Analyzing the Landscape of a Graph Based Hyper-heuristic for Timetabling Problems

Gabriela Ochoa, Rong Qu and Edmund K. Burke
Automated Scheduling, Optimisation and Planning (ASAP) Group
School of Computer Science
University of Nottingham, UK
{gxo, rxq, ekb}@cs.nott.ac.uk

ABSTRACT

Hyper-heuristics can be thought of as "heuristics to choose heuristics". They are concerned with adaptively finding solution methods, rather than directly producing a solution for the particular problem at hand. Hence, an important feature of hyper-heuristics is that they operate on a search space of heuristics rather than directly on a search space of problem solutions. A motivating aim is to build systems which are fundamentally more generic than is possible today. Understanding the structure of these heuristic search spaces is therefore, a research direction worth exploring. In this paper, we use the notion of fitness landscapes in the context of constructive hyper-heuristics. We conduct a landscape analysis on a heuristic search space conformed by sequences of graph coloring heuristics for timetabling. Our study reveals that these landscapes have a high level of neutrality and positional bias. Furthermore, although rugged, they have the encouraging feature of a globally convex or *big valley* structure, which indicates that an optimal solution would not be isolated but surrounded by many local minima. We suggest that using search methodologies that explicitly exploit these features may enhance the performance of constructive hyper-heuristics.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-Numerical Algorithms and Problems.

General Terms

Algorithms, Measurement, Performance.

Keywords

Hyper-heuristics, landscape analysis, graph coloring heuristics, timetabling.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

1. INTRODUCTION

Metaheuristics have been widely and successfully applied to a wide variety of computational search problems. However, significant development effort is often needed to produce fine tuned techniques for the particular problem (or even instance) at hand. A more recent research trend in search methodologies, particularly in timetabling and scheduling, is the study of *hyper-heuristics* [2, 19]. Hyper-heuristics aim at producing more general problem solving techniques, which can potentially be applied to different problems (or instances) with little development effort. An important feature of hyper-heuristics is that they operate on a search space of heuristics rather than directly on a search space of solutions. The problem of searching a good combination of heuristics is itself a combinatorial optimization problem. Understanding the structure of these heuristic search spaces, is therefore, the goal of this paper. Specifically, we use the notion of fitness landscapes for analyzing hyper-heuristics, and conduct a landscape analysis of the heuristic search space induced by the graph-based hyper-heuristic presented in [4, 16] for the educational timetabling problem. This approach operates upon a set of widely used constructive heuristics (graph coloring heuristics) in timetabling. A candidate solution in the heuristic search space is given by a sequence (list) of graph coloring heuristics. Each heuristic in this list is successively used to (re)-order the events (exams or courses in educational timetabling) and schedule them accordingly. To the best of our knowledge, this is the first time that the fitness landscape metaphor is used in the context of hyper-heuristic research. However, previous work has highlighted the existence of the two search spaces in constructive hyper-heuristics. In [4], the relationship between the search space of heuristics and the solution space of the problem is graphically illustrated. The authors observed that a large number of heuristic sequences generate unfeasible solutions in the problem space. This work is extended in [16] to construct a unified graph based hyper-heuristic framework (GHH), where the neighborhood structures and characteristics of the two search spaces are analyzed. Moreover, efficient hybridizations in GHH with local search operating on the solution space are investigated. Another related study [24] in the context of production scheduling, presented a formal definition of the heuristic search space and introduced the notion of a *decision block* to refer to a set of decisions that are treated as a single unit (i.e. processed by a single heuristic).

The paper is organized as follows. Section 2 introduces hyper-heuristics and describes in more details the hyper-heuristic under study. Section 3 presents the notion of fitness landscapes and describes how to conduct an autocorrelation and a fitness distance correlation analysis. Thereafter, section 4 describes the use of the notion of fitness landscapes for studying constructive hyper-heuristics. Section 5 describes in detail the methodology followed and reports the empirical results obtained. Finally, Section 6 summarizes our findings and suggests further work.

2. HYPER-HEURISTICS

Hyper-heuristics are heuristics that choose heuristics in order to solve a given combinatorial optimization problem. We study a hyper-heuristic framework where there is a high-level heuristic and a set of low-level heuristics. Given a problem instance, the high-level heuristic selects which low-level heuristic should be applied at each decision point in the problem solving process [2]. The motivation behind hyper-heuristics is the observation that different heuristics have different strengths and weaknesses. Therefore, it may be fruitful to combine them adaptively so that each makes up for the weaknesses of another [19]. The key idea is to use members of a set of known and reasonably understood heuristics to either: (i) transform the state of a problem (in a constructive strategy), or (ii) perform an improvement step (in a perturbative strategy). This last distinction leads us to a classification of hyper-heuristic approaches into *constructive* and *perturbative*. Constructive hyper-heuristics build a solution incrementally. Starting with an empty solution, they intelligently select and use constructive heuristics to gradually build a complete solution. They have been successfully applied to several combinatorial optimization problems such as: bin-packing [20], timetabling [5, 4, 16, 17, 23], production scheduling [24], and cutting stock [22]. On the other hand, improvement or perturbative search hyper-heuristics find a reasonable initial solution, either randomly or using a simple constructive heuristics, and then select heuristics to shift and swap solution components, with the aim of finding improved solutions. In other words, they start from a complete solution and then search for heuristics which will select among a set of neighborhoods, for better solutions. Perturbative or improvement hyper-heuristics have been applied to personnel scheduling [7, 3], timetabling [3], shelf space allocation [1], packing [9]and vehicle routing problems [15].

Graph-based hyper-heuristics for timetabling problems. The hyper-heuristic selected for our study is based on graph coloring constructive heuristics, and uses a Tabu search algorithm as the high level search strategy. Timetabling problems can be modeled as graph coloring problems, where nodes in the graph represent events, and edges represent conflicts between events (e.g. exams). Graph heuristics in timetabling use the information in the graph to order the events by their difficulties (e.g. number of conflicts with other events), and assign them one by one into the time-slots. These degrees indicate how difficult the events are to be assigned; therefore, the most difficult event will be assigned first by the corresponding ordering strategy. The GHH implemented the following 5 graph coloring-based heuristics, plus a random ordering heuristic:

Largest Degree (LD): orders the events decreasingly by the number of conflicts (events with common students involved) the event has with the others.

Largest Weighted Degree (LWD): the same as LD but weighting the events by the number of students involved.

Color Degree (CD): orders the events decreasingly in terms of the number of conflicts that they have with those already scheduled in the timetable.

Largest Enrolment (LE): order the events decreasingly by the number of enrols the event has.

Saturation Degree (SD): order the events increasingly by the number of time-slots available in the timetable for the event at that time.

A candidate solution in the heuristic search space corresponds to a sequence (list) of these heuristics. The solution (in the problem space) construction is an iterative process where, at the i^{th} iteration, the i^{th} graph-coloring heuristic in the list is used to order the events not yet scheduled at that step, and the 1^{st} e events in the ordered list are assigned to the first e least-cost timeslots in the timetable. Tabu Search was employed as the high-level search strategy for producing good sequences of the low-level heuristics. Each heuristic list produced by the Tabu Search algorithm is evaluated by sequentially using the individual heuristics to order the unscheduled events, and thus construct a complete timetable. As mentioned before, each heuristic in the list is used to schedule a number e of events. Therefore, the length of the heuristic list is n/e where n is the number of events to be scheduled. Values in the range of $e = 1, \dots, 5$ were tested. For the landscape analysis reported here, we set $e = 1$, that is a single event is scheduled by each heuristic in the list. Hence the number of events to be scheduled, n corresponds to the length of the heuristic list. Furthermore, two low-level graph coloring heuristics, namely, *least Saturation Degree first* (SD) and *Largest Weighted Degree first* (LWD), were considered because this combination was found the most successful at solving the educational timetabling instances explored in [4]

3. FITNESS LANDSCAPE ANALYSIS

The fitness landscape metaphor can be used for search in general. Given a search problem, the set of possible solutions can be coded using strings of fixed length from some finite alphabet. This encoding generates a representation space, which is a high dimensional space of all possible strings of a given length. There is also a neighborhood relation that defines which points in the representation space are connected. This relation depends on the specific search operator or combination of operators, used to search the space. Finally, there is a fitness function that assigns a fitness value to each possible string or point in the space.

More formally [14], a fitness landscape (S, f, d) of a problem instance of a given combinatorial optimization problem consists of a set of candidate solutions S, a fitness (or evaluation) function $f : S \mapsto R$, which assigns a real-valued fitness to each solution in S, and a distance metric d that defines the spatial structure of the landscape. This distance is related to the neighborhood relation described above. For binary encoded problems of length n (which is the case of the heuristic search space studied here), the search space is $S = \{0, 1\}^n$. The distance measure is the Hamming distance between bit strings. The minimum distance between two points in the search space is 1 (one bit with a different value), and the maximum distance, also known as the landscape diameter, is n.

In the context of meta-heuristics, it is important to identify the features of landscapes that would influence the effectiveness of heuristic search. Such knowledge may be helpful for both predicting the performance and improving the design of meta-heuristics. Statistical methods have been proposed for measuring fitness landscape properties. Two of the most commonly used landscape analysis techniques: fitness distance correlation and auto-correlation analysis, are described in detail below.

3.1 Fitness distance correlation analysis

The most commonly used measure to estimate the global structure of fitness landscapes is the *fitness distance correlation (FDC)* coefficient, proposed by Jones and Forrest [12]. It is used as a measure for problem difficulty in genetic algorithms. Given a set of points $x_1, x_2, \ldots x_m$ and their fitness values, the FDC coefficient ϱ is defined as:

$$\varrho(f, d_{opt}) = \frac{Cov(f, d_{opt})}{\sigma(f)\sigma(d_{opt})} \tag{1}$$

where $Cov(.,.)$ denotes the covariance of two random variables and $\sigma(.)$ the standard deviation. The FDC determines how closely related are the fitness of a set of points and their distances to the nearest optimum in the search space (denoted by d_{opt}). A value of $\varrho = -1.0$ ($\varrho = 1.0$) for maximisation (minimisation) problems indicates a perfect correlation between fitness and distance to the optimum, and thus predicts an easy search. On the other hand, a value of $\varrho = 1.0$ ($\varrho = -1.0$), means that with increasing fitness the distance to the optimum increases too, which indicates a deceptive and difficult problem. As suggested in [12], a value of $fdc \leq -0.5$ ($fdc \geq 0.5$) for maximisation (minimisation) problems indicates an easy problem.

Often, a fitness distance plot is made to gain insight into the structure of the landscape, in addition to (or instead of) calculating the correlation coefficient [14]. This is done by plotting the fitness of points in the search space against their distance to an optimum or best-known solution. This type of analysis can be used to investigate not only the correlation between arbitrary points in the search space, but also the distribution of local optima within the search space.

3.2 Autocorrelation analysis

An important characteristic of a landscape is its *ruggedness*, which is related to the difficulty of an optimization problem for heuristic algorithms. Weinberger [25] introduced a procedure to measure the correlation structure of a fitness landscape based on the *autocorrelation function* [11]. The idea is to generate a random walk, of size T, on the landscape via neighboring points. At each step, the fitness of the solution encountered is recorded, and thus, a time series of fitness values is generated. Thereafter, the autocorrelation function of the time series, ρ_i is calculated. The theoretical autocorrelation function, ρ_i, can be empirically estimated by r_i:

$$r_i = \frac{\sum_{t=1}^{T-i}(f_t - \overline{f})(f_{t+i} - \overline{f})}{\sum_{t=1}^{T}(f_t - \overline{f})^2}, \tag{2}$$

where $\overline{f}$ is the mean fitness of the T points visited, and i is the time lag or distance between points in the walk. A related measure is the *correlation length* of a fitness landscape. Several authors have proposed approaches to measure this quantity [11, 13, 25]. Statistically [11], the correlation length gives an indication of the largest "distance" (or time lag) between two points at which the value of one point can still provide information about the expected value of the other point. In other words, the correlation length is the largest time lag i for which one can still expect some correlation between two points i steps apart. We use here the correlation length measure based on the estimated autocorrelation function (Equation 2) [25]:

$$\ell = \frac{-1}{ln(|r_1|)}, \tag{3}$$

for $r_1 \neq 0$ (where r_1 is defined according to Eq. 2). The correlation length reflects the ruggedness of a landscape. The smaller the value for ℓ, the more rugged the landscape. The correlation length typically depends on the instance size [21]; therefore, it is often reported in relation to the landscape diameter n.

4. HYPER-HEURISTIC LANDSCAPES

In hyper-heuristics we deal with two search spaces: (i) the search spaces of heuristics, and (ii) the optimization problem solution space. However, we have a single landscape, as the objective function of a point in the heuristic search space, can only be known after calculating the objective value of the correspondent point in the problem solution space.

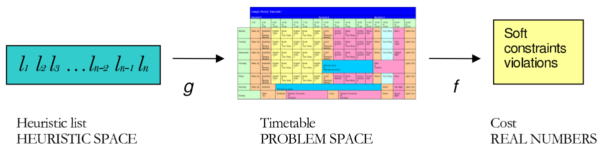

Figure 1: Constructive hyper-heuristic landscapes can be viewed as compositions $f(g(l))$, where g : Heuristic Space $\mapsto$ Problem Space represents the constructive process, and f : Problem Space $\mapsto \Re$ encodes the evaluation function of constructed solutions.

More formally, let m be the number of low-level heuristics in a constructive hyper-heuristic approach, and let n be the number of events to be scheduled in the underlying (timetabling or scheduling) problem. The heuristic search space, HS is, therefore, composed of strings of length n, in the alphabet $\{h_1, h_2, \ldots, h_m\}$ [1]. The size of HS is m^n, as any of the m low-level heuristic can be selected for scheduling each of the n events. Let PS be the problem space, and f the evaluation function that maps a point in PS to a real number, $f : PS \mapsto \Re$. Let g be a function that maps a sequence of heuristics s in HS to its corresponding solution in PS. Figure 1 illustrates the two search spaces discussed and the two mappings involved, namely, g from a heuristic list to its corresponding problem solution, and f from the solution to the objective value. Actually, g corresponds to a constructive solution process that, starting from an empty solution, successively applies the heuristic in the sequence s (following its order) to construct a complete solution. In consequence, the evaluation function of a heuristic sequence s in HS is given by the composition of g and f, namely,

[1]We consider here that $m = 2$ and that each heuristic in the heuristic list is used to schedule one event ($e = 1$). But different values of these parameters may be considered, producing different landscapes.

$f(g(s)) : HS \mapsto R$. The landscape of a constructive hyper-heuristic is, therefore, defined as the triplet $(HS, f(g), d)$, where d is the distance (in this case the Hamming distance), HS the heuristic space, and $f(g)$ the fitness function.

5. ANALYSIS OF THE GHH LANDSCAPE

We performed both a fitness distance correlation analysis (on a set of empirically generated local optima) and an autocorrelation analysis on random walks, for measuring respectively, the global and local properties of the GHH landscape. We also suggest some visualizations of the obtained local optima, that bring about an image of some features of the heuristic landscape. As mentioned above (section 2), two graph coloring heuristics, namely, *least Saturation Degree first* (SD) and *Largest Weighted Degree first* (LWD), were considered for our analysis. Therefore, the heuristic search space studied consists of n-dimensional 0-1 (binary) vectors. Where n is the number of events to be scheduled in the underlying timetabling instance (and thus also the length of the heuristic list), and 0 and 1 encode the graph heuristics SD and LWD, respectively. For measuring the distance between solutions, we used the standard Hamming distance, which counts the number of bits in which two solutions differ.

The GHH was tested on the widely used set of benchmark instances, the Toronto set, originally presented in [6], and further discussed in [18]. The size of these instances ranges from 81 to 682 exams and from 611 to 18419 students. The density of the conflict matrix, which gives the ratio of the number of conflicting exams over the overall number of exams, ranges from 0.06 to 0.42. For our hyper-heuristic landscape analysis, we selected a subset of these instances for illustrative purposes. We have conducted preliminary experiments in the remaining instances that show similar tendencies. The selected instances are described in Table 1. A detailed discussion of the complete set of instances can be found in [18].

Instance	hec92 I	sta83 I	ute92 I	ear83 I
Exams	81	139	184	190
Students	2823	611	2750	1125
Time-slots	18	13	10	24
Matrix density	0.42	0.14	0.08	0.27

Table 1: Characteristics of the benchmark exam timetabling instances selected for the analysis.

In our formulation, the hard constraints are given by the "conflicts" of scheduling two exams with common students into the same time-slot. Whereas the soft constraint is concerned with spreading out the students' exams over the timetable so that students will not have to sit exams that are too close to each other. The objective is to schedule all the exams into the time-slots, while minimizing the soft constraint violations per student. Thus, the objective function $C(t)$ calculating the cost of violations within a solution t is as follows:

$$C(t) = \frac{\sum_{s=0}^{4} w_s \times N_s}{S},$$

where, $w_s = 2^s$, $s \in \{0, 1, 2, 3, 4\}$ represents the importance of scheduling exams with common students either 4, 3, 2, 1, or 0 timeslots away in timetable t, N_s, $s \in \{0, 1, 2, 3, 4\}$ is the number of students involved in the violation of the soft constraints. The lower the cost, $C(t)$, the better the timetable. Infeasible solutions (i.e. those with violations of hard constraints) are discarded.

5.1 Fitness distance correlation analysis

FDC analysis requires knowledge of the optimal solution. However, given that the optimal solution is not generally known, many studies in the literature use the best-known solution instead. For a given instance, let x^* be the best-known solution on the heuristic search space. In order to have a wide distribution of distances to the optimum, a fixed number of feasible solutions (10 in our experiments) were randomly generated at each distance i (for $i = 1, \ldots, n$) away from x^*, where n is the length of the heuristic list. This set of points were thereafter used as starting points for generating a set of local optima. Thus, the set of generated local optima is of size $10 \times (n-1)$. The local optima were produced by a non-deterministic next-descent local search using the 1-move neighborhood (detailed in Figure 2). This procedure accepts solutions of the same objective value. Therefore, this method is similar to Davis's [8] *bit-climbing* scheme, in which the bits are mutated in a prefixed random order, and the current-best is reset to any string having equal or better cost value than the previous best evaluation.

```
Procedure Local-Search(s, n)
π ← permutation of n integers
repeat
  s' ← invert next bit in s (according to order π)
  if C(s') ≤ C(s) then
    s ← s'
  end if
until all bits in π tested, or n equal cost solutions accepted
```

Figure 2: Pseudo-code for the algorithm producing the local optima. s represents the current solution (initially received as a parameter), and n the problem size.

5.2 Visualizing local optima

This section illustrates some global features of the GHH landscape by visualizing the sets of local optima found on two selected timetabling instances: **hec92 I** and **sta83 I**. The remaining two instances show similar trends, and are not shown due to space constraints. Figure 3 illustrates the objective values (costs) of all the empirically found local optima, ordered from the best found at the origin of each plot (**hec92** I (top) and **sta83 I** (bottom)). Whereas, Figure 4 illustrates the local optima themselves (the top 10 %), again ordered according to their cost values from the best found at the top of each plot (**hec92 I** (top) and **sta83 I** (bottom)). Each horizontal line (pattern of white and black dots) in the plots accounts for the binary string encoding a single local optima. A white dot pictures a '1' (LWD heuristic) and a black dot a '0' (SD heuristic). Figure 3 shows that there are several local optima with the same cost value (plateaus). This feature is most clearly seen on **sta83 I**, which shows several plateaus at different cost levels.

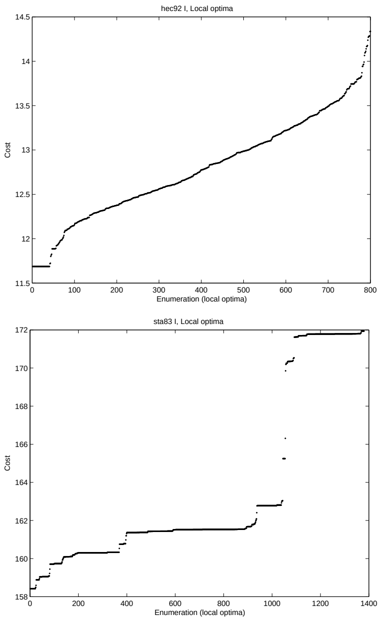

Figure 3: The complete set of empirically found local optima for two selected instances: hec92 I (top) and sta83 I (bottom), ordered from the for each instance, in ascending order of their costs. The y-axis shows the cost values, while the x-axis simply enumerates the local optima.

Figure 4 illustrates that some positions in the heuristic list are fixed for the top local optima. For example, the first four positions of most top 10% local optima in **hec92 I** (top plot) are white dots (i.e. LWD heuristics), whereas the first two positions in **sta83 I** (bottom plot) are black dots (SDs). Other patterns of fixed bits can be observed. Notice that a more randomized pattern is found towards the right-most positions. This is consistent with the intuition and previous observation that the choices of heuristics towards the end of a constructive process have less impact on the overall performance. A more randomized pattern can also be observed towards the lower quality solutions (bottom parts of each plot), suggesting that many different low quality local optima can be found in the landscape. Another interesting observation is that the balance between the proportion of SD and LWD heuristics in the local optima is not the same across all the instances. For example, there is a higher proportion of SDs (black dots) in the local optima of **sta83 I** (bottom plot, fig. 4), whereas a more balanced proportion of SDs and LWDs is observed in **hec92 I** (upper plot, fig. 4).

5.3 Local optima measures and cost-distance scatter plots

This section reports the empirical measurements taken from the set of generated local optima in each instance. Table 2 summarises, for each instance:

n**:** the landscape diameter, which corresponds to the heuristic list length and the underlying problem size (number of events to schedule).

$Samples$**:** the number of generated local optima, which is

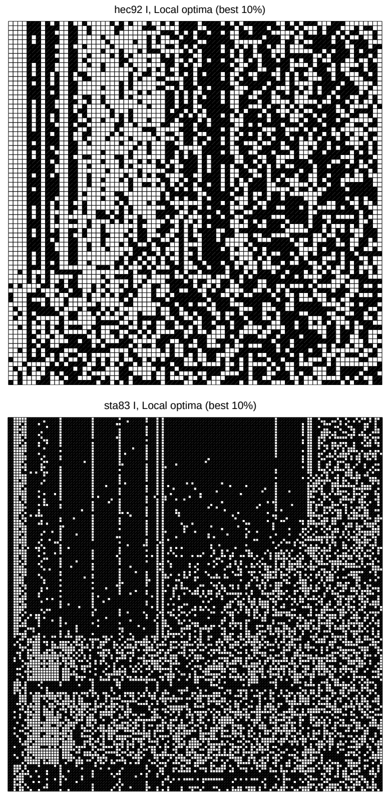

Figure 4: Top 10 % local optima for two selected instances: hec92 I (top) and sta83 I (bottom) ordered from the best at the top to the worst. Each horizontal line accounts for the binary string encoding a single local optima. A white dot pictures a '1' (LWD heuristic) and a black dot a '0' (SD heuristic).

$10 \times (n-1)$.

fdc**:** the fitness distance correlation coefficient.

$Cost$**:** the minimum, maximum, average, and standard deviation of cost values.

Hd_{opt} the minimum, maximum, average, and standard deviation of the Hamming distances to the best-known solution.

St_{impr}**:** the minimum, maximum, average, and standard deviation of the number of improvement[2] steps in the local search process (from random initial solutions to local optima).

For all studied instances, there is a moderate to high positive correlation between the cost of local optima and their distances to the global optimum (in the range of 0.51 to 0.64). The presence of such a correlation implies that the lower the cost the closer the local optima are to the global optimum (or best-known solution). It also suggests a *big valley* structure of the landscape, which means that on average, local optima are very much closer to the optimum than

[2]This measure sums the moves that improve the current solution cost. Steps of equal costs are not counted.

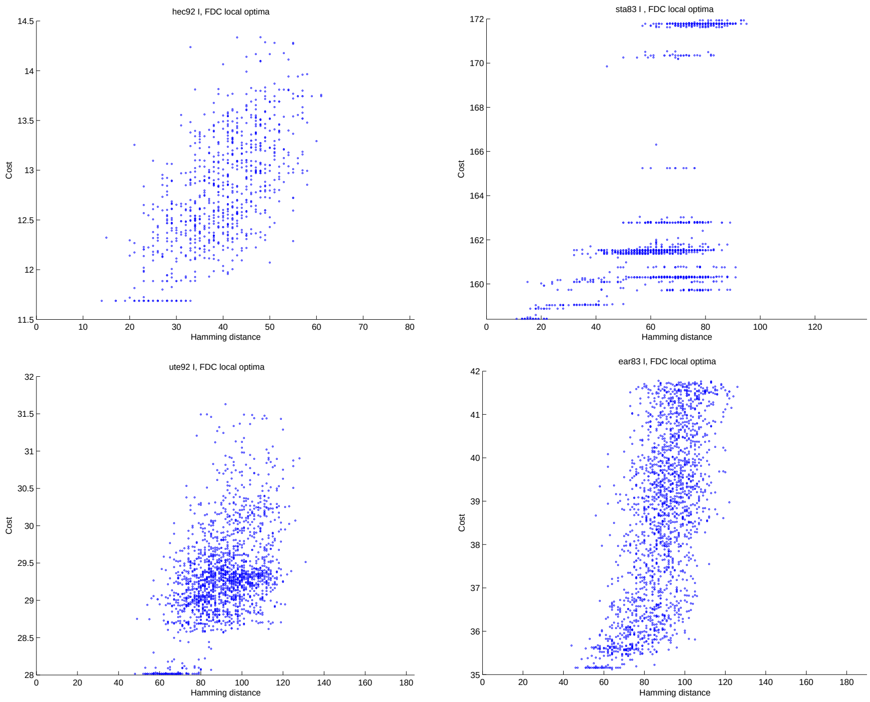

Figure 5: Cost-distance correlation analysis of local optima of the GHH landscape studied instances.

Instance	**hec92 I**	**sta83 I**	**ute92 I**	**ear83 I**
n	81	139	184	190
$Samples$	800	1380	1830	1890
fdc	0.64	0.51	0.51	0.63
$minCost$	11.69	158.42	28.02	35.16
$maxCost$	14.34	171.94	31.63	41.77
$avgCost$	12.79	163.56	29.31	38.60
$stdCost$	0.577	4.580	0.627	1.944
$minHd_{opt}$	14	11	48	44
$maxHd_{opt}$	61	95	131	126
$avgHd_{opt}$	39.66	64.97	90.37	88.61
$stdHd_{opt}$	8.884	15.577	14.823	13.840
$minSt_{impr}$	0	0	0	0
$maxSt_{impr}$	10	15	32	29
$avgSt_{impr}$	2.45	4.07	11.81	8.29
$stdSt_{impr}$	1.929	2.144	5.737	4.346

Table 2: GHH landscape local optima measurements on the four studied instances.

are randomly chosen points. In addition to the *fdc* coefficients, cost distance scatter plots (Figure 5) provide useful information about the landscape. Notice that the plots for instances **hec92 I** and **ute92 I** have similar features: the costs of local optima and distances to the optimum show a clear positive correlation, and the best local optima are concentrated on a bounded region of the search space. The scatter plot for instance **sta83 I** shows a different overall picture. A positive correlation between the cost of local optima and their distances to the optimum still holds, and the best local optima are close to the best-known solution. However, the distribution of local optima is different showing several wide plateaus, as many local optima having the same cost can be visualised at different levels. Some values in the range of local optima costs are not covered by any point. The cost-distance plot for instance **ear83 I** shows again a positive correlation, but only on the best local optima; that is, those with a cost value below 38. There are, in this case, a large number of low-quality local optima that show no clear cost-distance correlation and are randomly located in the search space (around $n/2$ bits away from the global optimum). In all the instances, the scatter plots confirm that there are several (different) local optima having the same cost as the best known solution, so there is a set of optimal solutions (instead of a single optimum) located in this plateau.

In terms of the number of downside (or improvement) steps for reaching the local optima, Table 2 indicates that this value is highly variable in all instances (has a high standard deviation with respect to its average). For all studied instances, this value ranges from 0 (i.e the starting point is already a local optima), up to about $n/8$. Moreover, its average value is rather low as compared to n, suggesting that the landscapes have many shallow valleys. The number of sidewards steps (moves with equal objective value) on the search trajectories are not reported as they always reached the maximum allowed (n), which suggests the presence of neutrality on these landscapes.

5.4 Auto-correlation analysis

The local structure of the hyper-heuristic landscapes was analyzed through the random walk correlation function (section 3.2). A single random walk of size $T = 1000$ was conducted on each instance, and equation 2 was used to calculate r_i, for $i = 1, \ldots, n$. Table 3 reports both the correlation length, ℓ, calculated according to equation 3, and the corre-

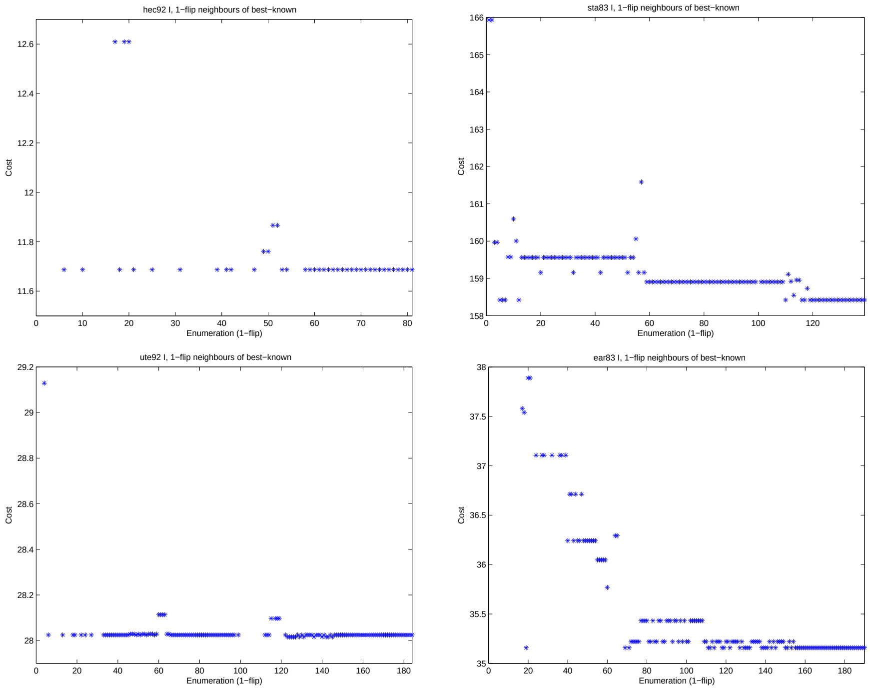

Figure 6: Visualising the cost values of the 1-flip neighborhood of the best-known solution for each instance. The neighbors that produce unfeasible solutions are not pictured.

lation length in relation to the diameter of the landscape n/ℓ. Two of the studied instances have a highly rugged landscape as reflected by low correlation length (n/ℓ of about 6 and 8, see Table 3). Whereas the other two landscapes are found to be smoother, with a higher relative correlation length as expressed by a lower n/ℓ (close to 3). The smoothest landscape corresponds to instance **sta83 I** with $n/\ell = 2.79$, which is consistent with the large plateaus observed for this instance.

Instance	**hec92 I**	**sta83 I**	**ute92 I**	**ear83 I**
n	81	139	184	190
ℓ	13.36	49.82	51.77	22.87
n/ℓ	6.06	2.79	3.55	8.30
pbf	0.444	0.201	0.065	0.311
puf	0.469	0.007	0.217	0.168

Table 3: GHH landscape correlation length. And measurements on the best-known solution plateau: plateau branching factor, pbf and plateau unfeasible factor, puf.

We also calculated the *plateau branching factor* [10] (pbf in Table 3), of the best-known solution in each landscape. This measure is defined as the fraction of direct neighbors of a solution l that are in the same plateau as l. Since the underlying timetabling problems in our study contain unfeasible solutions, we also calculated the *plateau unfeasible factor*, which we define here as the fraction of direct neighbors of a solution l that leads to unfeasible solutions (puf in Table 3). As discussed in [10], the efficacy of exploration and escape mechanisms can be affected by plateau branching. Our results show that the best-known solution in every instance is located in a plateau with a branching factor that is greater than 0. The specific pbf value is instance dependant and can be as high as almost half the diameter of the landscape. In order to study the order-dependence of the neutral and unfeasible direct neighbors, Figure 6 illustrates, for each instance, the cost values of all the 1-flip neighbours of the best-known solution. The unfeasible neighbors are not shown: i.e. an empty space appears in the corresponding position. The plots in Figure 6, strikingly show two features of these heuristics landscapes. First, the landscapes are highly rugged, in that small differences in the heuristic lists (1-flips) make a huge difference in the solution's cost, even shifting a best-known solution to an unfeasible solution. Second, there is a strong positional bias on the heuristic search spaces, where changes on the list's left-most positions have a much higher impact on the evaluation function as compared to the right-most positions. Indeed, the very last positions are neutral in that they do not produce changes in the cost value.

6. CONCLUSIONS

We have used the notion of fitness landscapes for studying constructive hyper-heuristics, and conducted a landscape analysis on the heuristic search space induced by the approach presented in [4, 16] for educational timetabling. This hyper-heuristic operates upon a set of constructive heuristics (graph coloring heuristics), widely used in modeling timetabling problems. The prominent features of the studied landscapes can be summarized as follows:

Big valley structure: the cost of local optima and their distances to the global optimum (best-known solution) are correlated, which suggests that these landscapes have a globally convex or big valley global structure. The best local optima are located in a relatively small region of the search space. There are, however, many low-quality local optima that are distant from the best-known solution. Also, the size and characteristics of the region of the search space that holds the local optima, seems to be instance dependant.

Large number of local optima: the landscapes contain a large number of distinct local optima, many of them of low quality.

High ruggedness: the landscape relative correlation length depends on the underlying instance. The correlation lengths vary from moderate to high. However, a detailed study of the plateaus containing the best-known solution, shows that small changes in a good solution can produce much worse or even unfeasible solutions.

Existence of wide plateaus (neutrality): many local optima are located at the same level (height) in the search space, that is, they have the same cost value.

Shallow valleys: the number of improvement steps from random points towards local optima, are on average low in comparison with the landscape diameter, which suggests that the landscapes have many shallow valleys.

Positional bias: left-most positions in the heuristic list have a much greater impact on the produced solution cost.

It should be noted that this study represents the first analysis of a heuristic search space composed of sequence of constructive heuristic. Work is in progress to analyze a related landscape composed of sequences of dispatching rules for production scheduling. We suggest that search algorithms that explicitly exploit the features described above, will enhance the search on this type of heuristic search spaces. These performance predictions should be tested in future work. Moreover, similar and more advanced landscape analysis techniques should be conducted on both larger set of instances and different application domains.

7. REFERENCES

[1] R. Bai, E. K. Burke, and G. Kendall, *Heuristic,meta-heuristic and hyper-heuristic approaches for fresh produce inventory control and shelf space allocation*, Journal of the Operational Research Society **59** (2008), 1387 – 1397.

[2] E. K. Burke, E. Hart, G. Kendall, J. Newall, P. Ross, and S. Schulenburg, *Hyper-heuristics: An emerging direction in modern search technology*, Handbook of Metaheuristics (F. Glover and G. Kochenberger, eds.), Kluwer, 2003, pp. 457–474.

[3] E. K. Burke, G. Kendall, and E. Soubeiga, *A tabu-search hyperheuristic for timetabling and rostering*, Journal of Heuristics **9** (2003), no. 6, 451–470.

[4] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, *A graph-based hyper-heuristic for educational timetabling problems*, European Journal of Operational Research **176** (2007), 177–192.

[5] E. K. Burke, S. Petrovic, and R. Qu, *Case based heuristic selection for timetabling problems*, Journal of Scheduling **9** (2006), no. 2, 115–132.

[6] M. W. Carter and G. Laporte, *Recent developments in practical course timetabling*, PATAT (Edmund K. Burke and Michael W. Carter, eds.), Lecture Notes in Computer Science, vol. 1408, Springer, 1997, pp. 3–19.

[7] P. Cowling, G. Kendall, and E. Soubeiga, *A hyperheuristic approach to scheduling a sales summit*, Selected papers from the 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT 2001), Springer, Aug 2001.

[8] L. Davis, *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York, 1991.

[9] K. A. Dowsland, E. Soubeiga, and E. K. Burke, *A simulated annealing hyper-heuristic for determining shipper sizes*, European Journal of Operational Research **179** (2007), no. 3, 759–774.

[10] H. H. Hoos and T. Stützle, *Stochastic local search: Foundations and applications*, Morgan Kaufmann / Elsevier, 2004.

[11] W. Hordijk, *A measure of landscapes*, Evolutionary Computation **4** (1997), no. 4, 335–360.

[12] T. Jones, *Crossover, macromutation, and population-based search*, Proceedings of the Sixth International Conference on Genetic Algorithms (San Francisco, CA) (L. J. Eshelman, ed.), Morgan Kaufmann, 1995, pp. 73–80.

[13] B. Manderick, M. de Weger, and P. Spiessens, *The genetic algorithm and the structure of the fitness landscape*, Proceedings of the 4th International Conference on Genetic Algorithms, Morgan Kaufmann, 1991, pp. 143–150.

[14] P. Merz and B.Freisleben, *Fitness landscapes, memetic algorithms, and greedy operators for graph bipartitiioning*, Evolutionary Computation **8** (2000), no. 1, 61–91.

[15] D. Pisinger and S. Ropke, *A general heuristic for vehicle routing problems*, Computers and Operations Research **34** (2007), 2403– 2435.

[16] R. Qu and E.K. Burke, *Hybridisations within a graph based hyper-heuristic framework for university timetabling problems*, Journal of the Operational Research Society **t.a.** (2008), t.a, to appear, doi: 10.1057/jors.2008.102.

[17] R. Qu, E.K. Burke, and B.McCollum, *Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems*, European Journal of Operational Research **t.a.** (2008), t.a, online publication, doi:10.1016/j.ejor.2008.10.001.

[18] R. Qu, E.K. Burke, B.McCollum, L.T. Merlot, and S.Y. Lee, *A survey of search methodologies and automated system development for examination timetabling*, Journal of Scheduling **t.a.** (2008), t.a., to appear, doi: 10.1007/s10951-008-0077-5.

[19] P. Ross, *Hyper-heuristics*, Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques (E. K. Burke and G. Kendall, eds.), Springer, 2005, pp. 529–556.

[20] P. Ross, J. G. Marin-Blazquez, S. Schulenburg, and E. Hart, *Learning a procedure that can solve hard bin-packing problems: A new ga-based approach to hyper-heuristics*, Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2003, Springer, 2003, pp. 1295–1306.

[21] P. F. Stadler, *Towards a theory of landscapes*, Complex Systems and Binary Networks (R. López-Pena, R. Capovilla, R. García-Pelayo, H. Waelbroeck, and F. Zertuche, eds.), Lecture Notes in Physics, vol. 461, Springer-Verlag, Berlin, Heidelberg, New York, 1995, pp. 77–163.

[22] H. Terashima-Marin, A. Moran-Saavedra, and P. Ross, *Forming hyper-heuristics with GAs when solving 2D-regular cutting stock problems*, Proceedings of the 2005 IEEE Congress on Evolutionary Computation (Edinburgh, Scotland, UK), vol. 2, IEEE Press, 2-5 September 2005, pp. 1104–1110.

[23] H. Terashima-Marin, P. Ross, and M. Valenzuela-Rendon, *Evolution of constraint satisfaction strategies in examination timetabling*, Proc. of the Genetic and Evolutionary Computation Conf. GECCO-99 (San Francisco, CA), Morgan Kaufmann, 1999, pp. 635–642.

[24] J. A. Vázquez-Rodríguez, S. Petrovic, and A. Salhi, *A combined meta-heuristic with hyper-heuristic approach to the scheduling of the hybrid flow shop with sequence dependent setup times and uniform machines*, Proceedings of the 3rd Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA 2007), 2007.

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