How Predictable are Fitness Landscapes with Machine Learning? A Traveling Salesman Ruggedness Study

Mohammed El Amrani[1], Khaoula Bouanane[2], Youssef Benadada[3]
ANISSE Team-Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco[1]
Smart Systems Laboratory, ENSIAS, Mohammed V University in Rabat, Rabat, Morocco[2,3]

***Abstract*—The notion of fitness landscape (FL) has shown promise in terms of optimization. In this paper we propose a machine learning (ML) prediction approach to quantify FL ruggedness by computing the entropy. The approach aims to build a model that could reveal information about the ruggedness of unseen instances. Its contribution is attractive in many cases like black-box optimization and in case we can rely on the information of small instances to discover the features of larger and time-consuming ones. The experiment consists in evaluating multiple ML models for the prediction of the ruggedness of the traveling salesman problem (TSP). The results show that ML can provide, for instances of a similar problem, acceptable predictions and that it can help to estimate ruggedness of large instances in that case. However, the inclusion of several features is necessary to have a more predictable landscape, especially when dealing with different TSP instances.**

***Keywords*—Fitness landscape analysis; optimization algorithms; machine learning; landscape ruggedness; traveling salesman problem**

I. Introduction

Over the last few decades, work on optimization algorithms has mainly focused on the algorithmic side, while the analysis of the problem itself has received relatively little attention. That is, most of the research published on this topic does not provide a sufficient analysis of the problem, why the algorithm works well and under which conditions [34]. Therefore, characterizing a problem should lead to a deeper understanding of it and better choices of algorithms and, hence, have an increased chance of producing better solutions. For this purpose, the concept of fitness landscape (FL) analysis was proposed with the aim of designing a generic approach that characterizes optimization problems. The concept was first introduced to illustrate the dynamics of biological evolutionary optimization [28], but it has also proved useful in understanding the behavior of optimization algorithms in both binary and continuous optimization problems. Thus, FL analysis is relevant both to predict the performance of algorithms and to improve their design. The interested reader is referred to [19] for more details on the transition from modeling real processes to modeling optimization problems. The issue of predictability of the FL was studied in biology (e.g. [4]) and in this paper, we aim to show a case in which it can be helpful for combinatorial optimization.

The integration of machine learning (ML) in operations research is increasingly crucial in light of recent advances [12], especially for the study of fitness landscapes in optimization problems such as the traveling salesman problem (TSP). ML can aid in characterizing ruggedness, thereby enhancing decision-making with predictive insights. This paper investigates the ruggedness of TSP landscapes using entropy as a key measure and introduces a novel ML-based approach for predictive analysis. The study addresses challenges in landscape characterization, extending traditional methods to unseen problem instances.

In general, the aim of FL analysis is to improve knowledge about the properties of a problem. Malan et al. [4] highlighted several characteristics of the FL as well as the measurements used for them. In particular, ruggedness is a property that often depends on the number and distribution of local optima and is related to the level of variation in fitness values in a FL. A number of measurements have been proposed to measure the landscape ruggedness. These measurements are the subject of the paper, and our aim is to deepen our understanding of this property, leveraging the great advances in data-driven approaches that have been carried out in recent years. More specifically, our approach consists of proposing a machine learning (ML)-based approach that extends existing methods for quantifying landscape ruggedness for unseen instances, in which its calculation can be time consuming.

This paper has three main objectives: (1) to evaluate the ability of ML to predict the robustness of TSP fitness landscapes, (2) to explore the role of entropy as a key metric in landscape analysis, and (3) to assess the generalization ability of ML to unseen problem instances. In pursuing these objectives, the paper makes the following contributions:

- It extends previous work by applying ML techniques to analyze TSP robustness, leveraging entropy to improve fitness landscape characterization.
- To the best of our knowledge, this is the first study to use ML to understand a TSP-specific fitness landscape feature, providing new insights into its structure and complexity.

The rest of paper is organized as follows. In Section II, we present the concept of ruggedness of the landscape and the measures proposed to quantify it. Section III is devoted to the description and discussion of the integration of ML in this problem. Both sections are interleaved with a brief literature review of the topic. Section IV presents the experiments.

Finally, the conclusion and the perspectives are depicted in Section V.

II. Background and Literature Review

The use of machine learning to understand the robustness of FL in combinatorial optimization problems, such as the TSP, is gaining increasing attention. A similar goal is explored in [30], where the authors examine the generalization ability of a machine learning model for problem reduction on the classical TSP. This paper shares a similar goal, aiming to explore the ability of ML to model landscape robustness, but distinguishes itself by focusing on entropy-based measures and their predictive potential.

A. Ruggedness of the Fitness Landscape

An important issue that may arise for FL analysis concerns to which extent we are able to generate generic FL measures. Although there are a number of independent measures such as the number of global optima, it is widely accepted that in many problems it is not possible to fully characterize the landscape using independent features [19]. Indeed, what is difficult to solve for a randomized hill climbing is not necessarily difficult for a genetic algorithm (GA) or when using a GA with different mutation operators. We can also note that many of the adopted measures depend on the definition of a neighborhood relation (e.g. the size of a basin of attraction). Based on this remark, the most well-known definition of a FL was proposed in [18] and consists of the triplet (X, N, ϕ), where:

- X is a set of candidate solutions (search space)
- N is a neighborhood relation
- ϕ: $X \to \mathbb{R}$ is the fitness function

We can see from this definition that the FL not only depends on the problem but is also strongly related to the choice of an algorithm's operator. Therefore, as pointed out in [10], the concept of landscape could only be fully characterized in the context of an associated neighborhood structure and a specific operator. In what follows, we are interested in the ruggedness feature which also depends on the operator.

The issue of fitness ruggedness has been addressed in the literature from a number of perspectives. For instance, Vassilev et al. [10] defined three properties to characterize the FL, which are neutrality, smoothness and ruggedness. Most often the difficult and problematic case concerns the rugged landscape. This case is considered the most challenging and several works have been adopted to treat it (although neutral and smooth FLs have also been examined in the literature).

Concerning measurements, the auto-correlation function (ACF) proposed by Weinberger [11] is the most classic one for measuring it. The idea shown in that paper consists of performing random walks using a specific operator to study the correlation structure of a landscape. More precisely, the author, by considering the case of mutation as an operator, carried out random walks starting from a point chosen at random; then at each step, a bit of the vector chosen at random is flipped. We note that in this case, the ACF could be considered as a time series [35]. However, the ACF measure has been criticized in some works, which have pointed out its weakness in the characterization of the FL (e.g. [7]). Moreover, [16] pointed out that the importance of autocorrelation is often overplayed in fitness landscapes studies. Thus, the measure proposed in [32] to study ruggedness, which is described below, has become the most common over the last decade. We refer to [32] for a detailed description of the approach based on ruggedness.

It should also be noted that ruggedness has been primarily studied for continuous optimization problems [33], but was extended to combinatorial optimization [17]. Another measurement of ruggedness has been proposed, which is information content [23], and is beyond the scope of this paper.

The ruggedness has been studied for some optimization problems. The study in [16] investigated the similarities and difference between four combinatorial optimization problems, including the traveling salesman problem (TSP) and the quadratic assignment problem (QAP). In particular, their study showed that the four problems have similar ruggedness. Furthermore, Tayarani and Bennett [31] studied the impact of the ruggedness among other measures for the graph-coloring problem. In addition, Kallel et al. [10] reviewed some mathematical properties of the ruggedness.

Although ML has not been used yet to predict the ruggedness; an approach to predict it was proposed using time series analysis. In fact, Hordijk [35] proposed an extension of [11] using the Box–Jenkins method [1]. The author's idea consists of exploring the landscape structure by studying the corresponding autoregressive moving average (ARMA) model [1] which, according to the author, characterizes the landscape ruggedness much more precisely; its contribution consists in providing a stochastic model which could be used to make predictions for fitness values of distant points in the landscape.

We can notice, in this section, that entropy is an important measure used to characterize landscape ruggedness in certain works. In this paper, we aim to bridge the gap between them and advancements in data-driven approaches. Therefore, in Section III, we highlight works focused on understanding the TSP landscape and main approaches using ML for FL analysis, then introduce our approach and the adopted ML algorithms.

B. Understanding of the Traveling Salesman Problem Landscape

There are multiple papers which studied the FL of the TSP. For example, Boese et al. [2] asserted that the search space of TSP instances (under 2-opt moves) has a big-valley structure, in which local optima are clustered around one central global optimum. However, this statement has been questioned in multiple papers (e.g. [5]) and its generalization is also an issue of discussion [24]. Indeed, as noted in study [25], the TSP structure is not yet fully understood. On the other hand, ML was used for the TSP but not to analyze its landscape. We refer to study [21] for more information on the topic. In particular, in study [36], the authors investigated the generalization error of a ML model when the training and test instances have different instance characteristics, sizes or are from different TSP variants. The authors have a goal similar to our paper, namely to test the generalization capacity of ML algorithms to large instances, but using a different approach. Another analysis technique, notably the principal component analysis, was used in study [35] to analyse several features of the TSP FL. The authors provided several conclusions which can mainly be summarized as follows: the difficulty of the instance (e.g. the number of local optima, the probability of reaching a global optimum) increases with the size of the problem and the level of the increase depends on the type of problem.

C. Machine Learning for Fitness Landscape Analysis

ML has been adopted in the context of FL in most cases with the aim of selecting the best algorithm based on the prediction of its performance, by adopting the information included in the computed features of the landscape. By analyzing the literature, we note that one of the first papers which paved the way for the emergence of such research is [15]. It introduced the concept of empirical hardness of optimization, showing how to build empirical hardness models which, given a new problem instance, predict an algorithm's runtime. Moreover, awareness of the importance of such approaches were reinforced with the appearance of the concept of exploratory landscape analysis (ELA) [13], shifting attention to this topic over the past decade.

However, we can notice that most of these researches are empirical-based and try to add features or to compute them in a less expensive way (e.g. [9]), without clearly contributing to the improvement of the problem understanding. The aim of these works is to build automatic tools for the selection or design of algorithms. Therefore, a number of tools have been proposed to extract FL features (e.g. Flacco package [14]).

In fact, we can see that such frameworks, even if useful in practice, cannot help to improve our knowledge on the problem which is of the utmost importance. That is, in most of these ML approaches, the authors try to use or define a large number of features, which could be in the hundreds, that may affect the performance of the algorithms. For instance, Mirshekarian et al. [22] proposed 380 features for the job-shop scheduling problem. But, many of the defined features might not be appropriate [37]. Hence, a typical phase is feature selection, which aims to select the most relevant ones. The ultimate goal of these approaches is to select the most suitable algorithms without providing an explanation for that selection. We note that we are aware that with the appearance of deep learning, the features could be computed automatically as investigated, for example, in [8]. However, deep learning is also unable to provide an explanation of the different selections.

The literature review elucidates the challenge of quantifying the ruggedness of fitness landscapes, emphasizing its dependence on problem characteristics and algorithmic operators. It surveys various methods for ruggedness characterization, with a focus on entropic measures, highlighting their significance in optimization studies. Moreover, it positions the current study as pioneering in utilizing machine learning to predict ruggedness, aiming to enhance understanding and inform algorithmic selection strategies.

Therefore, in this paper, our adoption of ML is different. In fact, we are interested in predicting the values of a specific and crucial feature (ruggedness). Although it is necessary to inclusion of several features for a better FL analysis, this work can be considered as a first and crucial work in this respect. To the best of our knowledge, despite its importance, this is the first investigation of the use of ML for predicting the ruggedness of FL. Ruggedness has been considered instead as a feature for most data-driven approaches for algorithm selection based on ELA. In the following, we describe our proposed approach.

III. The Proposed Approach

Our approach consists of two steps. The first concerns TSP optimization. In this step, the data necessary for the experiments is collected by running an optimization algorithm which consists of successive random walks. The number of iterations is fixed at 100. We choose (2-opt) as the neighborhood to implement the random walk for the TSP. Using this, we can calculate the entropy values of different problem instances as in Eq. (4). For each instance, we performed 30 executions of the random walk and computed the entropy for each instance and run. We are then able to obtain a sufficient sample size for the three experiments, which is 360, 300 and 240, respectively. As described below, in the first experiment, we generate random instances with different sizes. In the second, we analyze different TSPLIB instances while the third one is about a prediction for a specific TSPLIB instance family.

The second step consists of ML prediction. The target variable to be predicted (y) is the entropy. Two features (X) are used, which are the number of cities and the execution number, in addition to the instance family for TSPLIB instances. Based on the accuracy of predictions on unseen instances, we evaluate our approach. Below, we highlight the adopted ML algorithms and define the different steps of our approach.

Algorithm 1: Data Generation for ML Algorithms

Data: TSP instance
Random generation of distances for random instances
for $k = 1$ ***to*** 30 **do**
  for $l = 1$ ***to*** 100 **do**
    Perform a random walk
  Compute the entropy corresponding to execution k
Result: Entropy values in addition to TSP instances-related information

In Algorithm 1, we present the adopted ML algorithm and outline the key steps of our approach. Additionally, the flowchart (Fig. 1) provides a visual representation of the methodology, illustrating the process from data generation to ruggedness prediction for TSP instances using ML models.

In this paper, we have adopted a number of ML methods that yield satisfactory results for regression (continuous) problems such as gradient boosting (GR), random forest (RF) and support vector machines (SVM). These approaches, which are among the most adopted ones for regression problems, are the methods used to predict ruggedness. For more information about them, we refer to [20], [3] and [29], respectively.

IV. Experiments

In this section, a series of numerical experiments are carried out to evaluate the ML prediction of landscape ruggedness. The objective is twofold: first, we illustrate how ML can be adopted for this problem. Second, we seek to see what

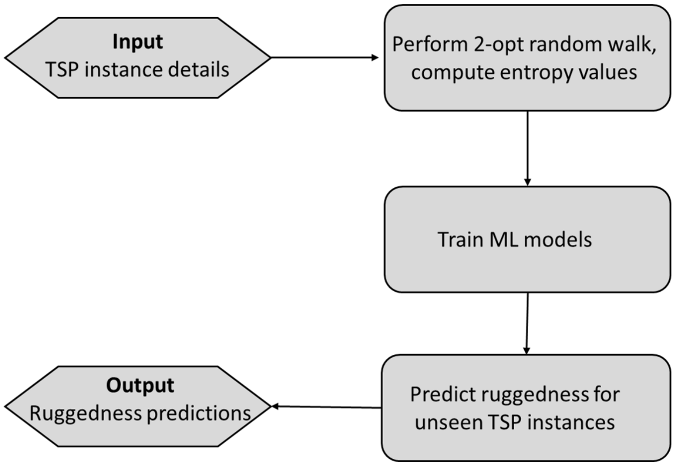

Fig. 1. Flowchart illustrating the proposed system for ruggedness prediction of TSP fitness landscapes.

this prediction can reveal about the structure of landscapes of unseen instances. More specifically, as mentioned above, our experiment first consists of running an algorithm, which is made up of consecutive randomized moves, on instances of classical combinatorial optimization problems while computing the ruggedness entropy value and second, of examining the ML predictive capacity of the entropy on these and other unseen instances by comparing the predicted values with the actual ones.

A. Experimentation Setup

In this paper, we consider the TSP, which is one of the most studied problems in combinatorial optimization. The Mlrose package [6] is adapted to implement the random walk with 2-opt neighborhood. First, we start the experiments with randomly generated TSP instances (the distances are generated randomly). Then, we experiment the approach on TSPLIB instances.

As mentioned earlier, the three ML algorithms chosen in this study are RF, GB and SVM. To implement them, we utilize the Scikit-learn library [27]. We optimize the parameters for RF and GB. Specifically, the number of trees in RF is set to 200, and the loss function to be optimized for GB is set to 'least absolute deviation' for better regression performance. For the other RF and GB parameters, we adopt the default parameters. Regarding SVM, the parameter γ (kernel coefficient) is automatically trained while the values of C (regularization factor) and ϵ are set to 1 and 0.01, respectively. We have chosen these values to enable a balance between overfitting and underfitting.

All experiments are conducted on a computer equipped with an Intel i7-9750H and 16GB of RAM. The measures adopted in this paper are: R^2 (coefficient of determination), mean absolute error (MAE) and mean square error (MSE).

To evaluate ML algorithms, there are two typical methods, notably training-test split or division and cross validation. In this paper, we have used the two methods depending on our objective. First, for random experiments and TSPLIB instances of a similar problem, we adopted the training-test split. More precisely, the first 75% of the data is used for training and 25% is used as test data. Our goal is to see if we can accurately predict the ruggedness of large TSP instances without needing to compute them and simply by examining small instances. Second, for different TSPLIB instances, we adopted 5-fold cross validation [36] to have a robust assessment of the ML predictions. In this case, the testing is performed across instances.

We have uploaded the used code for more details on our approach. The corresponding information is available in the Github repository (blinded for refereeing).

B. Random Instances

As a first experiment, we consider randomly generated TSP instances.

First, in our experiment, we choose nine instances for training and three for testing, depending on the number of cities. For training, we used the values (10, 20, 50, 100, 200, 500, 1000, 2000, 5000) and for testing, we adopted the values (6500, 8000 and 10000). As we have performed 30 runs for each instance, the training set size is 270 and the test set size is 90.

Second, for each of the three ML algorithms (RF and GB and SVM), we display in Table I a comparison of the prediction capabilities of the three algorithms in both training and test sets. That is, we show in Table I the R^2, MAE and MSE values obtained by comparing RF, GB and SVM predictions in training and test sets. The results of the test sets are those which are really necessary for evaluation but those of the training are given as additional information on the structure of the landscape.

TABLE I. Comparison of Entropy Predictions for Random TSP Instances

		R2	MAE	MSE
RF	Test acc.	0.7108	0.0202	0.0006
	Training acc.	0.9347	0.0180	0.0005
GB	Test acc.	0.6907	0.0214	0.0007
	Training acc.	0.7337	0.0190	0.0005
SVM	Test acc.	0.2137	0.0380	0.0020
	Training acc.	0.0498	0.0322	0.0016
NN	Test acc.	0.1641	0.6570	0.1300
	Training acc.	0.9630	0.2500	0.1023

Both RF and GB gave acceptable results on the test set. SVM, as trained, seems not to be suitable for this case. The best results are given by RF. That is, RF provided the best results on the test sets, which are the needed for unseen predictions. We can conclude that RF is the most suitable for this case study, with the proposed parameters.

Third, to give a better overview of the RF predictions, we depict in Fig. 2 and Fig. 3 the prediction given by RF in both test and training sets along with the real values.

We can notice from the Fig. 2 and Fig. 3 that the entropies are overall slightly positively correlated with the number of cities. In other words, the entropy in general increases slightly with the number of cities. This can be seen as the values in the

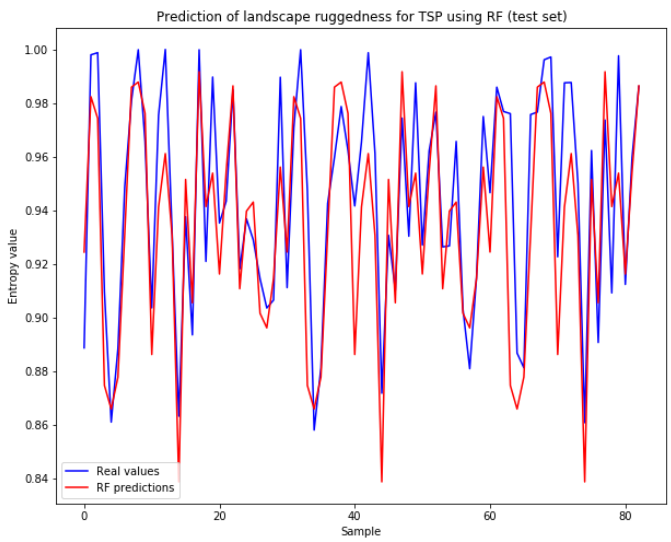

Fig. 2. RF prediction on the test set.

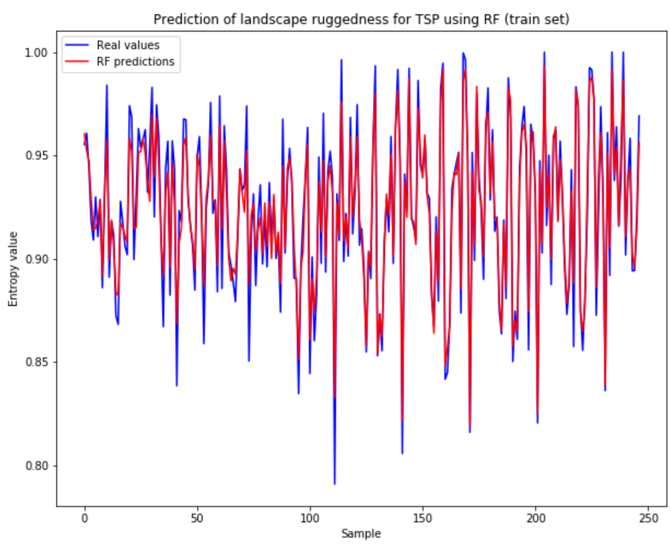

Fig. 3. RF prediction on the training set.

test sets are slightly, which corresponds to higher cities, are slightly higher than of the training set and there is a very weak increasing trend of the value in function of the sample (and then of the cities). Although ML prediction is not extremely accurate, it can detect patterns in the data and provide a fairly good ruggedness prediction.

C. Instances of Different TSPLIB Problems

After looking at the randomly generated TSP instances (Table II), we aim to study several TSPLIB instances.[1] Below we show the names of the instances with the corresponding number of cities.

Our goal in this part is to see how well we can predict ruggedness for problem instances, using the information from

TABLE II. TSPLIB Instances

Instance	Number of cities
Bays	29
Berlin	52
Brazil	58
Eil	51, 76, 101
Ch	130, 150
TSP	225
Fl	417

other instances, regardless of the instance family. To answer this issue, cross validation is more appropriate than training-test split. In this experiment, we conducted the 5-fold cross validation. In total, we get 300 sample and we use them for this purpose. In Table III, we display the results of the mean of the R^2, negative MAE^2 and MSE factors, which are used in the 5-fold cross validation.

TABLE III. Comparison of Entropy Predictions for Instances of Different TSPLIB Problem Instances

	R^2	Negative MAE	Negative MSE
RF	-0.8797	-0.0385	-0.1525
GB	-0.7478	-0.0352	-0.1422
SVM	-0.7514	-0.0550	-0.2057
NN	0.2660	-15877.1021	-102.0534

It is clear from Table III (e.g. the R^2 values) that the results are not good and the algorithms are not able to provide acceptable predictions. The reason for these unsatisfactory results compared to the previous case is due to the fact that ruggedness appears unpredictable if combined with other factors. It is necessary to combine several features to expect to have an accurate prediction. We note that the results are also not satisfactory when adopting the training-test split in the same way as in the first study.

D. Instances of a Similar TSPLIB Problem

In this section, we focus specifically on instances of a particular TSPLIB instance family and examine the evolution of ruggedness as a function of only the number of cities and, importantly, our ability to predict large unseen instances. More precisely, we consider the instance studied in [26]. In this case study, we consider the instances with cities of 76, 107, 124, 136, 144 and 152 as a training test (75%). The instances with 226 and 264 (25%) are the test set. All instances are executed 30 times. As mentioned before, the reason is that we aim to see if we can predict the ruggedness of instances with higher cities by simply getting information from lower cities, and cross validation is not needed in this case.

We can notice from Table IV, that GB gave the best results in the test set. (We note that the RF predictions are better in the training set but the GB results are more promising in our context.) In Fig. 4 and Fig. 5, we provide the prediction given by GB in the test and training sets with the actual values.

We can notice from Fig. 4 and Fig. 5 that entropy does not globally have a very significant variation in function of the

[1] The instances can be found in http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html

[2] For cross validation, scikit-learn uses negative MAE and MSE. More information about them can be found in https://community.dataquest.io/t/why-is-scoring-equal-to-neg-mean-squared-error/547283

TABLE IV. Comparison of Entropy Predictions for Instances of a Similar TSPLIB Problem

		R2	MAE	MSE
RF	Test acc.	0.3088	0.0220	0.0015
	Training acc.	0.7829	0.0100	0.0002
GB	Test acc.	0.2102	0.0250	0.0020
	Training acc.	0.4651	0.0150	0.0006
SVM	Test acc.	-0.0078	0.0174	0.0009
	Training acc.	-0.0092	0.0203	0.0011
NN	Test acc.	0.1641	0.1150	0.1314
	Training acc.	0.9630	0.9066	0.2023

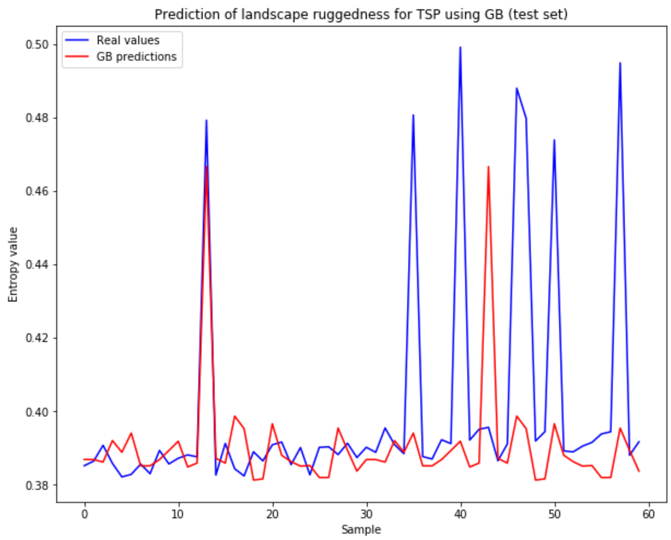

Fig. 4. GB prediction on the test set.

number of cities. The GB predictions are in general consistent with the actual values, and it can then be considered that ML can be useful in this situation.

The results for the three cases can be summarized as follows:

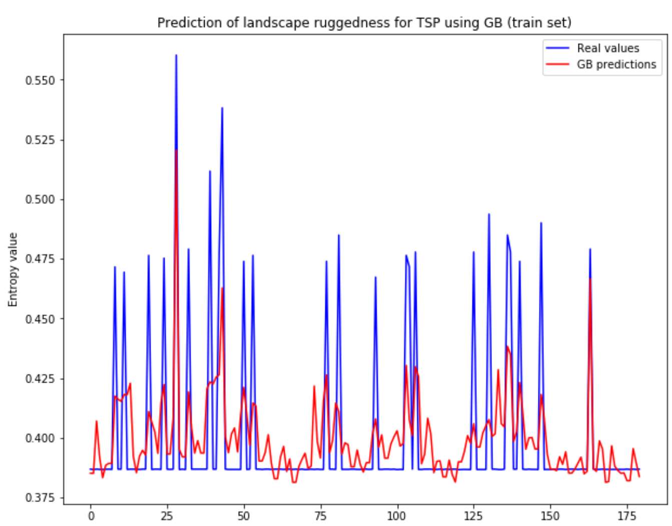

Fig. 5. GB prediction on the training set.

- For random TSP instances, the computed ruggedness factor seems to increase globally. ML can detect this pattern and then provide satisfactory predictions.
- ML algorithms failed to detect patterns in different TSPLIB instances. The reason seems that other factors than size must be included to be able to use ML in this case.
- For a specific TSPLIB instance, there is no significant increase in the factor. ML can also be useful in this case by giving a satisfactory prediction for the same unseen instance with a higher number of cities (although the increase in the size is not the same as for the first experiment).

V. Conclusion

Fitness landscape analysis has shown promise for better understanding the functionality of optimization algorithms and reducing their unpredictability. In this paper, we proposed a new ML design to predict the FL ruggedness of unseen large instances based on the values of historical small instances. This work, to the best of our knowledge, is the first attempt to take advantage of recent advances in data-driven approaches to analyze and estimate a feature of FL.

This work can be considered as the first step aimed at predicting the characteristics of problem instances in which their calculation is time-consuming. A practical exploitation of any optimization problem is to run the algorithms in the smallest instances, build the machine learning model, and then predict the features on the very large instances. Estimating the characteristics of a very large instance without needing to run the algorithms can be useful, e.g., to choose the appropriate algorithm to solve these instances.

In this paper, the experiment consists in evaluating the predictive capacity of 3 ML algorithms. The data sets in the three experiments are collected by running the 2-opt random walk 30 times on several instances. In our case, random forest was the best suited for the random TSP instances. For the different TSPLIB instances, no algorithm could find satisfactory results. When focusing on a specific TSPLIB problem, the results found by the gradient boosting are the best. We can conclude that machine learning prediction can be useful when we have identical or similar problem instances with difference mainly in number of cities. The contributions are summarized as follows:

- Our study reveals that machine learning models perform better on random TSP instances than on specific TSPLIB instances. This finding suggests that ML can effectively capture patterns in less complex, more uniform problem structures.
- The results highlight the importance of incorporating additional problem-specific features to improve prediction accuracy for TSP instances of different families. Our work lays the groundwork for future studies to explore more sophisticated models and feature sets, advancing the field's understanding of fitness landscape predictability.

These results are consistent with established findings, demonstrating the correlation between robustness and performance, as well as the difference of ML performance depending on the nature of TSP instances. This paper builds on this knowledge by providing significant new findings and results.

As the Concorde solver is able to easily solve many TSP instances of quite large size to optimality, another approach that may be investigated in the study of ruggedness is target analysis, i.e., giving an optimal solution and checking to which extent, possibly using ML, this can be found from a given starting solution and allowing to learn from the path between those solutions. Finally, even if the practical contribution is not well apparent in the above-mentioned data instances, this work can be considered as a first step that can be extended to much larger instances and problems in which the calculation of the factor can be very time-consuming. The exploitation of information of small instances can be much helpful. Further research should then focus on this issue. Indeed, it is of utmost importance to concretely show the practical impact of our approach (e.g. in black-box optimization). Moreover, it is important to further study the practical application of ML by finding the needed sample (number of instances) to have an accurate prediction on the different problem instances. Further research can also focus on applying the approach to other similar problems such as the family TSP or to integrate into other types of metaheuristics. The reason for the poor results seems to be that the three ruggedness prediction models considered are known to yield satisfactory results in continuous domains. In fact, the advancement in ML prediction is an outstanding area of research that could hold promise in estimating the FL features of unresolved instances and studying the links between these features.

The results of this study highlight the significant impact of robustness and landscape structure on algorithm performance. Building on these insights, future research could focus on designing ML-based adaptive optimization algorithms that can dynamically adjust strategies based on landscape features. Furthermore, leveraging ML to efficiently manage large-scale TSP instances could advance the field, especially in real-world applications requiring scalable solutions.

References

[1] Random Forests. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 1054–1054. Springer US, Boston, MA, 2017.

[2] Kenneth D. Boese, Andrew B. Kahng, and Sudhakar Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16(2):101–113, September 1994.

[3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] J. Arjan G. M. de Visser and Joachim Krug. Empirical fitness landscapes and the predictability of evolution. *Nature Reviews Genetics*, 15(7):480–490, July 2014. Publisher: Nature Publishing Group.

[5] D R Hains, L D Whitley, and A E Howe. Revisiting the big valley search space structure in the TSP. *Journal of the Operational Research Society*, 62(2):305–312, February 2011. Publisher: Taylor & Francis _eprint: https://doi.org/10.1057/jors.2010.116.

[6] Dr Genevieve Hayes. gkhayes/mlrose, March 2024. original-date: 2018-10-20T19:48:34Z.

[7] Wim Hordijk. A Measure of Landscapes. *Evolutionary Computation*, 4(4):335–360, December 1996.

[8] André Hottung, Shunji Tanaka, and Kevin Tierney. Deep Learning Assisted Heuristic Tree Search for the Container Pre-marshalling Problem. *Computers & Operations Research*, September 2019.

[9] Frank Hutter, Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206:79–111, January 2014.

[10] T Jones. *Evolutionary algorithms, fitness landscapes and search.* Thesis, The University of New Mexico, 1995.

[11] L. Kallel, B. Naudts, and C. R. Reeves. Properties of Fitness Functions and Search Landscapes. In Leila Kallel, Bart Naudts, and Alex Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, pages 175–206. Springer, Berlin, Heidelberg, 2001.

[12] Maryam Karimi-Mamaghan, Mehrdad Mohammadi, Patrick Meyer, Amir Mohammad Karimi-Mamaghan, and El-Ghazali Talbi. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, 296(2):393–422, January 2022.

[13] Pascal Kerschke and Mike Preuss. Exploratory landscape analysis. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19, pages 1137–1155, New York, NY, USA, July 2019. Association for Computing Machinery.

[14] Pascal Kerschke and Heike Trautmann. Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-Package Flacco. In Nadja Bauer, Katja Ickstadt, Karsten Lübke, Gero Szepannek, Heike Trautmann, and Maurizio Vichi, editors, *Applications in Statistical Computing: From Music Data Analysis to Industrial Quality Improvement*, pages 93–123. Springer International Publishing, Cham, 2019.

[15] Kevin Leyton-Brown, Nudelman Eugene, and Yoav Shoham. Empirical Hardness Models for Combinatorial Auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, page 0. The MIT Press, December 2005.

[16] Arnaud Liefooghe, Fabio Daolio, Sébastien Verel, Bilel Derbel, Hernán Aguirre, and Kiyoshi Tanaka. Landscape-Aware Performance Prediction for Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 24(6):1063–1077, December 2020. Conference Name: IEEE Transactions on Evolutionary Computation.

[17] Katherine M. Malan and Andries P. Engelbrecht. Quantifying ruggedness of continuous landscapes using entropy. In *2009 IEEE Congress on Evolutionary Computation*, pages 1440–1447, May 2009. ISSN: 1941-0026.

[18] Katherine M. Malan and Andries P. Engelbrecht. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences*, 241:148–163, August 2013.

[19] Katherine M. Malan and Andries P. Engelbrecht. Fitness Landscape Analysis for Metaheuristic Performance Prediction. In Hendrik Richter and Andries Engelbrecht, editors, *Recent Advances in the Theory and Application of Fitness Landscapes*, pages 103–132. Springer, Berlin, Heidelberg, 2014.

[20] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. *Advances in neural information processing systems*, 12, 1999.

[21] Umberto Mele, Luca Maria Gambardella, and Roberto Montemanni. Machine Learning Approaches for the Traveling Salesman Problem: A Survey. pages 182–186, January 2021.

[22] Sadegh Mirshekarian and Dušan N. Šormaz. Correlation of job-shop scheduling problem features with scheduling efficiency. *Expert Systems with Applications*, 62:131–147, November 2016.

[23] Mario A. Muñoz, Michael Kirley, and Saman K. Halgamuge. Exploratory Landscape Analysis of Continuous Space Optimization Problems Using Information Content. *IEEE Transactions on Evolutionary Computation*, 19(1):74–87, February 2015. Conference Name: IEEE Transactions on Evolutionary Computation.

[24] Gabriela Ochoa and Nadarajen Veerapen. Additional Dimensions to the Study of Funnels in Combinatorial Landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pages 373–380, New York, NY, USA, July 2016. Association for Computing Machinery.

[25] Gabriela Ochoa and Nadarajen Veerapen. Mapping the global structure of TSP fitness landscapes. *Journal of Heuristics*, 24(3):265–294, June 2018.

[26] Manfred Padberg and Giovanni Rinaldi. Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations Research Letters*, 9(5):353, September 1990.

[27] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[28] Jr. Skipper, Robert A. The Heuristic Role of Sewall Wright's 1932 Adaptive Landscape Diagram. *Philosophy of Science*, 71(5):1176–1188, 2004. Publisher: [The University of Chicago Press, Philosophy of Science Association].

[29] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.

[30] Yuan Sun, Andreas Ernst, Xiaodong Li, and Jake Weiner. Generalization of machine learning for problem reduction: a case study on travelling salesman problems. *OR Spectrum*, 43(3):607–633, September 2021.

[31] M. H. Tayarani-N. and Adam Prügel-Bennett. Anatomy of the fitness landscape for dense graph-colouring problem. *Swarm and Evolutionary Computation*, 22:47–65, June 2015.

[32] Mohammad-H. Tayarani-N. and Adam Prügel-Bennett. On the Landscape of Combinatorial Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 18(3):420–434, June 2014. Conference Name: IEEE Transactions on Evolutionary Computation.

[33] Vesselin K. Vassilev, Terence C. Fogarty, and Julian F. Miller. Smoothness, Ruggedness and Neutrality of Fitness Landscapes: from Theory to Application. In Ashish Ghosh and Shigeyoshi Tsutsui, editors, *Advances in Evolutionary Computing: Theory and Applications*, pages 3–44. Springer, Berlin, Heidelberg, 2003.

[34] Jean-Paul Watson. An Introduction to Fitness Landscape Analysis and Cost Models for Local Search. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics*, pages 599–623. Springer US, Boston, MA, 2010.

[35] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63(5):325–336, September 1990.

[36] Tzu-Tsung Wong and Po-Yang Yeh. Reliable Accuracy Estimates from k-Fold Cross Validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1586–1594, August 2020. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[37] Urban Škvorc, Tome Eftimov, and Peter Korošec. Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis. *Applied Soft Computing*, 90:106138, May 2020.