



**HAL**  
open science

## Exploiting landscape features for fitness prediction in university timetabling

Thomas Feutrier, Nadarajen Veerapen, Marie-Éléonore Kessaci

► **To cite this version:**

Thomas Feutrier, Nadarajen Veerapen, Marie-Éléonore Kessaci. Exploiting landscape features for fitness prediction in university timetabling. GECCO '22 Companion: Companion Conference on Genetic and Evolutionary Computation, Jul 2022, Boston, MA, United States. pp.192-195, 10.1145/3520304.3528910 . hal-03791808

**HAL Id: hal-03791808**

**<https://hal.science/hal-03791808>**

Submitted on 12 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Exploiting Landscape Features for Fitness Prediction in University Timetabling

Thomas Feutrier  
thomas.feutrier@univ-lille.fr  
Univ. Lille, CNRS, Centrale Lille,  
UMR 9189 - CRISTAL  
F-59000 Lille, France

Nadarajen Veerapen  
nadarajen.veerapen@univ-lille.fr  
Univ. Lille, CNRS, Centrale Lille,  
UMR 9189 - CRISTAL  
F-59000 Lille, France

Marie-Éléonore Kessaci  
mkessaci@univ-lille.fr  
Univ. Lille, CNRS, Centrale Lille,  
UMR 9189 - CRISTAL  
F-59000 Lille, France

## ABSTRACT

A small but growing number of papers have shown that landscape metrics can be useful for performance prediction, usually on classic unconstrained problems. In this paper, we consider the Curriculum-Based Course Timetabling problem, a heavily constrained problem known to have very neutral landscapes, and extract over 100 instance and landscape features to construct prediction models. An Iterated Local Search is used to sample the landscape, and the performance of both Simulated Annealing and a Hybrid Local Search algorithm are predicted using linear regression. Using as few as 4 features obtained via feature selection, our simple models are able to accurately predict the final fitness for either approach with an R-squared of approximately 0.95.

## CCS CONCEPTS

• **Computing methodologies** → **Discrete space search**; *Machine learning*.

## KEYWORDS

University timetabling, performance prediction, landscape analysis, local search

### ACM Reference Format:

Thomas Feutrier, Nadarajen Veerapen, and Marie-Éléonore Kessaci. 2022. Exploiting Landscape Features for Fitness Prediction in University Timetabling. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion)*, July 9–13, 2022, Boston, MA, USA. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3520304.3528910>

## 1 INTRODUCTION

University Timetabling problems have been widely studied both because of scientific interest, as NP-hard optimization problems, and out of practical interest for better timetables for institutions, staff and students. Research into the topic is further stimulated by the International Conference on the Practice and Theory of Automated Timetabling (PATAT) and the associated International Timetabling Competition (ITC). Fitness landscapes [12] and their analysis can help to understand the nature of the search space. Over the last ten years or so, landscape analysis has moved from an admittedly

mainly theoretical construct to being a more practical tool [6, 7]. A small but growing number of papers have shown that landscape analysis and associated metrics can be used to build relevant and accurate models. In the combinatorial context, Daolio et al. [1] and Liefvooghe et al. [5] respectively applied mixed-effects multi-linear regression and random forest models to multiobjective combinatorial optimization, and Thomson et al. [13] considered random forests and linear regression on the Quadratic Assignment Problem. In this work, we consider one specific University Timetabling problem: the Curriculum-Based Course Timetabling (CB-CTT). We first use an Iterated Local Search (ILS) to explore and sample the search space and gather landscape information for problem instances introduced in the ITC 2007 competition. We describe and analyze the main features to find common behaviors. Secondly, we use the collected landscape features to build a model and predict the best fitness that simulated annealing and a hybrid local search, the winner of ITC 2007 [8], can attain. In particular, we use feature selection to identify the most useful features and compare different linear regression models. We find that a select few features can be used to build accurate regression models for both solvers.

The paper is organized as follows: Section 2 presents university timetabling; Section 3 introduces relevant definitions; the experimental protocol is in Section 4; features used are in Section 5; Section 6 describes the preprocessing and evaluation procedure of the models; the models themselves and results are presented and discussed in Section 7; the conclusion is found in Section 8.

## 2 UNIVERSITY TIMETABLING

University Timetabling problems belong to the family of Scheduling problems and include different variants such as Examination Timetabling and University Course Timetabling.

Curriculum-Based Course Timetabling CB-CTT uses the notion of curriculum. A student subscribes to one curriculum. This is the case in many European universities.

ITC 2007 was especially important for CB-CTT because it formalized a number of instances and the runtime limit of 5 minutes meant that the solving methods needed to be fast metaheuristics, including the hybrid local search we consider in this paper. The next and most recent competition to feature University Timetabling was ITC 2019 which considered a more complex, and realistic, problem formulation with additional constraints. In addition no time limit was imposed, which favored exact approaches. For this present work, using the ITC 2007 setting is therefore more suitable.

*Curriculum-Based Course Timetabling (CB-CTT)*. In CB-CTT, a curriculum is a simple package of courses. Courses are sets of

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*GECCO '22 Companion*, July 9–13, 2022, Boston, MA, USA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9268-6/22/07.

<https://doi.org/10.1145/3520304.3528910>

lectures and have only one referent teacher. Teachers can teach an unlimited number of different courses. One course can belong to several different curricula. In this case, all students from curricula attend lectures at the same time in the same room. The problem is scheduled over a limited number of days, divided into periods or timeslots. One period corresponds to the duration of one lecture.

A solution consists in scheduling the lectures in the timeslots and available rooms following constraints. These are of two types: hard and soft. Hard constraints must always be respected. A timetable is said to be feasible when all the hard constraints are met. An example of a hard constraint is the rule called Conflicts that prevents one teacher from teaching two lectures at once. On the contrary, soft constraints are optional and represent targets to strive for.

The fitness function is a weighted sum of 4 soft constraint violations: room capacity, minimum working days, curriculum compactness and room stability. Solutions that do not respect all the hard constraints are considered as infeasible, irrespective of the value computed with the fitness function.

### 3 SEARCH LANDSCAPE

Educational timetabling, and Curriculum Based-Course Timetabling in particular, is known to be a very neutral problem [9]. Neutrality may hinder the solving process because finding a suitable trajectory in the search landscape becomes more difficult as it is not easy to discriminate between solutions having the same fitness. In this context, landscape analysis can help to understand the nature of the search space, for instance by characterizing the ruggedness of the landscape or its connectivity patterns.

#### 3.1 Definitions

*Fitness Landscape.* A landscape [12] may be formally defined as a triplet  $(S, N, f)$  where

- $S$  is a set of solutions, or search space,
- $N : S \rightarrow \wp(S)$ , the neighborhood structure, is a function that assigns, to every  $s \in S$ , a set of neighbors  $N(s)$  ( $\wp(S)$  is the power set of  $S$ ), and
- $f : S \rightarrow \mathbb{R}$  is a fitness function.

*Plateau.* A plateau is usually defined as a set of connected solutions with the same fitness value. Two solutions  $s_1$  and  $s_2$  are linked if they are neighbors, i.e.,  $s_2 \in N(s_1)$ . With our ILS sampling, we identify plateaus as sequences of consecutive solutions with the same fitness. Given the trajectory-based nature of the sampling, it is generally not possible to identify whether two separate sequences with the same fitness are part of the same plateau.

*Local optimum.* A local optimum is a solution  $s^* \in S$  such that  $\forall s \in N(s^*), f(s^*) \leq f(s)$ . In order to allow for plateaus and neutral landscapes, the inequality is not strict. Minimization is considered since we wish to minimize constraint violations.

A number of landscape metrics can be measured by building Local Optima Networks (LONs) [14]. These provide compressed graph models of the search space, where nodes are local optima and edges are transitions between them according to some search operator. LONs for neutral landscapes have been studied before by Verel et al. [15]. We will consider two slightly different kinds of networks: Timeout Plateau Networks and Fitness Networks.

*Timeout Plateau.* The ILS we use to sample the search space contains a hill-climber that stops if it remains on the same plateau for too long (more precisely 50,000 consecutive iterations at the same fitness). We call the last plateaus thus found timeout plateaus because the hill-climber has not been able to escape from them within a given maximum number of iterations.

*Timeout Plateau Network.* A Timeout plateau network is a graph where each node is a contracted timeout plateau and an edge represents a transition between two such plateaus. In practice, here this transition is an ILS perturbation followed by neutral acceptance hill-climbing. Given the high degree of neutrality of the problem, it is unlikely that sampling with an ILS across multiple runs will find many common solutions. Therefore our Timeout Plateau Networks do not exhibit much connectivity.

*Fitness Network.* This is a simplification of Timeout Plateau Networks where all nodes with the same fitness are contracted together. This provides a graph structure with much higher connectivity than a Timeout Plateau Network. From there, a number of different graph metrics can be computed. Note that this is an even greater simplification than *Compressed LONs* [10] which only aggregate nodes sharing the same fitness that are connected at the LON level.

### 4 EXPERIMENTAL PROTOCOL

Experiments use 19 of the 21 instances proposed for ITC 2007. Instances 01 and 11 are set aside because they are simple to solve. In this paper, our first solver of choice is the Hybrid Local Search (HLS) proposed by Müller [8]. It combines Hill-Climbing, Great Deluge and Simulated Annealing algorithms in an Iterated Local Search and won ITC 2007. In addition, an iterated Simulated Annealing (SA) is tested. These solvers are tailored to face neutrality and find good solutions quickly [2] since the ITC 2007 time limit was 5 minutes. All executions run on Intel Xeon Silver 4114 CPUs @ 2.20GHz, HLS and SA have a time budget of 5 minutes. Performance data was gathered by running these algorithms 100 times on each instance.

*Sampling the Search Space.* To sample our search space, we use an Iterated Local Search (ILS) based on algorithmic components found in the HLS mentioned above. ILS saves information used to generate networks and their features. In particular we use the same Hill-Climbing and a specific perturbation called Iterative Bounded Perturbation. The Hill-Climbing follows a first improvement strategy and accepts equal and better solutions as new current (neutral acceptance). It stops when it finds a local optimum or when it has encountered 50,000 consecutive solutions with the same fitness, which happens most of the time.

The ILS sampling method memorizes information about all timeout plateaus traveled during the optimization phase. However, due to memory constraints, it only saves a representation of the first and the last solutions of those plateaus. At the end of each run, we keep information that is relevant to compute the metrics mentioned in Section 5.2. Each ILS is run with a budget of 30 seconds and 100 runs are performed.

## 5 FEATURES

### 5.1 Instance Features

Instance features include basic descriptive data such as the number of courses, curricula, lectures, and days. Other measures quantify the complexity of the problem. *Lectures by Course* counts the minimum, maximum and average number of lectures for a course for one problem instance. *Teachers Needed* computes the average number of teachers needed for one timeslot, the number of lectures divided by the number of timeslots. Finally, the number of courses divided by the number of curricula is also used. As curricula create additional constraints, this feature measures the difficulty of scheduling without violations. In total, there are 24 instance features.

### 5.2 Landscape metrics

Local optima networks (LONs) are useful to describe and analyze combinatorial optimization problems [11]. LONs capture relevant topological features of the underlying search spaces and can be adapted and transformed to better represent search space with neutrality [15]. In LONs, a node represents a local optimum (LO) and direct edges the possibility to move from a LO (initial node) to another LO (terminal node). When the problem is neutral, many LO share the same height (i.e., fitness value) and so, the numbers of nodes in the LON can be huge and make difficult the LON analysis.

This network model brings metrics, initially derived from graph theory, to characterize the structure of combinatorial landscapes. These metrics can also be applied to our Fitness Networks.

**5.2.1 Node-Level Metrics.** A first set of metrics relates to what the nodes represent, including plateaus and the number of solutions within. A second set of metrics relates to the connectivity of the nodes within the network, including the degrees and weights. We also consider two variants of weight and degree metrics. For some given node, the *better* (resp. *worse*) variant only considers arcs between this node and better (resp. *worse*) nodes.

The above metrics are computed for each node and five points corresponding to the quartiles (Q1, median and Q3) and the 10th and 90th percentiles of the distribution are used as features for our models. The number of features calculated with node-level metrics amounts to 65 features.

**5.2.2 Network Metrics.** To describe the networks themselves, we used other measures, including mean fitness, number of sinks and coefficients of assortivity and transitivity.

Sink nodes are ones that do not have any outgoing arcs to nodes with better fitness. Assortivity is a measure of similarity between linked nodes [11]. The more numerous the connected nodes with the same attributes, the higher the coefficient. The transitivity coefficient, also called the clustering coefficient, is the probability of a link between adjacent neighbors and one chosen vertex [11].

Finally, we use the PageRank centrality to analyze the networks since Herrmann et al. [3] showed that related metrics were useful when working with LONs. In total, we consider 23 network metrics.

### 5.3 Identifying Major Sections of the Landscape

To identify and measure the most and least promising regions of the network, we plotted the cumulative distribution of some promising features as a function of fitness in order to identify low and high

fitness regions. From the sigmoid shape of the distributions, we identified three groups of nodes: A) The first sub-network has a low local density. Its fitness values are little visited and are the best found. B) This set of nodes represents a big part of networks. Nodes correspond to good fitness values, and solving methods often find them. Vertices are inter-connected and arcs are frequently traveled, with high weights. C) The nodes in this group are almost all of size one. They represent the worst fitness values found.

In order to automate the partitioning of nodes into the above groups, we identify two kinks on the cumulative distribution of plateaus using the following method. The first kink is found with a threshold set at 1 percentage point. If the percentage point difference between two consecutive fitness values represents a variation greater than 1 point, the first fitness values are part of Group A and the following ones are from Group B. Afterwards, the second kink is identified when the difference drops below 1 point, and the remaining fitness values are in Group C.

For sub-networks A and B, the mean fitness, number of nodes, number of plateaus, and the number of sink nodes are computed. There are thus 8 features relative to sub-networks.

## 6 MODEL CONSTRUCTION AND EVALUATION

A small but growing number of landscape analysis papers [1, 5, 13] have successfully shown that landscapes contain meaningful information that is linked to search algorithm performance. We attempt here to test this assumption on the CB-CTT problem which, contrary to other problems investigated, is highly neutral. Furthermore, we employ a different model building process consisting in a feature selection followed by linear regression.

**Pre-processing.** Before the preprocessing phase, our total number of features amounts to 120. Clearly, not all features can be useful. First, the preprocessing algorithm removes features with a constant value. For instance, the 90th percentile metrics concerning the degree and weight are, for the most part, equal to 100%. This phase removes 21 features. The features are then standardized. Then we use correlation preselection to select promising features. It selects all features correlated with fitness above a fixed threshold. We consider four correlation thresholds: 0.9, 0.8, 0.7 and 0. Then the process is repeated for predicting Simulated Annealing results.

**Evaluation.** We use linear regression on the selected features to build fitness prediction models for SA and HLS. In order to obtain a robust evaluation of the models, complete group 5-fold cross-validation is used [4]. This complete approach has two main advantages. The first is to check whether the model can predict final fitness for new instances. The second smooths out the impact of how the data are split between training and test sets. When two problem instances are very similar and are not in the same fold, information about first helps prediction. However, our objective is to obtain a robust model for all problem instances and not only very similar instances. Testing all combinations reduces this effect.

We consider two indicators computed through cross-validation. Firstly, the Mean Absolute Error (MAE) computes the mean of the absolute differences between the predicted and real value. Secondly, the coefficient of determination,  $R^2$ , is a well-known indicator for regression tasks. A good model should have an  $R^2$  close to 1.

## 7 MODELS AND RESULTS

For both HLS and SA, we build regression models to predict fitness using three sets of features obtained via feature selection and one considering all features. Cross-validation results are presented in Table 1. As we can see, the feature selection preprocessing step is a key element of obtaining a good linear regression model. When all features are considered the models obtained perform very poorly, with a  $R^2$  below 0.5 and an arbitrarily poor MAE. The models obtained with the 0.7 correlation threshold do not fare very well either. In contrast, setting the correlation threshold for feature selection to 0.8 and 0.9 produce good regression models for both HLS and SA, with an  $R^2$  of about 0.95 and relatively low MAE. The behavior of the best models is quite similar. It is especially interesting that this good performance holds for both HLS and SA. While one might have expected HLS performance to be somewhat in line with a sampling method based on the same hill-climbing found within HLS, SA is quite different in how it operates.

Previously, Thomson et al. [13] used a specific sampling method to accurately predict the performance of both an ILS and Tabu Search. While search landscapes are intrinsically linked to the solving method, our result, together with theirs, may point to common landscape information that may be extracted and used to predict the behavior of different algorithms, as long as some core algorithmic components, such as the neighbor function, are preserved. However, Thomson et al. found that linear regression performed very poorly on their problem (QAP) while random forests worked very well. Speculatively, this difference could be explained by any number of factors, including the neutral nature of our problem, the specific features or the feature selection process.

Together with similar performance with the 0.8 and 0.9 thresholds for HLS and SA, the number of selected features is the same. All models use mean fitness features, including the mean fitness computed across the Group A and B sub-networks (Section 5.3). They correspond respectively to the best fitnesses and improving fitness subsets. The number of curricula, an instance feature, is also present throughout. At the 0.8 level, three additional features are used: course curriculum and the number of sinks. The latter was also previously found to have predictive power by Ochoa et al. [10] in a different context.

Our results highlight that a mix of landscape and instance features are useful for prediction, and that these predictions can be fairly accurate. Naturally, there are limitations to the approach and the present results will not necessarily translate to predicting the performance of another algorithm or problem.

**Table 1: Evaluation of regression models predicting fitness for HLS and SA, obtained on different feature sets returned by the feature selection. NB is the number of selected features.**

Correlation Threshold	HLS			SA		
	NB	$R^2$	MAE	NB	$R^2$	MAE
0.90	4	0,96	5,30	4	0,94	8,86
0.80	7	0,95	5,89	7	0,937	9,46
0.70	20	0,74	$> 10^{10}$	19	0,6	$> 10^{10}$
–	all	0,4	$> 10^{10}$	all	0,4	$> 10^{10}$

## 8 CONCLUSION

In this paper, we sampled the search landscape of Curriculum-Based Course Timetabling instances using an ILS to gather a number of landscape metrics. We considered a very simplified version of the landscape, the fitness network, and identified three subsets of nodes where we expected the search to have different behaviors.

After performing feature selection on instance and landscape features, we built linear regression models for simulated annealing and a hybrid local search algorithm. The better models, using 4 and 7 features, performed well on both algorithms. Our work therefore follows in the footsteps of a small but growing number of papers that show that landscape features are useful for performance prediction. Our results tend to indicate that this is also valid for neutral landscapes, and that even simple models like linear regression can be useful. There are naturally a number of limitations to this study, notably that landscapes are problem-, instance-, and algorithm-specific.

In future work, we intend to continue working on these issues in order to improve the use of landscape analysis in practical situations. In particular, we will focus on what makes a landscape useful for predicting performance across different algorithms.

## REFERENCES

- [1] Daolio, F., Liefoghe, A., Verel, S., Aguirre, H., Tanaka, K.: Problem Features vs. Algorithm Performance on Rugged Multiobjective Combinatorial Fitness Landscapes. *Evolutionary Computation* **25**(4), 555–585 (2017)
- [2] Feutrier, T., Kessaci, M.E., Veerapen, N.: Investigating the landscape of a hybrid local search approach for a timetabling problem. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1665–1673. GECCO '21, Association for Computing Machinery, New York, NY, USA (Jul 2021)
- [3] Herrmann, S., Ochoa, G., Rothlauf, F.: Pagerank centrality for performance prediction: the impact of the local optima network model. *J. Heuristics* **24**(3), 243–264 (2018)
- [4] Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, p. 1137–1143. IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
- [5] Liefoghe, A., Daolio, F., Verel, S., Derbel, B., Aguirre, H., Tanaka, K.: Landscape-Aware Performance Prediction for Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation* **24**(6), 1063–1077 (Dec 2020)
- [6] Malan, K.M., Engelbrecht, A.P.: A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences* **241**, 148–163 (2013)
- [7] Malan, K.M.: A survey of advances in landscape analysis for optimisation. *Algorithms* **14**(2) (2021)
- [8] Müller, T.: ITC2007 solver description: a hybrid approach. *Annals of Operations Research* **172**(1), 429–446 (Nov 2009)
- [9] Ochoa, G., Qu, R., Burke, E.K.: Analyzing the landscape of a graph based hyperheuristic for timetabling problems. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2009)* (2009)
- [10] Ochoa, G., Veerapen, N., Daolio, F., Tomassini, M.: Understanding Phase Transitions with Local Optima Networks: Number Partitioning as a Case Study. In: Hu, B., López-Ibáñez, M. (eds.) *Evolutionary Computation in Combinatorial Optimization*, pp. 233–248. *Lecture Notes in Computer Science*, Springer International Publishing, Cham (2017)
- [11] Ochoa, G., Verel, S., Daolio, F., Tomassini, M.: Local Optima Networks: A New Model of Combinatorial Fitness Landscapes. In: Richter, H., Engelbrecht, A. (eds.) *Recent Advances in the Theory and Application of Fitness Landscapes*, pp. 233–262. *Emergence, Complexity and Computation*, Springer (2014)
- [12] Stadler, P.F.: Fitness landscapes. In: Lässig, M., Valleriani, A. (eds.) *Biological Evolution and Statistical Physics*, pp. 183–204. *Lecture Notes in Physics*, Springer (2002)
- [13] Thomson, S.L., Ochoa, G., Verel, S., Veerapen, N.: Inferring future landscapes: Sampling the local optima level. *Evolutionary Computation* **28**(4), 621–641 (2020)
- [14] Tomassini, M., Verel, S., Ochoa, G.: Complex-network analysis of combinatorial spaces: The  $nk$  landscape case. *Phys. Rev. E* **78**, 066114 (Dec 2008)
- [15] Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of  $nk$  landscapes with neutrality. *IEEE Transactions on Evolutionary Computation* **15**(6), 783–797 (2011)