Tables 1 and 2), are all relatively low indicating a good fit to the data. Error values increase a little from 2D to 5D, with these sample sizes. Note that a model can still be capturing some important landscape properties and have a large error value.

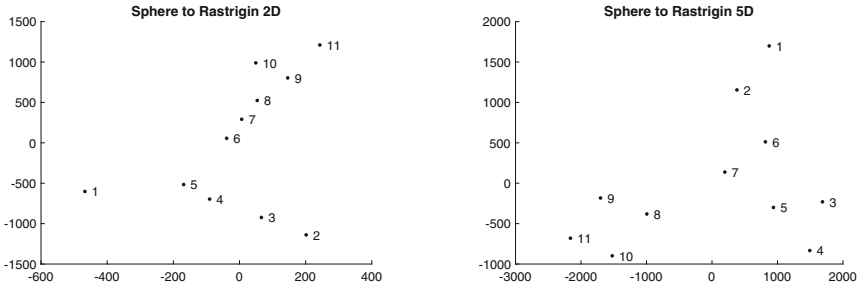

Fig. 4. t-SNE visualisations of the Sphere to Rastrigin set. Left: 2D, Right: 5D.

The Linear to Sphere transformation is done in a piece-wise way which may contribute to high NMSE values in the middle functions of the transformation.

Function 1 is the Linear function and Function 11 is the Sphere function and both have a very good model fit. Both of these functions are smoothly structured functions and we can see in the Fig. 5 that both are rather close to each other. The remaining problems also form a cluster which show their similarity.

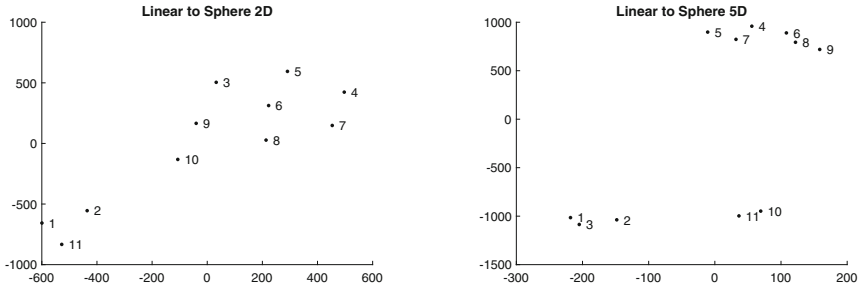

Fig. 5. t-SNE visualisations of the Linear to Sphere problem set. Left: 2D, Right: 5D

A more direct way of examining the results is to look at a pairwise distance matrix. Figure 6 shows two examples for the 5D Sphere to Ellipse and 5D Linear to Sphere problem sets visualized as heat maps. The pattern across the Sphere to Ellipse transformation reflects almost perfectly the definition of the transformation: nearby problems are close to the diagonal and have a low distance value, which smoothly increases as we move away from the diagonal. For the Linear to Sphere transformation we can see that the model distances between problems 1–3 and 5–7 are greater than expected, as are the distances between 5–8 and 10–11. This agrees with the t-SNE plot for this problem set (Fig. 5, right). Finally, dendrograms offer another popular way of displaying distance data. Figure 7 shows examples for the 2D Linear to Sphere and 5D Sphere to Rastrigin problem sets. For some problems (Figs. 5, left and 4, right), the relative magnitude of the distances between problems is more accurately represented in dendrograms as compared to the t-SNE plots for these problem sets.

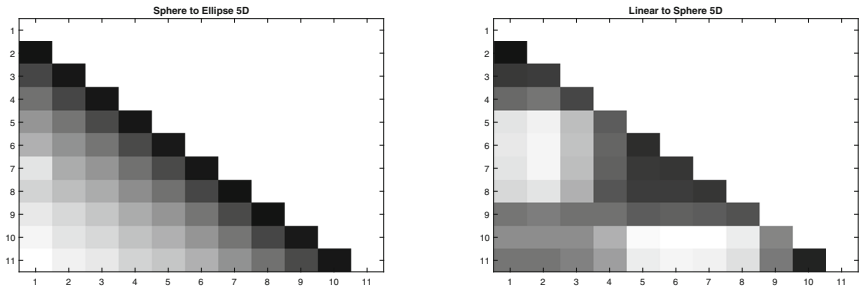

Fig. 6. Heat maps for two of the problem sets. Left: 5D Sphere to Ellipse, Right: 5D Linear to Sphere. Greyscale shows the log of the distance values for better contrast.

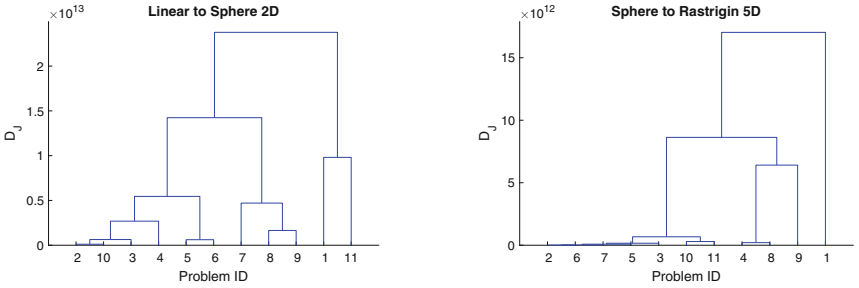

Fig. 7. Dendrograms for two of the problem sets. The relative distances between problems is given by the y-axis. (Left) 2D Linear to Sphere transformation, (Right) 5D Sphere to Rastrigin transformation.

6 Conclusion

Exploratory landscape analysis and algorithm selection frameworks have motivated the research in understanding problems. Here, we have proposed a model based framework for understanding problems using Gaussian Processes. Being a regression model, we have a measure of goodness of fit which provides a source of verification of model. To get the distance between models of problems, GP provide a closed form expression for measuring KL divergence which avoids any numerical approximations. We tested the methodology on a set of problem transformations with pre-defined similarity ranking. The framework is tested on its ability to identify the distance between problems in each transformation. GPs are known to provide a surrogate model of the function using a small set of samples. In future we will extend our methodology on problems with smaller sample sizes. The experiments in this paper are limited to 2D and 5D problems, but higher dimensional problems also need to be explored. The experimental results presented here indicate that measures based on these models can detect small differences between problems and recover much of an specified problem ordering.

References

1. Bajer, L., Pitra, Z., Holeňa, M.: Benchmarking Gaussian processes and random forests surrogate models on the BBOB noiseless testbed. In: Proceedings of the Companion Publication of the 2015 Conference on Genetic and Evolutionary Computation, pp. 1143–1150. ACM (2015)
2. Belkhir, N., Dréo, J., Savéant, P., Schoenauer, M.: Surrogate assisted feature computation for continuous problems. In: Festa, P., Sellmann, M., Vanschoren, J. (eds.) LION 2016. LNCS, vol. 10079, pp. 17–31. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50349-3_2
3. Forrester, A., Keane, A., et al.: Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, Hoboken (2008)
4. Frean, M., Boyle, P.: Using Gaussian processes to optimize expensive functions. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 258–267. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89378-3_25

5. Hansen, N., Finck, S., Ros, R., Auger, A.: Real-parameter black-box optimization benchmarking: noiseless functions definitions. Technical report (2009)
6. Jin, Y.: Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol. Comput. **1**(2), 61–70 (2011)
7. Legrand, L., Grivel, E., Giremus, A.: Jeffrey's divergence between autoregressive moving-average processes. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 1085–1089. IEEE (2017)
8. Maaten, L.V.D., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
9. Malan, K.M., Engelbrecht, A.P.: Fitness landscape analysis for metaheuristic performance prediction. In: Richter, H., Engelbrecht, A. (eds.) Recent Advances in the Theory and Application of Fitness Landscapes. ECC, vol. 6, pp. 103–132. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-41888-4_4
10. Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., Rudolph, G.: Exploratory landscape analysis. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, pp. 829–836. ACM (2011)
11. Mersmann, O., Preuss, M., Trautmann, H., Bischl, B., Weihs, C.: Analyzing the bbob results by means of benchmarking concepts. Evol. Comput. **23**(1), 161–185 (2015)
12. Morgan, R., Gallagher, M.: Analysing and characterising optimization problems using length scale. Soft Comput. **21**(7), 1735–1752 (2017)
13. Muñoz, M.A., Kirley, M., Halgamuge, S.K.: A meta-learning prediction model of algorithm performance for continuous optimization problems. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012. LNCS, vol. 7491, pp. 226–235. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32937-1_23
14. Munoz, M.A., Kirley, M., Halgamuge, S.K.: The algorithm selection problem on the continuous optimization domain. In: Moewes, C., Nürnberger, A. (eds.) Computational Intelligence in Intelligent Data Analysis, pp. 75–89. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-32378-2_6
15. Muñoz, M.A., Smith-Miles, K.A.: Performance analysis of continuous black-box optimization algorithms via footprints in instance space. Evol. Comput. **25**(4), 529–554 (2017)
16. Muñoz, M.A., Sun, Y., Kirley, M., Halgamuge, S.K.: Algorithm selection for black-box continuous optimization problems: a survey on methods and challenges. Inf. Sci. **317**, 224–245 (2015)
17. Osborne, M.A., Garnett, R., Roberts, S.J.: Gaussian processes for global optimization. In: 3rd International Conference on Learning and Intelligent Optimization (LION3), pp. 1–15. Citeseer (2009)
18. Rasmussen, C.E., Nickisch, H.: Gaussian processes for machine learning (GPML) toolbox. J. Mach. Learn. Res. **11**(Nov), 3011–3015 (2010)
19. Rasmussen, C.E., Williams, C.K.: Gaussian Process for Machine Learning. MIT press, Cambridge (2006)
20. Rice, J.R.: The algorithm selection problem. Adv. Comput. **15**, 65–118 (1976)
21. Su, G.: Accelerating particle swarm optimization algorithms using Gaussian process machine learning. In: 2009 International Conference on Computational Intelligence and Natural Computing, CINC 2009, vol. 2, pp. 174–177. IEEE (2009)
22. Sun, Y., Halgamuge, S.K., Kirley, M., Munoz, M.A.: On the selection of fitness landscape analysis metrics for continuous optimization problems. In: 2014 7th International Conference on Information and Automation for Sustainability (ICIAfS), pp. 1–6. IEEE (2014)