# Investigating the Landscape of a Hybrid Local Search Approach for a Timetabling Problem

Thomas Feutrier
thomas.feutrier@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

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

## ABSTRACT

Curriculum-Based Course Timetabling is an NP-hard problem that can be efficiently solved by metaheuristics. The International Timetabling Competition (ITC) 2007 was won by a hybrid local search (HLS) combining Hill Climbing, Great Deluge and Simulated Annealing. HLS remains one of the best local search algorithms to solve this problem. In this paper, we investigate the search landscape of 21 instances to analyze the behavior of the HLS components. We also propose a new distance metric that aims to be more robust and be less influenced by symmetry. Experiments show that the HLS and the embedded simulated annealing have the same general behavior but HLS leads to better robustness. This analysis strongly suggests that the HLS components and/or parameter values should be automatically configured to further improve performance.

## CCS CONCEPTS

• **Computing methodologies → Discrete space search**.

## KEYWORDS

University timetabling, landscape analysis, local search

**ACM Reference Format:**
Thomas Feutrier, Marie-Éléonore Kessaci, and Nadarajen Veerapen. 2021. Investigating the Landscape of a Hybrid Local Search Approach for a Timetabling Problem. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449726.3463175

## 1 INTRODUCTION

University Timetabling is an active research area of NP-hard optimization problems. Curriculum-Based Course Timetabling is one variant of university timetabling where students belong to one curriculum. Since 2002, the International Timetabling Competition makes algorithms compete on various timetabling problems. Curriculum-Based Course Timetabling is one of the three problems tackled by the 2007 event. The winner algorithm is a hybrid local search (HLS) designed by Müller [7]. It iterates sequentially between two intensification algorithms, a Hill Climbing (HC) and a Simulated Annealing (SA) with a cooling schedule, and a Great Deluge (GD) algorithm to diversify the search.

The particularity of Curriculum-Based Course Timetabling instances is the numerous equivalent timetables that can disturb the neighborhood exploration. The author chose to take into account this particularity in the design of HLS since the hill climbing and the simulated annealing accept fitness-equivalent solutions. However, the choice of using both hill climbing and simulated annealing is not explained by the author and we wonder if using simulated annealing only would lead to same performance.

We chose to study HLS in particular because it is a local search with good performance within a short running time and produces results that remain competitive to this day, especially for a comparatively simple and generic hybrid method.

In this paper, we propose to analyze HLS and its algorithmic components through a search landscape analysis. We propose a new distance metric that we compare to two distance metrics of the literature [2]. Our analysis shows that while HLS performance is influenced by its components, the SA mechanism within is very powerful by itself. We examine neutrality both at a global level and at the search landscape level. The first one concerns only optima produced by the HC procedure before GD, while the second deals with all solutions explored during HLS process. We observe a high level of neutrality at the latter level but almost none at the former.

The paper is organized as follows. Section 2 presents Curriculum-Based Course Timetabling and its neighborhoods. Then, HLS and its embedded algorithmic components are described in Section 3. The search landscape is presented in Section 4 and the distance metrics are explained. Section 5 gives the experimental protocol to conduct the search landscape analysis and Section 6 presents and analyzes the experimental results. The last section gives the conclusion and discusses the perspectives.

## 2 CURRICULUM-BASED COURSE TIMETABLING

Timetabling problems represent a class of NP-hard combinatorial optimization problems where, basically, resources have to be assigned to events [3]. In the educational context, events are lectures (classes) and resources are teachers, rooms and other specific equipment like video projector. Timetabling problems are highly studied by the Operations Research field, and several international timetabling competitions (ITC) over years have been organized [11] to encourage the design of algorithms and to evaluate their performance. In this paper, we focus on a particular university timetabling called Curriculum-Based Course Timetabling (CB-CTT). It is one of the timetabling problems tackled in the ITC 2007 competition [10].


*GECCO '21 Companion, July 10–14, 2021, Lille, France*

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463175

## 2.1 Definition

The main difference with the general university timetabling problem is the implementation of the curriculum system. Indeed, a student belongs to a curriculum, corresponding to several courses, and a course implies a specific number of lectures. Here, a course is given by one teacher only, but there is no limit on the number of courses given by each teacher. A course can be included in two, or more, different curricula. In that case students of each curriculum will attend together the lectures of such a course.

A timetable is divided into days, themselves divided into timeslots or periods. The number of timeslots per day is fixed by the problem instance. For example for a semester, there are about 100 days. If the parameter *Number of periods per day* is 6, there are 600 timeslots for one semester. A solution consists in scheduling the lectures in the timeslots and available rooms following hard and soft constraints, detailed below.

## 2.2 Constraints

Within CB-CTT, a timetable has to respect four constraints to be considered as feasible. These *hard constraints* may be described as follows:

**Conflicts:** Two lectures from the same course or from the same curriculum or with the same teacher cannot take place at the same time.
**Room occupancy:** Only one lecture can take place at a time in a room.
**Availability:** If a teacher is not available, their lectures cannot take place.
**Lectures:** All lectures must be scheduled.

While hard constraints need to be obeyed, *soft constraints* may be violated and only indicate some preferred outcome. The fewer the violations, the better the timetable. This is the basis for the objective function that drives the optimization process.

Four soft constraints were considered by ITC 2007 [10]. These are listed below.

(1) **Room Capacity:** A maximum number of students can be sat in the room.
(2) **MinWorkingDays:** A course has lectures which should be scheduled within a minimum number of days in order to avoid students having all lectures of one course over two days, that would be inconvenient.
(3) **Curriculum Compactness:** Within each curriculum, lectures on the same day should be consecutive, more precisely if there are more than one lecture of one curriculum on the same day, each needs to immediately precede or follow another.
(4) **Room stability:** Lectures of a course should be in the same room.

## 2.3 Objective function

The Curriculum-Based Course Timetabling Problem is a minimization problem. The objective function to optimize is a weighted sum of the constraint violations, as detailed in the following expression.

$$
\begin{aligned}
Cost(s) = & RoomCapacity(s) * \omega_{rc} & (1)\\
& + MinWorkingDays(s) * \omega_{mw} & (2)\\
& + CurriculumCompactness(s) * \omega_{cc} & (3)\\
& + RoomStability(s) * \omega_{rs} & (4)
\end{aligned}
$$

In the cost function, $s$ is a solution and *RoomCapacity(s)* represents the number of RoomCapacity constraint violations in the timetable represented by *s*. In the context of ITC 2007, and for our work, the weights $\omega_{rc}$, $\omega_{mw}$, $\omega_{cc}$ and $\omega_{rs}$ are set to 1, 5, 2 and 1 respectively.

## 2.4 Neighborhood

Local search algorithms start from an initial solution and then iteratively move from one solution to another in order to explore the search space. This is based on a neighborhood relation. The HLS method, described Section 3, uses six different neighborhoods.

The neighborhoods considered are listed below. Some of them may be known under different names in the literature. In that case we give the alternative names.

**Lecture Move or Time Move:** A lecture and a timeslot are selected. The lecture is assigned to the timeslot.
**Room Move:** A lecture and a room are selected. The lecture is assigned to the room.
**Lecture Room Move or Lecture Swap:** A lecture, a room and a timeslot are selected. The lecture is assigned to the room and timeslot. If the room is already used on this timeslot, the timeslot and room of the selected lecture are swapped with those of the conflicting lecture.
**Room Stability Move:** A course and a room are selected, then each lecture of course is assigned to the room. If the room is not available, the lectures of the course swap rooms with the conflicting lectures.
**MinWorkingDays Move:** A course with a MinWorkingDays violation is selected. Lectures for that course, on days with more than one lecture, are moved to another timeslot to minimize MinWorkingDays violation. The room can be changed if it is already used by other lecture.
**CurriculumCompactness Move:** A curriculum with Curriculum-Compactness penalty is selected. An isolated lecture within the curriculum is assigned to a different timeslot so that it becomes adjacent to other curriculum lectures. The room can be changed if not available.

# 3 HYBRID LOCAL SEARCH

This paper focuses on the winning approach for ITC 2007 that was proposed by Müller [7]. It is a hybrid method combining three search algorithms.

In HLS, an initial solution is first constructed using Iterative Forward Search (IFS) [6], then Hill Climbing (HC), Great Deluge (GD) [4] and Simulated Annealing (SA) [5] are applied in sequence. The result from SA is then fed back to HC. HLS stops when a solution with a fitness value of 0 is found or when the maximum run time is reached. This process is illustrated in Figure 1.

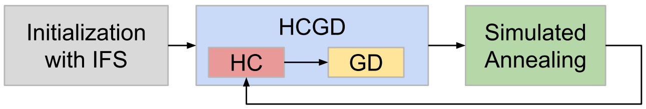


**Figure 1: HLS Method, IFS corresponds to Iterative Forward Search, HC to Hill climbing and GD to Great Deluge.**

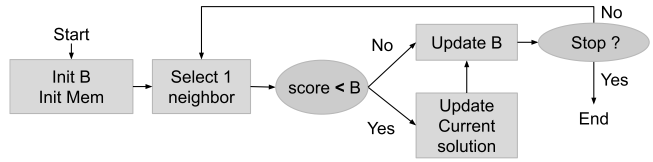


**Figure 2: Great Deluge algorithm**

IFS is the Iterative Forward Search proposed by Müller. It consists in assigning resources to lectures in no specific order, then if a resource cannot be assigned to a lecture, conflicting lectures are unscheduled to free this resource. Therefore, the considered resource is scheduled and the unscheduled ones return in the queue. This method runs until all the timetable is scheduled.

Algorithms HC, GD and SA share the same neighborhoods defined previously in Section 2.4. During the neighborhood exploration, one of them is selected at random. Actually, the selection probability is not completely uniform since the *CurriculumCompactness* neighborhood has a lower probability of being selected (only 0.1 the likelihood of the others). The author explains that this neighborhood is the most computationally expensive. In order to fairly balance the allocated time among neighborhoods, its selection probability is lowered.

When two rooms, sharing the same properties, are available to schedule a lecture, the resulting timetables have the same fitness value and so, are equivalent. Müller [7] chose to take this into account this CB-CTT specificity in the design of HLS directly. Therefore, a solution is accepted not only when it strictly improves upon the current solution but also if it has the same fitness. The search space therefore includes neutral neighbors. This applies to all three search algorithms. Moreover exploration follows a first improvement strategy.

## 3.1 Hill Climbing

The HC embedded in HLS resembles the netcrawling algorithm proposed by Barnett [1] since equivalent neighbors are accepted in the neighborhood exploration. However, one stopping criterion, corresponding to the number of iterations without finding a new best global solution, has been added to avoid unsuccessful explorations of plateaus, i.e., where the accepted solutions are fitness-equivalent. This criterion is controlled by parameter `n`, fixed to 50,000 in HLS.

## 3.2 Great Deluge

GD acts as a perturbation method within HLS. Its aim is to move to a different region of the search space once HC fails to find improving solutions. While three of its parameters are static and are used to compute the lower bound, the upper bound and the cooling rate at the beginning of the run, two further parameters are dynamic. The first one, denoted `B`, is initialized at the beginning of GD and updated during the loop to control the strength of the perturbation. The second parameter, denoted `Mem` (named `at` in the original paper), memorizes the number of iterations of HLS without improvement and sets bounds for the perturbation strength to allow more or less diversification in order to escape the current local optimum.

Figure 2 illustrates the GD algorithm. For each application of GD, parameters `B` and `Mem` are initialized. Then, while `B` is over a threshold, a neighbor is randomly selected in the neighborhood of the current solution, a score is computed and depending on its value relative to `B`, it replaces the current solution and/or `B` is updated.

## 3.3 Simulated Annealing

SA is applied after GD and uses the same strategy as HC: a first improvement neighbor selection accepting neutral solutions. The method features classic simulated annealing parameters: cooldown factor, initial temperature and final temperature. SA uses the classic cooling mechanism. Like HC, SA stops after a number of iterations without finding a new best global solution that depends on instance features. Details can be found in Müller [7].

# 4 SEARCH LANDSCAPES

Neighborhood relations between solutions induce a network structure where vertices are solutions and edges are neighborhood relations. A *fitness* or *search landscape* is a commonly-used metaphor to describe such a structure. Solution fitness is considered as the height of the landscape, thus forming surfaces that can be smooth or rugged, and that may contain ridges and plateaus. These topological features influence the difficulty of the search. Naturally, the metaphor has its limits given that any non-trivial landscape is actually highly multidimensional.

Search heuristics, such as the local search components in HLS, drive the search across the landscape looking for the best solution. In addition to fitness, solutions can be compared using the notion of distance. This enables quantifying the similarity between solutions.

## 4.1 Definition and Properties

*Definition 4.1 (Landscape).* A landscape may be formally defined [12] as a triplet $(S, N, f)$ where

- $S$ is a set of potential solutions, i.e. a 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 (the *height* in the landscape).

*Definition 4.2 (Local optimum).* A local optimum is a solution $s^* \in S$ such that $\forall s \in N(s^*)$, $f(s^*) \leq f(s)$. The inequality is not strict, to allow neutral landscapes. In this definition a local minimum is considered, without loss of generality.

In the context of HLS, the Hill Climbing component includes a time-out condition that does not guarantee that the solution obtained is a true local optimum because in some cases the neighborhood of the obtained solution will not have been completely explored. For this reason we call the solutions *pseudo-local optima*.

*Definition 4.3 (Basin of attraction).* The basin of attraction of some local optimum $s_i^* \in S$, according to a hill climbing operator $h$, is the set $b_i = \{s \in S | h(s) = s_i^*\}$.

*Definition 4.4 (Plateau).* A plateau is a set of connected solutions that share the same fitness value. Two vertices in a plateau are connected if they are neighbors with the same fitness.

As later detailed in Section 5, we observe pseudo-local optima in order to sample a landscape (though currently only the fitness is recorded). This therefore closely resembles Local Optima Networks (LON) [9] that model the global structure of landscapes as graphs. In these structures, nodes are local optima and edges represent possible transitions induced by one or more operators, usually a perturbation followed by a local search.

Plateaus may exist at the LON or pseudo-LON level. They may be actual search landscape plateaus or connected basins of attraction. We call the plateaus we observe with our sampling of HLS, *pseudo-LON plateaus*. Plateaus at the LON level have been referred to as meta-plateaus [13].

## 4.2 Distance Computation

In this study three distances are used. The first two come from the literature [2] and we propose the third one. These metrics are influenced by the solution output format, which contains lectures, their courses, rooms, and the day and timeslot. There is no information about the curriculum. The following metrics do not take into account room information and focus on timeslot and course information. This choice is motivated by the fact that, in real life, changing a room for another one is often less important than the timeslot because there are lot of similar room (by their capacity).

In order to be more explicit about the link between distances and neighborhoods, we present each metric below and list the possible distance values between neighbors. Room and RoomStability moves are not concerned because rooms are not taken into account in the distance computations.

*4.2.1 Day and Timeslot Distance.* This metric, or $\text{Dist}_{\text{DT}}$ for short, represents the most precise distance among the three. This is because, for each lecture of each course, if the lecture does not have the same timeslot and the same day, distance increases by a single unit. The goal of this metric is to compare if two timetables are really the same within rooms.

$\text{Dist}_{\text{DT}}$ between two neighbors may take the following integer values depending on the neighborhood:

- Lecture Move: 1
- Lecture Swap: 1 or 2
- MinWorkingDays Move: 1 to $\text{NLectures}_{\text{course}}$-1
- CurriculumCompactness Move: 1 to $\text{NLectures}_{\text{curriculum}}$-1

*4.2.2 Timeslot Only Distance.* This metric, or $\text{Dist}_{\text{TO}}$, is computed as follows: for each lecture of each course, the day the lecture is scheduled does not matter, only the difference in time of the day matters. A one hour difference implies a distance of one unit. As an example, if two timetables only differ by some lecture being scheduled on Thursday 15 at 5pm in one timetable and on Monday 25 but also at 5pm in the other, they are considered the same. Thus timetables that only differ by some permutation of days will be considered equivalent.

$\text{Dist}_{\text{TO}}$ between two neighbors may take the following values:

- Lecture Move: 0 or 1
- Lecture Swap: 0, 1 or 2
- MinWorkingDays Move: 0 to $\text{NLectures}_{\text{course}}$-1
- CurriculumCompactness Move: 0 to $\text{NLectures}_{\text{curriculum}}$-1

*4.2.3 Working Period Distance.* We propose the *Working Period* metric, or $\text{Dist}_{\text{WP}}$, in order to have a different point of comparison between solutions, focused more on periods instead of dates.

$\text{Dist}_{\text{WP}}$ is calculated as follows: for each course, the period between the first day and the last day a course is taught is computed. Distance is increased by the gap between the two corresponding working periods.

This metric is inspired by the Min Working Days soft constraint. The aim is to compare the distribution of each course and consider the global dynamics of timetables. In fact, the focus is not on *when* but *how long*. Neighborhood operators all have the same range of effect on this distance, except for room-linked operators. Indeed the working period depends on only two lectures for each course so one rescheduling can drastically extend or reduce the working period. For example if a course is taught on the first day, and if one of its lectures is moved on the last day (Lecture Move operator), the distance is NTimeslots. $\text{Dist}_{\text{WP}}$ may thus take values between 0 and NTimeslots for the Lecture Move operator.

# 5 EXPERIMENTAL PROTOCOL

HLS obtained the best performance on the instances proposed by ITC 2007 for the CB-CTT problem and accordingly won the competition. Yet, HLS is composed of three search components, HC, GD and SA presented in Section 3. We propose to analyze the behavior of HC and GD when run together (hereafter called HCGD), SA and HLS in order to better understand the merits of each one. HC by itself is of lesser interest because it is purely focused on intensification and its performance is not as good.

We use the source code provided by Müller [7], with the recommended parameter settings and the 21 instances of ITC 2007. In these instances, the number of courses belongs to the domain $[\![30, 131]\!]$, the number of lectures to $[\![160, 390]\!]$, curricula $[\![13, 150]\!]$, timeslots $[\![25, 45]\!]$ and rooms $[\![5, 20]\!]$.

We instrumented the original HLS code in order to be able to record the fitness of each accepted solution during the HC and SA steps. We also recorded the final solution obtained after each complete run of HLS. Unfortunately recording the complete solution representation for all solutions is currently prohibitively expensive memory-wise, but finding an efficient way around this is an interesting avenue for future research.

Initially, 1,000 runs of HCGD, SA and HLS were executed on Intel Xeon Silver 4114 CPUs @ 2.20GHz with a run time of 300 seconds each, for a total time budget of almost 220 days. The returned solution of each run was recorded and used to analyze distances $\text{Dist}_{\text{DT}}$, $\text{Dist}_{\text{TO}}$ and $\text{Dist}_{\text{WP}}$ with respect to the best known solution. The fitness after each HC was also recorded to provide a somewhat global picture. Recording the fitness of all accepted solutions within each run would be prohibitively expensive to store, potentially taking up 7 TB of space. In a second phase of the experiments (for Section 6.3), only 100 additional runs were carried out (time budget of almost 22 days), this time recording each accepted solution fitness during the HC and SA steps, thus providing a trace of each run.

# 6 SEARCH LANDSCAPE ANALYSIS

In this Section, we compare the performance between HLS, HCGD and SA, we discuss the distance metrics and analyze the plateaus to attempt to understand the structure of the landscape.

## 6.1 Performance Assessment

HLS is composed of three search algorithms, each presumably having some impact on the efficiency of the overall approach. To investigate the contribution of SA, we disabled GD and HC. For HCGD (Hill Climbing and Great Deluge), we disabled only the SA part of the code. This allows the comparison between HCGD, SA and HLS.

To support the performance analysis, Table 1 provides summary statistics of the fitness values for the solutions returned by each of the three methods. For each instance, the best algorithm(s), according to the Friedman statistical significance test, is indicated in bold. Instances are classified into 4 groups, as explained hereafter.

SA seems to outperform HLS and HCGD for most instances. Indeed, SA is the best on 12 out of 21 instances, HLS on 7 and HCGD on 4. To investigate this observation further, we propose to rank the algorithms for each instance. Table 1 shows the rank of each algorithm, as given by the Friedman Test for each instance. The sum of all ranks per method is then computed.

Contrary to expectations based on the previous observation, HLS beats SA by a single point when the ranks are summed. In fact Table 1 shows that HLS is only ranked first or second (and never third), contrary to SA which underperforms at third place on a number of instances.

We use the ranks as a means to classify the instances into 4 groups (A, B, C, and D). Group A corresponds to the two very easy instances where the ranks do not actually mean anything much, as ascertained from the distribution of fitness values. Instances where HCGD comes third are in Group B and those where SA comes last in Group C. The remaining two instances are in Group D. These groups will help the analysis in following sections.

Results reported on Table 1 show that SA works well for most of instances while some of them remain quite difficult. These latter instances are in groups A and C where HCGD gets the best performance. HLS performance is fairly consistent on each instance, and that may explain why Müller exploited the advantages of both HCGD and SA in sequence to win ITC 2007 [7]. These results give interesting hints for future investigation to analyze and understand how each algorithmic component could be optimally configured and potentially triggered depending on instance or landscape characteristics.

## 6.2 Analysis with the Distance Metrics

The distance metrics $\text{Dist}_{\text{DT}}$ and $\text{Dist}_{\text{TO}}$ are commonly used for CB-CTT [2]. We compute the correlation between these distances and our proposal $\text{Dist}_{\text{WP}}$ using all solutions returned by the different methods. $\text{Dist}_{\text{DT}}$ and $\text{Dist}_{\text{TO}}$ are highly correlated with a value of 0.9, and each one is correlated to $\text{Dist}_{\text{WP}}$ with a value of 0.82 and 0.8 respectively. These high values indicate that they give similar information about similarities between solutions.

Figure 3 presents scatter plots for HCGD (blue points), SA (green points) and HLS (red points) and the three distance metrics $\text{Dist}_{\text{DT}}$ (left), $\text{Dist}_{\text{TO}}$ (middle) and $\text{Dist}_{\text{WP}}$ (right). Following the groups defined in Section 6.1, only one instance per group is presented. Instances 01, 06, 12 and 13 were chosen because the shape of the scatter plots is representative of their own group, featuring the same general behavior. The x-axis represents the distance between the 1,000 optima and the best known solution and the y-axis their fitness. Clearly, the scatter plots of the three distance metrics show the same general shape for each single instance. However, the values of the distance metrics are different. $\text{Dist}_{\text{WP}}$ gives the smallest values while $\text{Dist}_{\text{DT}}$ the highest one. $\text{Dist}_{\text{WP}}$ implies the most narrow scatter plots regardless of the instance. That could be interpreted as a proof of the robustness of this metric. Indeed, we proposed $\text{Dist}_{\text{WP}}$ in order to be less sensitive to what can be considered as symmetric solutions, such as solutions exhibiting month shift or the exchange of all timeslots between two courses. This will be useful for our future work.

For instances in Groups A, B and D, the values of the three distance metrics are very different and almost do not overlap each other. However, for instances of group C, $\text{Dist}_{\text{TO}}$ and $\text{Dist}_{\text{WP}}$ share identical values even if $\text{Dist}_{\text{WP}}$ values are lower.

For Groups B, C and D, the scatter plots do not show an observable fitness-distance correlation for any Distance metric. In particular, it means that the solutions found by HCGD, SA and HLS may be close in quality to the best known solution even if they can be very different. This may suggest a shallow valley structure [8] because it seems that the bottom of the basin of attraction of the best solution is fairly large and composed of plateaus since many solutions are equivalent, sharing the same fitness value.

## 6.3 Neutrality Analysis

Neutrality, or the lack thereof, can be observed at different levels. To achieve some insight into what can be considered the global level of the landscape, let us first consider the pseudo-local optima returned by each HC call when running HLS. These are recorded across 1,000 runs per instance in order to obtain as many fitness sequences.

These sequences can give us some insight into the nature of the pseudo-LON plateaus (Section 4.1). A pseudo-LON plateau may be part of an actual plateau.

We choose to observe the number of consecutive iterations that have the same fitness in order to gauge how easy it is for HLS to escape from a pseudo-LON plateau. For example, say we observe the sequence 15 12 12 11 10 5 5. We can count the plateaus: there are two of them, at fitness 12 and 5. And we can compute the fraction of iterations that form part of a plateau, here 4 out of 7 iterations. This is in fact the neutral degree, i.e., the percentage of neutral solutions divided by all the solutions considered.

The third column of Table 2 gives the mean neutral degree for HLS for each of our 21 instances. The numbers seem to indicate that, at the pseudo-LON level, there are actually not so many plateaus and that the Great Deluge perturbation is relatively successful in escaping from pseudo-LON plateaus. Note that further analysis shows that, when it succeeds, the escape mechanism followed by

**Table 1: Fitness values of the local optima found by HCGD, SA and HLS. For each distance, the minimum, the average, the median, the standard deviation and the maximum are reported. Bold values mean that the algorithm is statistically better than the others. The rank of each method is provided in the three last columns. The four groups are based on these rankings.**

| Gr | Inst | HCGD | | | | SA | | | | HLS | | | | Rank | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min | Mean$_{(sd)}$ | Med | Max | Min | Mean$_{(sd)}$ | Med | Max | Min | Mean$_{(sd)}$ | Med | Max | HCGD | SA | HLS |
| A | 01 | 5 | **5**$_{(0)}$ | 5 | 5 | 5 | $5.1_{(0.2)}$ | 5 | 7 | 5 | **5**$_{(0)}$ | 5 | 5 | 1 | 3 | 1 |
| | 11 | 0 | **0**$_{(0)}$ | 0 | 0 | 0 | $0_{(0)}$ | 0 | 1 | 0 | **0**$_{(0)}$ | 0 | 0 | 1 | 3 | 1 |
| B | 02 | 40 | $72.9_{(14.4)}$ | 71 | 138 | 39 | $61.6_{(8.3)}$ | 61 | 103 | 37 | **58.2**$_{(7.5)}$ | 58 | 80 | 3 | 2 | 1 |
| | 03 | 70 | $93_{(9.9)}$ | 92 | 128 | 68 | $91_{(8.2)}$ | 91 | 127 | 69 | **84.9**$_{(5.9)}$ | 85 | 102 | 3 | 2 | 1 |
| | 06 | 41 | $67_{(9.7)}$ | 66 | 99 | 35 | **51.7**$_{(5.4)}$ | 52 | 70 | 40 | $55.5_{(5.9)}$ | 55 | 77 | 3 | 1 | 2 |
| | 07 | 8 | $32.2_{(9.2)}$ | 32 | 62 | 6 | **15.4**$_{(3.4)}$ | 15 | 26 | 8 | $20.1_{(4.4)}$ | 20 | 36 | 3 | 1 | 2 |
| | 08 | 39 | $45.7_{(3.2)}$ | 45 | 59 | 37 | **42.8**$_{(2.5)}$ | 43 | 51 | 39 | $44.5_{(2.8)}$ | 45 | 54 | 3 | 1 | 2 |
| | 10 | 16 | $40.8_{(9.1)}$ | 40 | 75 | 8 | **18.9**$_{(4.8)}$ | 19 | 34 | 9 | $21.5_{(5.1)}$ | 21 | 37 | 3 | 1 | 2 |
| | 14 | 51 | $61.2_{(4)}$ | 61 | 79 | 51 | **59.9**$_{(3.6)}$ | 59 | 76 | 51 | $60.3_{(3.2)}$ | 60 | 72 | 3 | 1 | 2 |
| | 15 | 68 | $92.2_{(9.9)}$ | 91 | 130 | 69 | $90.3_{(8.2)}$ | 90 | 127 | 68 | **85.1**$_{(5.8)}$ | 85 | 106 | 3 | 2 | 1 |
| | 16 | 34 | $57.1_{(8)}$ | 57 | 82 | 26 | **38.8**$_{(4.9)}$ | 39 | 60 | 26 | $41.3_{(5.5)}$ | 41 | 58 | 3 | 1 | 2 |
| | 17 | 66 | $87.3_{(7.5)}$ | 87 | 114 | 66 | **82.1**$_{(5.2)}$ | 82 | 100 | 67 | $84_{(5.4)}$ | 84 | 99 | 3 | 1 | 2 |
| | 19 | 58 | $77.7_{(8.8)}$ | 77 | 112 | 59 | **70**$_{(4.9)}$ | 70 | 91 | 58 | **69.7**$_{(4.4)}$ | 69 | 84 | 3 | 1 | 1 |
| | 20 | 18 | $52.1_{(13.3)}$ | 51 | 93 | 16 | **31.5**$_{(6)}$ | 31 | 53 | 19 | $36.5_{(6.8)}$ | 36 | 64 | 3 | 1 | 2 |
| | 21 | 92 | $118.6_{(10.2)}$ | 118 | 156 | 88 | **106.6**$_{(6.5)}$ | 106 | 139 | 90 | $107.8_{(7.2)}$ | 108 | 132 | 3 | 1 | 2 |
| C | 05 | 286 | **331.9**$_{(13.6)}$ | 332 | 384 | 296 | $362.5_{(29.8)}$ | 358 | 521 | 300 | $335.9_{(14.1)}$ | 336 | 403 | 1 | 3 | 2 |
| | 09 | 97 | **109.2**$_{(4)}$ | 109 | 124 | 97 | $110.3_{(4.7)}$ | 110 | 125 | 99 | **109.3**$_{(3.7)}$ | 109 | 122 | 1 | 3 | 1 |
| | 12 | 306 | **345.4**$_{(9)}$ | 346 | 396 | 325 | $373.1_{(15.4)}$ | 374 | 417 | 322 | $350.7_{(8.8)}$ | 351 | 384 | 1 | 3 | 2 |
| | 18 | 69 | **85**$_{(3.8)}$ | 85 | 98 | 75 | $92.6_{(5.3)}$ | 93 | 109 | 74 | $87.2_{(3.4)}$ | 87 | 96 | 1 | 3 | 2 |
| D | 04 | 35 | $40.9_{(3.2)}$ | 41 | 52 | 35 | **39.8**$_{(2.8)}$ | 39 | 49 | 35 | $40.9_{(2.6)}$ | 41 | 51 | 2 | 1 | 2 |
| | 13 | 59 | $73.8_{(3.9)}$ | 74 | 84 | 59 | **73**$_{(4.7)}$ | 73 | 87 | 63 | $74_{(3.8)}$ | 74 | 84 | 2 | 1 | 2 |

SA and HC intensification phases does not necessarily reach a better solution.

Naturally, the low degree of neutrality at the pseudo-LON level does not mean that there is little neutrality at the search landscape level, as we will see shortly. Neutrality is a well-known property of timetabling problems and has been demonstrated via landscape analysis, for instance by Ochoa et al. [8]. This previous work shows there are multiple pseudo-LON plateaus that are not necessarily of the same fitness and this is what we are observing here, with Great Deluge usually being able not to fall back into the same pseudo-LON plateau (or at least the same fitness level).

Another landscape characteristic that can be obtained from the fitness sequences is auto-correlation, a proxy for ruggedness, i.e., the measure of how smooth or rugged a landscape is. Following the approach in [8], we computed the auto-correlation with a step of 1 for each sequence. Moreover we use the auto-correlation to get correlation length. This represents the largest time lag between two points where there is some correlation. The smaller the length, the more rugged the landscape.

The auto-correlation computation shows that the correlation length is quite low: below 2 for each instance. So we can conclude that the landscape at the pseudo-LON level is rugged. And the smoothest landscapes are for instance 05 and 12 with a length of about 2. A deeper study of instance characteristics might explain the ruggedness at this level.

Let us focus now on the search landscape level. Recall that we recorded the fitness of all accepted solutions during HC and SA phases across 100 runs in a second experiment phase. This allowed us to compute the neutral degree but this time at the level of accepted solutions within the run (the landscape level), and not only after each HC (the global level). Numbers are provided in Table 2.

At the landscape level, we can observe neutrality, and that there is a striking difference between HCGD and the two other methods. In fact the neutral degree for HCGD is almost 100% for all instances. That means HCGD spends most of its time on plateaus and that is probably the cause of its weakness. HLS also features a comparatively high neutral degree on most instances, whereas we observe the opposite for SA. Clearly, the neutral degree for HLS is influenced by the combination of HCGD and SA.

SA presents a neutral degree of around 50% in most of the cases. This could be explained by the fact that each time SA accepts a worse solution, a new improvement is found just after. However for instances of groups A and C (the most difficult for SA), the neutral degree is usually lower. That could mean that SA, on these instances, meets too many poor solutions and so accepts a lot of degradation in the quality of solutions, which then causes artificially many improving solutions (it is easy to improve once there has been a lot of degradation).

Table 3 shows the size of plateaus encountered during solving. Size represents the number of equal-fitness solutions successively observed until a new strictly improving solution. Table 3 displays means of medians and not actual medians due to how the data is recorded becase of its size. In fact there are too many plateaus to compute real medians in reasonable time, for one instance file

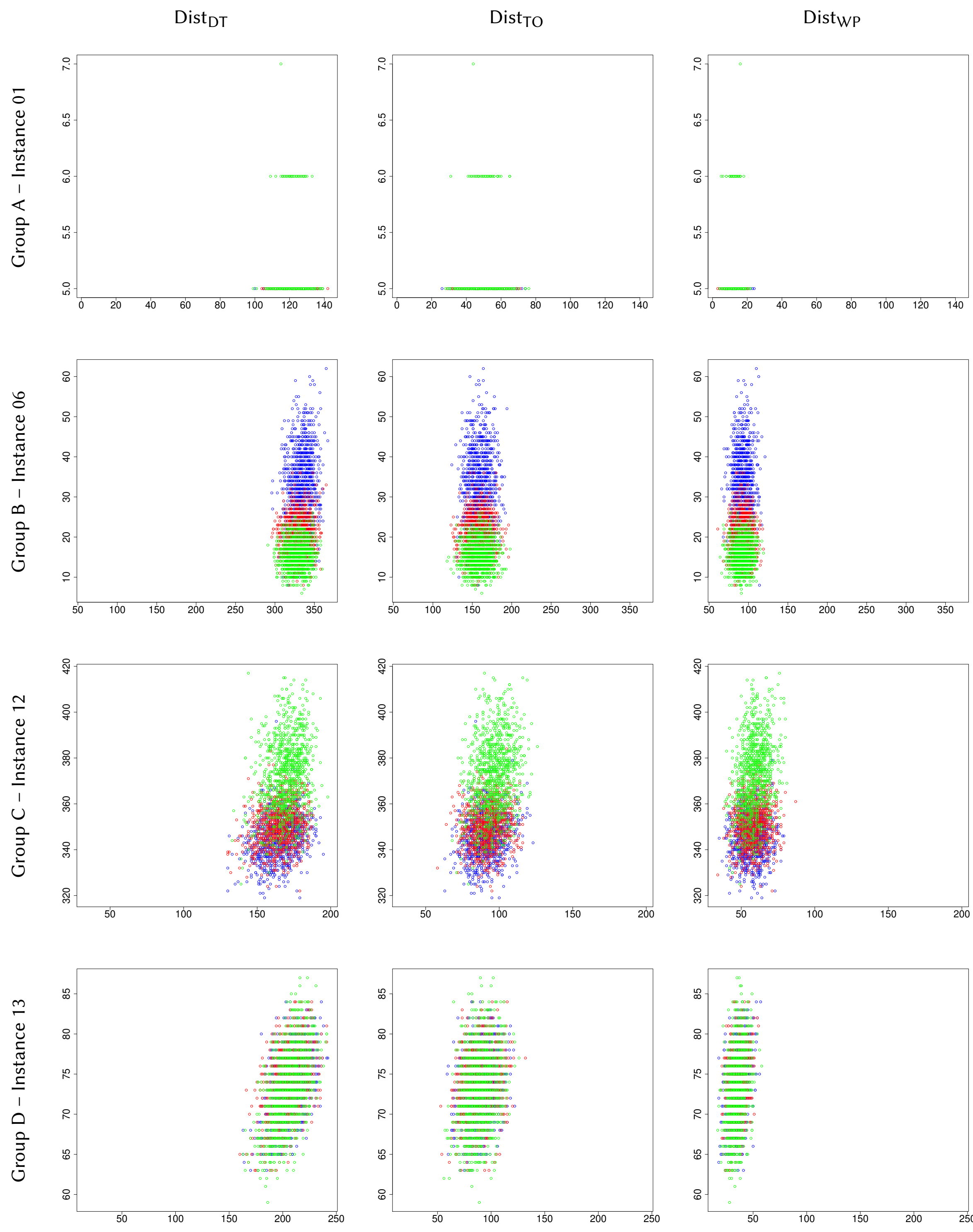


**Figure 3: Fitness-Distance Scatter Plots for the three distance metrics. HCGD, SA and HLS solutions are represented in blue, green and red respectively. Distance is on x-axis and Fitness on the y-axis.**

**Table 2: Mean neutral degree for each instance and method, for accepted solutions within the HC and SA phases and mean neutral degree for HLS at the pseudo-local optima level across 100 runs.**

| Gr | Inst | Global Level | Landscape Level | | |
|---|---|---|---|---|---|
| | | HLS | HCGD | SA | HLS |
| A | 01 | 7.76 | 99.20 | 29.80 | 37.40 |
| | 11 | 0.20 | 99.50 | 35.50 | 85.60 |
| B | 02 | 9.23 | 99.10 | 53.40 | 46.30 |
| | 03 | 3.43 | 99.10 | 38.70 | 44.90 |
| | 06 | 1.41 | 99.20 | 48.10 | 52.90 |
| | 07 | 3.12 | 99.10 | 55.40 | 48.50 |
| | 08 | 12.65 | 99.30 | 49.80 | 69.70 |
| | 10 | 2.15 | 99.20 | 53.70 | 75.60 |
| | 14 | 3.58 | 99.10 | 46.40 | 94.10 |
| | 15 | 8.98 | 99.10 | 45.10 | 97.80 |
| | 16 | 2.48 | 99.10 | 52.50 | 99.00 |
| | 17 | 2.44 | 99.30 | 50.40 | 99.20 |
| | 19 | 5.18 | 99.30 | 42.50 | 79.60 |
| | 20 | 0.97 | 99.00 | 55.80 | 59.10 |
| | 21 | 4.90 | 99.30 | 52.00 | 80.00 |
| C | 05 | 2.66 | 99.10 | 36.00 | 51.00 |
| | 09 | 3.72 | 99.20 | 48.00 | 95.40 |
| | 12 | 1.47 | 99.10 | 50.40 | 78.90 |
| | 18 | 4.40 | 99.20 | 29.30 | 98.40 |
| D | 04 | 1.85 | 99.20 | 50.20 | 45.60 |
| | 13 | 3.28 | 99.40 | 43.60 | 74.70 |

**Table 3: Plateau size at search landscape level, where $\overline{\text{Median}}$ represents the mean across runs of the median plateau size in each run.**

| Inst | HCGD | | SA | | HLS | |
|---|---|---|---|---|---|---|
| | Mean | $\overline{\text{Median}}$ | Mean | $\overline{\text{Median}}$ | Mean | $\overline{\text{Median}}$ |
| 01 | 197.40 | 14.1 | 2.80 | 2 | 2.60 | 2.2 |
| 11 | 271.10 | 21.7 | 3.00 | 2 | 4.80 | 14.4 |
| 02 | 163.50 | 20.8 | 3.50 | 2.3 | 3.10 | 2.1 |
| 03 | 186.20 | 18.8 | 2.90 | 2.1 | 3.00 | 2.2 |
| 06 | 211.00 | 18.3 | 3.30 | 2.1 | 3.00 | 3.9 |
| 07 | 156.40 | 22.7 | 3.70 | 2.4 | 2.90 | 3.6 |
| 08 | 208.80 | 25.4 | 3.30 | 2.1 | 3.00 | 12.7 |
| 10 | 168.80 | 24.6 | 3.50 | 2.5 | 3.00 | 14.8 |
| 14 | 212.90 | 16.5 | 3.20 | 2.1 | 4.60 | 15.3 |
| 15 | 180.30 | 19 | 3.10 | 2.1 | 8.50 | 18.5 |
| 16 | 171.10 | 19.5 | 3.40 | 2.3 | 58.90 | 19.6 |
| 17 | 208.00 | 23.6 | 3.40 | 2.2 | 155.70 | 23.2 |
| 19 | 203.00 | 23.3 | 3.00 | 2.1 | 3.20 | 15.3 |
| 20 | 144.60 | 16.7 | 3.70 | 2.4 | 3.00 | 5.7 |
| 21 | 235.90 | 25 | 3.40 | 2.3 | 3.20 | 16.7 |
| 05 | 265.40 | 22 | 3.00 | 2.2 | 3.20 | 3.5 |
| 09 | 205.40 | 21.7 | 3.20 | 2.1 | 4.80 | 20 |
| 12 | 222.50 | 18.6 | 3.20 | 2.4 | 3.30 | 12.1 |
| 18 | 389.10 | 17.8 | 2.60 | 2 | 22.40 | 17.2 |
| 04 | 207.10 | 19.8 | 3.40 | 2.1 | 2.90 | 2.8 |
| 13 | 254.60 | 24.1 | 3.10 | 2.1 | 3.00 | 14 |

weighs more than 10 GB. An major difference between the mean and the mean of medians can be observed for HCGD and for some instances for HLS. This implies that there are very large plateaus which impact the mean.

Moreover mean and mean of medians confirm HCGD has almost always larger plateaus; that explains the worst results due to the consequent time wasted during plateau exploration. SA features a constant low plateau size between 2 and 4. The HLS plateau sizes are as expected, since HLS sometimes acts as HCGD but with smaller plateaus and sometimes as SA. As one may recall, HLS is composed of HCGD followed by SA. Furthermore there are no real pattern which could help us to predict the instance group or the single best method to use. Nevertheless, we know that SA and HLS obtain the best results and we can observe that their plateau sizes are smaller. As one may expect, less time spent exploring plateaus leads to better results.

## 7 CONCLUSION

In this paper we conducted a search landscape analysis of HLS and two of its search components when solving the CB-CTT problem: a hill climbing with a perturbation mechanism (HCGD) and a pure simulated annealing (SA). HLS won the ITC competition in 2007 but still remains one of the best algorithms to solve the provided instances. Our experiments show that HLS has been designed, as one might expect, to benefit from the advantages of both HCGD and SA. In order to analyze the structure of the landscape, we propose a new distance metric based on the working periods. This distance is highly correlated with the two metrics usually used in the literature and gives smaller distances between local optima. This distance is a good candidate for a more detailed landscape analysis, such as the analysis of local optima networks [9], without loss of information regarding the other ones. This is part of our planned future work. This will build upon an ongoing reimplementation from scratch of HLS because the original HLS cannot be easily instrumented given that not all the source code is publicly available.

We also looked into the nature of the neutrality, both at a global and at a search landscape level. Interestingly, we observed opposite behaviors. This may point to some landscape features from these levels being more useful than others to guide the search in the context of a landscape-aware algorithm.

The parameter values of HLS, as well as the algorithmic components, have been manually fixed by its original author. These choices are debatable following the fairly similar average performance between HLS and the embedded simulated annealing. Within the context of hyper-heuristic search, we would like to create a program that automatically configures the best algorithm between HLS, HCGD, SA, or others, using instance and landscape characteristics to obtain the best solution.

## BIBLIOGRAPHY

[1] Lionel Barnett. 2001. Netcrawling-optimal evolutionary search with neutral networks. In *Proceedings of the 2001 Congress on Evolutionary Computation*, Vol. 1. IEEE, Seoul, South Korea, 30–37 vol. 1. https://doi.org/10.1109/CEC.2001.934367

[2] Edmund Burke, Yuri Bykov, and Sanja Petrovic. 2001. A Multicriteria Approach to Examination Timetabling. In *Practice and Theory of Automated Timetabling III (Lecture Notes in Computer Science)*, Edmund Burke and Wilhelm Erben (Eds.). Springer, Berlin, Heidelberg, 118–131. https://doi.org/10.1007/3-540-44629-X_8

[3] Michael W. Carter. 2013. Timetabling. In *Encyclopedia of Operations Research and Management Science*, Saul I. Gass and Michael C. Fu (Eds.). Springer US, Boston, MA, 1552–1556. https://doi.org/10.1007/978-1-4419-1153-7_1047

[4] Gunter Dueck. 1993. New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel. *J. Comput. Phys.* 104, 1 (Jan. 1993), 86–92. https://doi.org/10.1006/jcph.1993.1010

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science* 220, 4598 (May 1983), 671–680. https://doi.org/10.1126/science.220.4598.671

[6] Tomáš Müller. 2005. *Constraint-based timetabling. PhD thesis, Charles University in Prague, Faculty of Mathematics and Physics.* Ph.D. Dissertation. Charles University in Prague, Faculty of Mathematics and Physics, Prague.

[7] Tomáš Müller. 2008. ITC2007 solver description: A hybrid approach. *Annals of Operations Research* 172 (Jan. 2008), 429–446. https://doi.org/10.1007/s10479-009-0644-y

[8] Gabriela Ochoa, Rong Qu, and Edmund K. Burke. 2009. Analyzing the landscape of a graph based hyper-heuristic for timetabling problems. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO '09)*. Association for Computing Machinery, New York, NY, USA, 341–348. https://doi.org/10.1145/1569901.1569949

[9] Gabriela Ochoa, Marco Tomassini, Sebástien Vérel, and Christian Darabos. 2008. A Study of NK Landscapes' Basins and Local Optima Networks. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* (Atlanta, GA, USA) *(GECCO '08)*. ACM, New York, NY, USA, 555–562. https://doi.org/10.1145/1389095.1389204

[10] PATAT. 2007. International Timetabling Competition 2007. http://www.cs.qub.ac.uk/itc2007/

[11] PATAT. 2019. ITC 2019: International Timetabling Competition 2019. https://www.itc2019.org/home

[12] Peter F. Stadler. 2002. Fitness Landscapes. *Appl. Math. and Comput* 117 (2002), 187–207.

[13] Nadarajen Veerapen and Gabriela Ochoa. 2018. Visualising the global structure of search landscapes: genetic improvement as a case study. *Genet. Program. Evolvable Mach.* 19, 3 (Aug. 2018), 317–349. https://doi.org/10.1007/s10710-018-9328-1