Analysis of Search Landscape Samplers for Solver Performance Prediction on a University Timetabling Problem

Thomas Feutrier, Marie-Éléonore Kessaci, Nadarajen Veerapen

Analysis of Search Landscape Samplers for Solver Performance Prediction on a University Timetabling Problem

Thomas Feutrier[0000−0003−0854−3176], Marie-Éléonore Kessaci[0000−0002−4372−5162], and Nadarajen Veerapen[0000−0003−3699−1080]

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
{thomas.feutrier, mkessaci, nadarajen.veerapen}@univ-lille.fr

Abstract. Landscape metrics have proven their effectiveness in building predictive models, including when applied to University Timetabling, a highly neutral problem. In this paper, two Iterated Local Search algorithms sample search space to obtain over 100 landscape metrics. The only difference between the samplers is the exploration strategy. One uses neutral acceptance while the other only accepts strictly improving neighbors. Different sampling time budgets are considered in order to study the evolution of the fitness networks and the predictive power of their metrics. Then, the performance of three solvers, Simulated Annealing and two versions of a Hybrid Local Search, are predicted using a selection of landscape metrics. Using the data gathered, we are able to determine the best sampling strategy and the minimum sampling time budget for models that are able to effectively predict the performance of the solvers on unknown instances.

Keywords: University timetabling, performance prediction, landscape analysis, local search

1 Introduction

University Timetabling is an active research area in combinatorial optimization. Research into the topic is stimulated, notably, by the International Timetabling Competition (ITC) organized at PATAT. Many different heuristics [6] have been proposed to solve University Timetabling problems, including both local search [12] and crossover-based algorithms [18]. Given the complex nature of the problems, hybrid and hyper-heuristic approaches are also well suited [17].

Fitness landscapes [19] 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 [8,10].

In general, the whole landscape cannot be enumerated and sampling is required. One such approach relies on gathering the traces of Iterated Local Search (ILS) runs. This is the approach taken in this paper, where we compare two ILS

samplers with different exploration strategies. We wish to investigate whether considering the neutrality of the landscape will provide a better sample or not. We rely on the predictive ability of performance models built from features derived from the samples as a proxy for the quality of the samples.

In the context of performance prediction, some papers have shown that landscape analysis and associated metrics can build accurate models [10]. In particular, when considering continuous optimization, Bisch et al. [1] have used support vector regression models, Muñoz et al. [11] considered neural networks to predict the performance of CMA-ES, Malan and Engelbrecht [9] used decision trees to predict failure in particle swarm optimization, while Jankovic and Doerr [4] considered random forests and CMA-ES. In the combinatorial context, Daolio et al. [2] and Liefooghe et al. [7] respectively applied mixed-effects multi-linear regression and random forest models to multiobjective combinatorial optimization, and Thomson et al. [20] considered random forests and linear regression on the Quadratic Assignment Problem, finding that random forests performed better.

In this paper, we consider the Curriculum-Based Course Timetabling problem (CB-CTT). We first use two variants of an ILS to explore and sample the search space for problem instances of the ITC 2007 competition [16] across 4 different time budgets. After a feature selection step, a model is built for each time budget and each sampler. Each model is evaluated, via cross-validation, according to its ability to predict the final fitness on 3 different solvers on unseen instances. Our results show that, on our instances, sampling the search space with 100 ILS runs of 5 seconds each allows us to build models that can accurately predict fitness across solvers.

The paper is organized as follows: Section 2 presents our problem; Section 3 introduces fitness landscapes and relevant definitions; we develop our experimental protocol in Section 4; the features used in our models are laid out in Section 5; Section 6 analyzes the effects of the sampler time budget on networks; Section 7 describes the preprocessing involved in building the models as well as the evaluation procedure; the models obtained are discussed in Section 8; finally, Section 9 concludes the paper and outlines potential for future research.

2 Curriculum-Based Course Timetabling

The paper focuses on a specific University Timetabling problem: Curriculum-Based Course Timetabling (CB-CTT). ITC 2007 was especially important for CB-CTT because it formalized several instances and imposed a runtime limit of 5 minutes, encouraging the use of metaheuristics instead of exact solvers.

CB-CTT is centered on the notion of a curriculum, that is a simple package of courses. This may be a simplification of the real-life problem since a student can choose only one curriculum. Curricula cluster students together, with each change of event impacting all students in the same way.

Courses are sets of lectures and are taught by one teacher. One course can belong to several different curricula. In this case, all students across curricula

attend lectures together. The problem is scheduled over a limited number of days, divided into periods or timeslots. One period corresponds to the duration time of one lecture.

In CB-CTT, a solution consists of scheduling lectures in timeslots and available rooms following hard and soft constraints. 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 preventing one teacher from teaching two lectures at once. On the contrary, soft constraints are optional and generally represent targets to strive for. The violations of each soft constraint are represented as a function to minimize.

The objective function to optimize for CB-CTT is a weighted sum of the constraint violations: $f(s) = \sum_{i=1}^{4} \mathit{SoftConstraints}_i(s) * \omega_i$, where S represents a timetable. The weights, as used for ITC 2007, are set to 1, 5, 2 and 1 for ω_1, ω_2, ω_3 and ω_4 respectively. $\mathit{SoftConstraints}_i(s)$ represents the number of violations for the soft constraints listed below.

1. *RoomCapacity*: All students can be sat in the room.
2. *MinWorkingDays*: A course has lectures which should be scheduled within a minimum number of days.
3. *CurriculumCompactness*: A student should have always two consecutive lectures before a gap.
4. *RoomStability*: Lectures of a course should be in the same room.

3 Search Landscape

Educational timetabling, and CB-CTT in particular, is known to be a very neutral problem [13]. This means that a large number of similar solutions share the same fitness value. Neutrality may hinder the solving process because finding a suitable trajectory in the search landscape becomes more difficult. 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.

The notion of landscape is strongly tied to the algorithm and neighborhood operators used to explore the search space.

Landscape. A landscape [19] may be formally defined as a triplet (S, N, f) where

- S is a set of solutions, or search space,
- $N : S \longrightarrow \wp(S)$, 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
- $f : S \longrightarrow \mathbb{R}$ is a fitness function.

In CB-CTT, a problem instance establishes fixed relationships between curricula, courses, lectures and teachers. A solution then describes the tripartite graph that links lectures, rooms and timeslots together. We choose to implement this as an object-oriented representation where lecture, room and timeslot objects are connected together as appropriate.

We use 6 classic timetabling neighborhood operators: 3 basic ones and 3 designed to deal with specific soft constraints. A link between two solutions in our landscapes is the result of the application of any one of these operators. The 3 simplest operators focus, at each call, on a single lecture. `RoomMove` and `TimeMove` respectively change the room and the timeslot. `LectureMove` changes the room and timeslot at the same time. The 3 other operators attempt to lower the violation penalty of their associated soft constraint using a combination of the basic moves. For instance, the `CurriculumCompactnessMove` identifies an isolated lecture in a curriculum and performs a `TimeMove` to bring it closer to another lecture.

Besides exhaustive exploration that gives a perfect model of the landscape, any other sampling method will only provide an approximation. In our case, exhaustive exploration is computationally infeasible. We therefore rely on an ILS to sample the search space, as has been done for instance by Ochoa et al. [14]. We focus in particular on local optima.

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 deal with constraint violations.

A number of landscape metrics can be measured by building Local Optima Networks (LONs) [21]. 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. [22]. The latter work introduces the concept of *Local Optimum Neutral Network* that considers that a neutral network is a local optimum if all the configurations of the neutral network are local optima. Another approach to neutrality in LONs is by Ochoa et al. [14] who develop the notion of *Compressed LONs* where connected LON nodes of equal fitness are aggregated together. For our purposes, we will consider two slightly different kinds of networks: Timeout Plateau Networks and Fitness Networks.

Plateau. A plateau, sometimes called a neutral network, is usually defined as a set of connected solutions with the same fitness value. Two solutions s_1 and s_2 are connected if they are neighbors, i.e., $s_2 \in N(s_1)$. Depending on the neighborhood function, we may also have $s_1 \in N(s_2)$. Plateaus are defined as sequences of consecutive solutions with the same fitness.

Timeout Plateau. The first ILS sampler we consider, $\text{ILS}_{\text{neutral}}$, contains a hill-climber that stops if it remains on the same plateau for too long (50,000 consecutive evaluations at the same fitness). Furthermore, the hill-climber used accepts the first non-deteriorating move (i.e., an improving or a neutral neighbor). We call the last plateaus thus found *timeout plateaus* because the hill-climber has not been able to escape from them within a given number of iterations. However they are not necessarily a set of local optima. Some exploratory analysis showed at least 1 % of these solutions were not actual local optima. The other sampler,

$\text{ILS}_{\text{strict}}$, has the same components but will only accept a strictly improving solution. Its timeout plateaus therefore trivially only contain one solution.

Timeout Plateau Network. A Timeout plateau network is a graph where each node represents one timeout plateau and an edge represents a transition between two such plateaus. Here this transition is an ILS perturbation followed by hill-climbing. Timeout Plateau Networks are a set of independent chains, where each represents one ILS run.

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. While it is not meant as an accurate representation of the landscape, several different metrics related to connectivity between fitness levels can be computed. Note that this is an even greater simplification than *Compressed LONs* [14] which only aggregate nodes sharing the same fitness that are connected together 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 they are very easy to solve. All instances contain between 47 and 131 courses, 138–434 lectures, 52–150 curricula, 25–36 timeslots and 9–20 rooms.

Our solver of choice is the Hybrid Local Search (HLS) proposed by Müller [12]. It combines Hill-Climbing (HC), Great Deluge and Simulated Annealing algorithms in an Iterated Local Search and won ITC 2007. Our experiments use two versions of the HLS. We will refer to the default version, that accepts equal or better solutions during HC, simply as HLS. The other, $\text{HLS}_{\text{strict}}$, uses a strict acceptance criterion for HC. In addition, iterated Simulated Annealing (SA) is tested. We reuse the SA component found within HLS and place it inside a loop that stops when runtime budget is reached. All executions run on Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz , all solvers have a time budget of 5 minutes and were tested 100 times on each instance.

Sampling the Search Space. To sample our search space, we use one of two ILS, based on algorithmic components found in the HLS mentioned above. Both ILS include an Iterative Bounded Perturbation (IBP) and use a hill-climber (HC) following a first improvement strategy. HC stops when it finds a local optimum or when it has evaluated 50,000 solutions without strict improvement. The distinction between the two ILS samplers lies in the acceptance criterion. The first follows the same strategy as HLS, i.e., accepting any equal or better solution; it is called $\text{ILS}_{\text{neutral}}$. Second sampler, $\text{ILS}_{\text{strict}}$, follows the strategy of $\text{HLS}_{\text{strict}}$, accepting only strictly improving solutions.

IBP takes a baseline fitness, $Fit_{FirstSol}$, corresponding to the fitness of the first solution after construction. Then it deteriorates the final solution found by

HC, $Fit_{LastSol}$, to reach a solution with fitness equal to $Bound = Fit_{LastSol} + 0.1(Fit_{FirstSol} - Fit_{LastSol})$.

Due to memory constraints, $\text{ILS}_{\text{neutral}}$ only records fitness and size for all timeout plateaus met. $\text{ILS}_{\text{strict}}$ just saves the fitness of last solution of each HC, its timeout plateaus have a size of 1. Both ILS samplers are run 100 times with a time budget of {5, 10, 20, 30} seconds per run on each instance. This budget was set in order to obtain enough predictive information on the landscape. Each (time budget, sampler) pair produces one network, so we have 8 networks for each instance.

Timeout Plateau Network. Each Timeout Plateau Network is built from the data gathered from 100 runs. Each node of the network is a timeout plateau and each directed edge is a transition corresponding to a perturbation followed by a hill-climber. At this stage the weight of a directed edge is 1 and there is no connectivity between the trajectories of the individual ILS sampler runs. Thus, a contraction step is required to obtain further information.

Fitness Network. Given the high level of neutrality of the problem, we have chosen to consider that all solutions with the same fitness belong to a single wide plateau. This hypothesis allows us to obtain a connected network very easily. The contraction process of the Timeout Plateau Network into a Fitness Network preserves all the data required to compute the metrics mentioned in Section 5.2. The new weights on the directed edges correspond to the sum of the contracted directed edges.

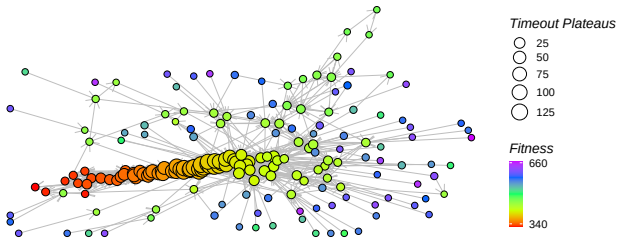

Fig. 1: Fitness Network of Instance 12 built by $\text{ILS}_{\text{neutral}}$ with 30 seconds.

5 Features

Gathering a large number of features ranging from instance features to global and local landscape metrics increases the probability of finding combinations of them that contain complementary information for prediction.

5.1 Instance Features

Instance features include descriptive data about the problem instance. The most basic ones count the courses, curricula, lectures, and days. Others quantify the complexity of the problem. `LecturesByCourse` counts the minimum, maximum and average number of lectures for one course. `TeachersNeeded` is the number of lectures divided by the number of timeslots. Finally, `CourseByCurriculum` is number of courses divided by the number of curricula. It measures the difficulty of scheduling without violations. In total, there are 24 instance features.

5.2 Landscape metrics

We consider different metrics that are computed on the landscape.

Node-Level Metrics. A first set of metrics relates to what the nodes represent. `Plateaus` is the number of plateaus that have been contracted to form the node. It is used as the size property of nodes in Figure 1. `Size` is the sum of number of accepted solutions in the contracted plateaus. `Fit` corresponds to the Fitness of all Timeout Plateaus represented by the current node. `Loops` is the number of consecutive loops on the same fitness, an estimator for attraction power.

A second set of metrics relates to the connectivity of the nodes within the network. Each connectivity metric has 9 variants by combining whether *all*, *ingoing* or *outgoing* edges are considered, together with whether we considered directed edges that reach *any* node or only select the ones that reach *better* (resp. *worse*) nodes. The `Degree` metric measures the number of different arcs connected to the nodes. The `Weight` metric represent the number of times the samplers have passed from one timeout plateau to another.

We also consider two variants of weight and degree metrics. For some given node, the *better* (resp. *worse*) variant only considers directed edges 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.

Network Metrics. To describe the networks themselves, we used additional metrics including the Mean fitness, the number of timeout plateaus, nodes and edges. Moreover, the number of sink nodes is stored, as well as the matching coefficients of assortivity and transitivity.

Sink nodes are ones that do not have any outgoing edges to nodes with better fitness. Assortivity is a measure of similarity between linked nodes [15]. 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 [15]. In total, we consider 23 network metrics.

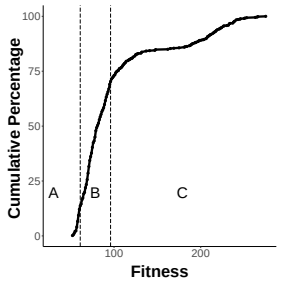

Fig. 2: Cumulative percentage of the number of plateaus as a function of fitness, and the associated groups, for $\text{ILS}_{\text{neutral}}$ with 30 seconds.

We also wish to identify and quantify the most and least promising regions of the network. To do so, we consider the cumulative percentage distribution of the number of plateaus with respect to fitness, as illustrated in Figure 2 for a representative instance, and split the network into 3 groups:

- *Group A*: The first sub-network has a low local density. Its fitness values are little visited and are the best found.
- *Group B*: This set of nodes represents a big part of the 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.
- *Group C*: The nodes in this group are almost all of size one. They represent the worst fitness values found. They are not connected to each other because their arcs lead only to vertices belonging to Group B.

For all instances, the distributions are of the same shape. That implies that behaviors are similar. In order to automate the partitioning of nodes into the above groups, we identify two points of inflection on the curve as follows. The first point is found when a percentage difference of at least 1 percentage point is observed. If the difference in percentage between two consecutive fitness values represents a variation greater than 1 percentage point, the first fitness values are part of Group A and the following ones are from Group B. In the cases where this point is reached very early, the 10 best fitness values are assigned to Group A, as in Figure 2. Afterwards, the second point is identified when the difference drops below 1 percentage 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 Effects of Sampler Choice on Sampled Fitness Values

Here we consider the effects of the (sampler, time budget) pairs on fitness distribution. Recall that the samplers are two Iterative Local Search algorithms,

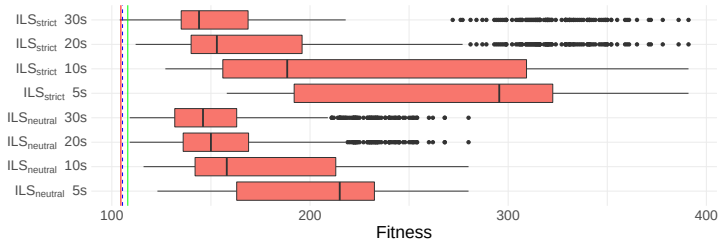

Fig. 3: Fitness distribution for each (sampler, time budget) pair on instance 21. Vertical lines represent the mean fitness values for each solver.

$\text{ILS}_{\text{strict}}$ and $\text{ILS}_{\text{neutral}}$, differing in the neutral or strict acceptance strategy of neighboring solutions.

Figure 3 shows the fitness distribution of the timeout plateaus obtained by each sampler on one representative instance as boxplots. As might be expected, we can observe a clear relationship between the time budget and the skew of the distribution towards better solutions. This stands for both samplers. The best solutions found with 20 and 30 second sampling runs reach, or are very close to, the mean fitness value obtained by the solvers when run for 5 minutes. In addition, it stands to reason that as time increases, timeout plateaus are added but those found within shorter time budgets remain, only they represent a smaller proportion. The tighter time budgets also, naturally, sample fewer solutions. The study of network features shows that networks become sparser in terms of connectivity as time is reduced. This behavior is similar for both samplers.

The fact that some sampling scenarios reach, or are very close to, the mean solver fitness likely indicates that any feature that encodes some information about the best sample fitness will be very important to the model. In our case this is exactly what $\overline{Fit}_{\text{A}}$, the mean fitness of Group A, does. This is investigated further in Section 8.

The boxplots also show that the two samplers do not have the same behavior. It is clear that $\text{ILS}_{\text{neutral}}$ finds, within the same time, better solutions in terms of fitness. Furthermore, Figure 3 reveals that as the time decreases, the gap between the two ILS widens. With the neutral acceptance policy, the first plateau from which the ILS needs to escape via a perturbation is further down the landscape. These observations imply that $\text{ILS}_{\text{neutral}}$ is the most efficient and the most robust in finding better solutions in the face of time. Thus, the neutral strategy improves ILS performance and sampling effectiveness on the most promising parts of the space.

7 Model Construction and Evaluation

A small but growing number of landscape analysis papers [2,7,20] have successfully shown that landscapes contain meaningful information that is linked to

Table 1: Selected features with respect to sampler and budget.

ILS_{strict}				$ILS_{neutral}$				Feature	Description
5s	10s	20s	30s	5s	10s	20s	30s		
✓	✓	✓	✓	✓	✓	✓	✓	Cu	Number of Curricula
✓	✓	✓	✓	✓	✓	✓	✓	$\overline{Fit}_{\mathrm{A}}$	Mean fitness of Group A
✓	✓	✓	✓		✓	✓	✓	$\overline{Fit}_{\mathrm{B}}$	Mean fitness of Group B
✓	✓	✓	✓	✓	✓	✓	✓	$\overline{Fit}$	Mean fitness
					✓			$\mathrm{Sink}_{\mathrm{B}}$	Number of sink nodes in Group B
					✓			CC	Number of connected components

search algorithm performance. We employ a model building process consisting of feature selection followed by linear regression to predict the fitness value. The objective will then be to assess how the sampler and its budget affect the quality of the prediction for different solvers.

Pre-processing. After merging all data, there is a total of 120 features, some of which may not be useful. We first remove 21 features with a constant value: most of them are 90th percentile features, i.e., they describe the top of the landscape. Features are then standardized. After these two steps, features have to be selected in order to improve the potential success of our models.

We use correlation preselection which computes the correlation value between the outcome variable and features. Here, the outcome variable corresponds to the fitness value, which is the result of one of the three solvers used. This step selects all the features correlated with fitness above a fixed threshold that we set to a relatively high value of 0.9. Table 1 summarizes the selected features depending on the version of the ILS sampler and the allotted budget. Between 3 and 6 features are selected, with the number of curricula, the mean fitness across all sampled timeout plateaus, and the mean fitness of groups A and B always being present. These last two features are not only about fitness but also indirectly encode some information about the proportion of plateaus since this is used to create the groups. It is interesting to note the relative homogeneity of the selected features across samplers and budgets, as well as the absence of more complex features.

One potential caveat of this restricted feature set relates to the different fitness features and the associated multicollinearity that is not usually recommended for linear regression. Multicollinearity makes it difficult to interpret regression coefficients, however in this work we are essentially interested in the models' predictions and their precision, so this is not a concerning problem.

Evaluation. In order to obtain a robust evaluation of the models, especially given that we have few instances, we use complete 5-fold cross-validation.

Cross-validation uses complementary subsets of the data for training and testing in order to assess the model's ability to predict unseen data. With a k-fold

approach, data are partitioned into k subsets, one of which is retained for testing, the remaining $k-1$ being used for training the model. The cross-validation process is repeated k times such that each subset is used once for testing. The results are then averaged to produce a robust estimation. The specificity of the complete cross validation is to apply a k-fold on all possible cutting configurations [5]. Therefore $\binom{m}{m/k}$ configurations are considered instead of only k. In our case, with 19 instances and 5 folds, we have $\binom{19}{4} = 3876$ configurations. This complete 5-fold cross-validation algorithm 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 the 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 boosting effect.

The quality of the regression is assessed using the coefficient of determination, R^2, a well-known indicator for regression tasks.

8 Model Results and Discussion

Using the data collected and the selected features, we build linear regression models for each of the 2 ILS samplers, $\text{ILS}_{\text{neutral}}$ and $\text{ILS}_{\text{strict}}$, across 4 different time budgets to predict the performance of the 3 solvers considered, HLS, or $\text{HLS}_{\text{strict}}$ and SA. Since we observed that the $\overline{Fit}_{\text{A}}$ feature was likely to have a major impact on the model, we also build a set of trivial models that incorporate this single feature and another set of models that exclude this feature. There are therefore 72 different models in total. The resulting $\mathbf{R^2}$ values are plotted on Figure 4 where each line represents a (sampler, solver) pair.

The first observation is that in all cases we obtain relatively good to very good models, with $\mathbf{R^2}$ values ranging from 0.62 to 0.97. As was expected, $\overline{Fit}_{\text{A}}$ by itself is a very good predictor of the final fitness obtained by the different solvers. Nonetheless, models that use all the selected features come out on top, even if the advantage is somewhat marginal, indicating that it is worth using the extra features. Models that do not use $\overline{Fit}_{\text{A}}$ perform less well but remain competitive. The outliers to this general trend are the models built using $\text{ILS}_{\text{neutral}}$ and a 5 second budget per sampling run, where removing $\overline{Fit}_{\text{A}}$ causes a major decrease in $\mathbf{R^2}$. In that specific scenario, however, the sampling did not reach the mean fitness of the solvers and so $\overline{Fit}_{\text{A}}$ is a non-trivial feature.

If we consider neutral versus strict acceptance sampling, $\text{ILS}_{\text{neutral}}$ is always better except in the scenario mentioned before. This is to be expected since the landscape is known to be neutral. What is more surprising, is that strict acceptance holds up nonetheless and provides decent models.

Next, we consider what happens w.r.t. the different solvers. The performance of all 3 solvers is adequately predicted, even though the sampling algorithm differs from the solvers, as has been observed in the literature [20,3]. The prediction for SA is slightly worse since it is the most different from the sampling algorithm,

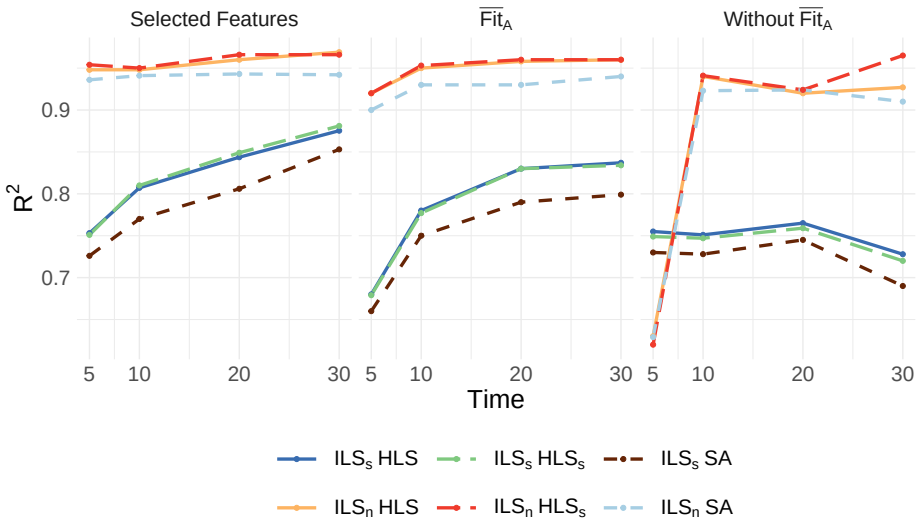

Fig. 4: R^2 values for models, where each point represents a (sampler, solver) pair.

whereas ILS is a component within HLS. Interestingly, there is no major difference between HLS with neutral (default version) and strict acceptance.

A general observable trend that holds for most cases, is that R^2 improves as the sampling budget increases, which seems fairly intuitive. Perhaps surprisingly however, predictive performance remains almost flat (but very good) when all the selected features and neutral sampling are used. This robustness with respect to time makes it easy to recommend using the smallest budget in that scenario.

9 Conclusion

In this paper, we considered the Curriculum-based Course Timetabling problem known to be a very neutral problem where a large number of solutions share the same fitness value. We proposed to characterized the search landscape taking into account the specificity of neutrality and used relevant metrics to build predictive models of solver performance. We compared two ILS samplers, $\text{ILS}_{\text{strict}}$ and $\text{ILS}_{\text{neutral}}$, based on hill-climbers that differ in their acceptance criterion. We showed that the sampler that considers the neutral specificity of the search space leads to better predictive models and is more time-robust.

In future work, we intend to consider different types of solvers, for example evolutionary algorithms, in order to study further the notion of neutrality and how it relates to predicting the performance of the solving algorithm. Moreover, we plan to investigate other university timetabling problems to observe the similarities and differences, and especially to check whether the same models can be used elsewhere.

References

1. Bischl, B., Mersmann, O., Trautmann, H., Preuß, M.: Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation. p. 313–320. GECCO '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2330163.2330209
2. Daolio, F., Liefooghe, 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). https://doi.org/10.1162/EVCO_a_00193, publisher: Massachusetts Institute of Technology Press (MIT Press)
3. Feutrier, T., Kessaci, M.E., Veerapen, N.: Exploiting landscape features for fitness prediction in university timetabling. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. GECCO '22, Association for Computing Machinery, New York, NY, USA (Jul 2022, [accepted as poster paper])
4. Jankovic, A., Doerr, C.: Landscape-aware fixed-budget performance regression and algorithm selection for modular cma-es variants. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference. p. 841–849. GECCO '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3377930.3390183
5. 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)
6. Lewis, R.: A survey of metaheuristic-based techniques for University Timetabling problems. OR Spectrum **30**(1), 167–190 (Jan 2008). https://doi.org/10.1007/s00291-007-0097-0
7. Liefooghe, 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). https://doi.org/10.1109/TEVC.2019.2940828
8. 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). https://doi.org/https://doi.org/10.1016/j.ins.2013.04.015
9. Malan, K.M., Engelbrecht, A.P.: Particle swarm optimisation failure prediction based on fitness landscape characteristics. In: 2014 IEEE Symposium on Swarm Intelligence. pp. 1–9 (2014). https://doi.org/10.1109/SIS.2014.7011789
10. Malan, K.M.: A survey of advances in landscape analysis for optimisation. Algorithms **14**(2) (2021). https://doi.org/10.3390/a14020040
11. 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.) Parallel Problem Solving from Nature - PPSN XII. pp. 226–235. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
12. Müller, T.: ITC2007 solver description: a hybrid approach. Annals of Operations Research **172**(1), 429–446 (Nov 2009). https://doi.org/10.1007/s10479-009-0644-y
13. Ochoa, G., Qu, R., Burke, E.K.: Analyzing the landscape of a graph based hyper-heuristic for timetabling problems. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2009) (2009)

14. 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). https://doi.org/10.1007/978-3-319-55453-2_16
15. 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, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-642-41888-4_9
16. PATAT: International Timetabling Competition 2007 (2007), publication Title: International Timetabling Competition
17. Pillay, N.: A review of hyper-heuristics for educational timetabling. Annals of Operations Research **239**(1), 3–38 (Apr 2016). https://doi.org/10.1007/s10479-014-1688-1
18. Salwani, A.: On the use of multi neighbourhood structures within a Tabu-based memetic approach to university timetabling problems - ScienceDirect (2012)
19. Stadler, P.F.: Fitness landscapes. In: Lässig, M., Valleriani, A. (eds.) Biological Evolution and Statistical Physics, pp. 183–204. Lecture Notes in Physics, Springer, Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-45692-9_10
20. Thomson, S.L., Ochoa, G., Verel, S., Veerapen, N.: Inferring future landscapes: Sampling the local optima level. Evolutionary Computation **28**(4), 621–641 (2020). https://doi.org/10.1162/evco_a_00271
21. Tomassini, M., Verel, S., Ochoa, G.: Complex-network analysis of combinatorial spaces: The nk landscape case. Phys. Rev. E **78**, 066114 (Dec 2008). https://doi.org/10.1103/PhysRevE.78.066114
22. Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of nk landscapes with neutrality. IEEE Transactions on Evolutionary Computation **15**(6), 783–797 (2011). https://doi.org/10.1109/TEVC.2010.2046175