Local Optima Networks of the Permutation Flowshop Scheduling Problem: Makespan vs. Total Flow Time

Leticia Hernando*, Fabio Daolio†, Nadarajen Veerapen† and Gabriela Ochoa†
*Faculty of Computer Science, 20018 Donostia, Spain
University of the Basque Country UPV/EHU
†Computing Science and Mathematics, University of Stirling, Stirling FK9 4LA, Scotland, UK
email: leticia.hernando@ehu.eus, {fda,nve,goc}@cs.stir.ac.uk

***Abstract*—Local Optima Networks were proposed to understand the structure of combinatorial landscapes at a coarse-grained level. We consider a compressed variant of such networks with features that are meaningful for the study of search difficulty in the context of local search. In particular, we investigate different landscapes of the Permutation Flowshop Scheduling Problem. The insert and 2-exchange neighbourhoods are considered, and two different objective functions are taken into account: the makespan and the total flow time. The aim is to analyse the network features in order to find differences between the landscape structures, giving insights about which features impact algorithm performance. We evaluate the correlation between landscape properties and the performance of an Iterated Local Search algorithm. Visualisation of the network structure is also given, where evident differences between the makespan and total flow time are observed.**

I. Introduction

The study of the Flowshop Scheduling Problem (FSP) has gained importance in the last decades due to its relevance in many real-world settings [1], [2], [3]. In the FSP there are n jobs that consist of m operations and they have to be scheduled in m machines. We consider that the jobs are processed in the same order on different machines, what is known as the Permutation Flowshop Scheduling Problem (PFSP). When dealing with this problem, researchers have focused on different objectives: makespan, total flow time, tardiness, earliness, idle time, etc. In fact, most real life scheduling problems usually involve multiple objectives. Therefore, several techniques have been proposed in order to solve the Multi-Objective Flowshop Scheduling Problem (MOFSP). In [4], one can find the different approaches designed for the MOFSP.

We focus on algorithms based on local search, which are among the most widely used metaheuristics to solve combinatorial optimisation problems. These algorithms structure the search space defining a neighbourhood. In this sense, the concept of landscape is defined as the triple $(\Omega, f, \mathcal{N})$, where Ω is the search space, f is the objective function to optimise and $\mathcal{N}$ is the neighbourhood. The neighbourhood imposes some properties on the search space that condition the behaviour of this kind of algorithm. The performance of these algorithms is also affected by the objective function considered. Therefore, a relevant approach when developing effective metaheuristics is the analysis of alternative search landscapes: considering different neighbourhoods or different objective functions, for the same underlying problem.

Local Optimal Networks (LONs) were first introduced in [5], [6] as a useful tool for understanding the structure and properties of combinatorial landscapes. LONs model landscapes as graphs where nodes are the local optima and edges account for the probabilities of connecting the basins of attraction. The LON properties of permutation based problems have also been studied [7], [8]. In [9], a modification in the concept of LON was proposed in order to deal with landscapes with neutrality. The characteristics derived from local optima networks have also been found to correlate with the performance of heuristic search algorithms [10], [11], [12], [13], [14].

In this paper, a compressed variant of the LON is considered useful to deal with neutrality at the level of local optima and to study the notions of funnels in fitness landscapes. The term 'funnel' was first introduced in theoretical chemistry, and it was later studied in the combinatorial optimisation field [15]. It refers to the communities of local optima that can act as traps: when trapped in a sub-optimal funnel, a local search heuristic will not be able to escape even with relatively large random perturbations. We analyse the compressed network features of PFSP instances, working with different neighbourhoods and objective functions. The 2-exchange and the insert neighbourhoods are considered, as they are the most commonly used in the space of permutations. Regarding the objective functions, as suggested in [4], the majority of PFSP studies consider bi-objective formulations, where the two objectives are to minimise makespan and flow time. Therefore, our analysis is focused on optimising these two objective functions. However, instead of considering the bi-objective formulation, we study the two objectives separately. Our aim is to analyse the network properties in order to highlight differences between the landscapes, giving insights about features that impact algorithm performance in the single-objective formulation. Firstly, we study features of the compressed version of the LON, referring to sinks

978-1-5090-4601-0/17/$31.00

and local-optima neutrality, which have never been studied for this problem in previous work. Secondly, we analyse the correlation between these properties and the performance of an Iterated Local Search (ILS) implementation showing that it is more difficult for the algorithm to minimise total flow time than makespan. Notice that in [8], the features analysed were those derived from the non-compressed LONs and just makespan was considered as the objective function. Finally, we provide examples of local optima network visualisation, where evident differences in the structure given by the total flow time and makespan are observed.

The rest of the paper is organised as follows. The PFSP and the two objective functions, makespan and total flow time, are formally introduced in Section II. In section III, the LON model is briefly described. The network analysis is presented in Section IV, and in Section V, the correlation between the network features and ILS performance is studied. Some examples of local optima network visualisations are shown in Section VI. Finally, in Section VII the conclusions and future work are described.

II. Permutation Flowshop Scheduling Problem

In the Permutation Flowshop Scheduling Problem (PFSP), n jobs have to be scheduled on m machines. A job consists of m operations, and the j-th operation $(j = 1, \ldots, m)$ of each job must be processed on machine j for a given specific processing time without interruption. The jobs are processed in the same order on different machines. The processing times are fixed non-negative values, and every job is available at time zero. At a given time, a job can start on the j-th machine when its $(j-1)$-th operation has finished on machine $(j-1)$, and machine j is idle. So, the goal of the PFSP is to find the order in which the jobs have to be scheduled on the machines, in such a way that a criterion is minimised.

The solutions of this problem are coded as permutations. So, the search space is the space of permutations of size n. The number of feasible solutions is $n!$.

A. Makespan

The makespan is the total length of the schedule. Traditionally, it has been the criterion to be optimised in the PFSP: a shorter makespan corresponds to an overall higher throughput of the system.

Let $p_{\pi(i),j}$ be the processing time required by job $\pi(i)$ on machine j, then the completion time of job $\pi(i)$ on machine j, $c_{\pi(i),j}$, can be recursively calculated as:

$$c_{\pi(i),j} = \begin{cases} p_{\pi(i),j} & i = j = 1 \\ p_{\pi(i),j} + c_{\pi(i)-1,j} & i > 1, j = 1 \\ p_{\pi(i),j} + c_{\pi(i),j-1} & i = 1, j > 1 \\ p_{\pi(i),j} + \max\{c_{\pi(i)-1,j}, c_{\pi(i),j-1}\} & i > 1, j > 1 \end{cases}$$

Thus, in order to minimise the makespan, we need to minimise the completion time of job $\pi(n)$ on machine m:

$$C_{mak}(\pi) = c_{\pi(n),m}.$$

B. Total Flow Time

Recently, Total Flow Time (TFT) has captured the attention of the scientific community since it is more meaningful in the industry: a shorter TFT means a more responsive system, with short waiting times for jobs execution.

The TFT is the sum of the completion times of all the jobs on the machine m. The following formula expresses mathematically the concept of TFT for a permutation π of jobs, where $c_{\pi(i),m}$ stands for the completion time of job $\pi(i)$ $(i = 1, \ldots, n)$ on machine m:

$$C_{TFT}(\pi) = \sum_{i=1}^{n} c_{\pi(i),m}.$$

III. Local Optima Networks

A neighbourhood $\mathcal{N}$ in a search space Ω is a mapping that assigns, to each solution $\pi \in \Omega$, a set of neighbouring solutions $\mathcal{N}(\pi) \in P(\Omega) \setminus \emptyset$, where $P(\Omega)$ is the powerset of Ω:

$$\begin{array}{rccc} N: & \Omega & \longrightarrow & P(\Omega) \\ & \pi & \longmapsto & \mathcal{N}(\pi) \end{array}$$

Two examples of the most commonly used neighbourhoods in the space of permutations are given by the 2-exchange and the insert operators. The 2-exchange neighbourhood considers that two solutions are neighbours if one is generated by swapping two elements of the other. In the case of the insert neighbourhood, two solutions are neighbours if one is the result of moving an element of the other to a different position.

A solution $x^* \in \Omega$ is a local optimum (minimum) under a neighbourhood $\mathcal{N}$ if $f(x^*) \leq f(x),\ \forall x \in \mathcal{N}(x^*)$. Each local optimum has an associated basin of attraction. In general, the basin of attraction of a local optimum π^*, $\mathcal{B}(\pi^*)$, is the set composed of all the solutions that, after applying the local search algorithm starting from each of these solutions, the algorithm finishes in π^*. Particularly, a Stochastic Hill-Climbing algorithm is considered in this study (Algorithm 1). Note that, in the presence of neutrality, Algorithm 1 could reach different local optima starting from the same solution. Thus, one solution could belong to more than one basin of attraction. Let us denote by $\mathcal{H}$ the stochastic operator that associates, to each solution π, the local optimum π^* obtained after applying the algorithm ($\mathcal{H}(\pi) = \pi^*$). So, the basin of attraction $\mathcal{B}(\pi^*)$ of a local optimum π^* can be defined in the following way:

$$\mathcal{B}(\pi^*) = \{\pi \in \Omega \mid p(\mathcal{H}(\pi) = \pi^*) > 0\},$$

where $p(\mathcal{H}(\pi) = \pi^*) > 0$ means that, starting from π, the algorithm has a non-zero probability of reaching π^*.

There is a transition between two basins of attraction $\mathcal{B}(\pi_i^*)$ and $\mathcal{B}(\pi_j^*)$, if $\exists \pi \in \Omega$ such that:

$$d(\pi, \pi_j^*) \leq D \text{ and } \pi \in \mathcal{B}(\pi_i^*),$$

where $d(\pi, \pi_j^*)$ denotes the distance between π and π_j^* (minimum number of movements in the neighbourhood to convert π into π_j^*), and $D \in \mathbb{N}$ is a fixed distance threshold. Given a

Algorithm 1 Stochastic Best-improvement Hill-Climber

Choose an initial solution $\pi \in \Omega$
repeat
  $\pi^* = \pi$
  Randomly choose $\sigma \in \mathcal{N}(\pi)$ s.t. $f(\sigma) = \min_{\pi' \in \mathcal{N}(\pi)} \{f(\pi')\}$
  if $f(\sigma) < f(\pi)$ **then**
    $\pi = \sigma$
  end if
until $\pi = \pi^*$

value of D, the weight w_{ij} of the transition between $\mathcal{B}(\pi_i^*)$ and $\mathcal{B}(\pi_j^*)$ is:

$$w_{ij} = \left|\{\pi \in \mathcal{B}(\pi_i^*) | d(\pi, \pi_j^*) \leq D\}\right|$$

The **Local Optima Network (LON)** is the weighted and directed graph $G = (V, E)$, where V is the set of vertices that are the local optima, $V = \{\pi_1^*, \pi_2^*, \ldots, \pi_{nv}^*\}$, and there is an edge $e_{ij} \in E$ between two local optima i and j if $w_{ij} > 0$, that is, $E = \{e_{ij} | w_{ij} > 0\}$. We work with a reduced version of these graphs (lower number of vertices and edges) that is helpful to extract features meaningful for the study of the search difficulty. Given a LON, the **Compressed Local Optima Network (CLON)** is built by following two rules:

1) Only those edges e_{ij} that arrive at a local optimum π_j^* that is not worse than the departure local optimum π_i^* ($f(\pi_j^*) \leq f(\pi_i^*)$) are taken into account. The rest of the edges are removed from the network.
2) A set is formed by those connected local optima with the same fitness value. This set contracts to a single vertex in the new network.

Thus, the CLON is the directed graph $G_c = (V_c, E_c)$, where $V_c = \{v_1, v_2, \cdots, v_{nv_c}\}$ is the set of vertices that are local optima or sets of local optima with the same fitness (and connected in G), and $E_c \subseteq E$ is the set of edges e_{ij} that departure from a vertex worse than the destination vertex. In this sense, the network only accounts for non-deteriorating escaping moves, and thus, it allows for the identification of funnels and the extraction of meaningful features from the point of view of (iterated) stochastic local search algorithms.

IV. Results of the Network Analysis

A. Experimental Design

The instances of the PFSP used in the experiments have been created using the problem generator proposed by [16], which is based on the well-known Taillard's benchmark ([17], [18]). $n = 10$ jobs and $m \in \{5, 6, 7, 8, 9, 10\}$ machines are considered. For each combination of n and m, 30 instances are generated. Thus, all the instances are of permutation size $n = 10$, so that the experimentation is computationally affordable: $|\Omega| = 10! \approx 3.63 \cdot 10^6$. Algorithm 1 is applied starting from each solution of the search space, using two different neighbourhoods: insert and 2-exchange. As a result, the different local optima and their basins of attraction are obtained for both neighbourhoods. We fix $D = 1$ under the 2-exchange neighbourhood as the edge-escape distance, and the connectivities between the basins of attraction are calculated. Of course, different escaping criteria could be chosen. Once the LON is built, applying 1) and 2) described in Section III, we obtain the CLON for each instance, each objective function and each neighbourhood.

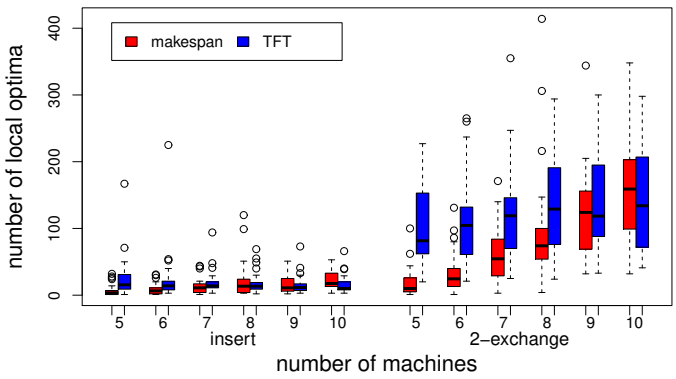


Fig. 1. Number of local optima of the PFSP instances according to the number of machines, distinguishing between the makespan (red) and the TFT (blue), and using the insert and the 2-exchange neighbourhoods.

B. Number of local optima

Figure 1 shows the number of local optima for the instances studied and both objective functions: makespan and TFT. When minimising the makespan, the number of local optima increases with the number of machines considering both neighbourhoods. This is consistent with the observation that increasing the number of machines (number of operations in each job) makes the problem harder to solve. However, in the case of the TFT and the insert neighbourhood, the number of local optima remains nearly constant as the number of machines increases. When using the 2-exchange neighbourhood, the number of local optima increases but much more slowly than for the makespan. For a small number of machines, when minimising the TFT, instances have a larger number of local optima and with a larger variance than when minimising the makespan. For a large number of machines, the number of local optima is similar for both objective functions. As expected, the number of local optima using the insert neighbourhood is lower than with the 2-exchange, as the insert neighbourhood is larger and thus, the landscape induced is smoother, this move operator being more suitable for solving instances of the PFSP.

C. Sinks

The CLON model is useful to study some features that it was not possible to examine with the LONs. These features seem to be more meaningful in order to analyse the search difficulty as they just consider the escaping edges that end at a better vertex. We focus on those CLON properties that refer to sinks. A **sink** s_i can be defined as a node that has at least one incoming edge but no outgoing edges:

$$s_i = \{v_i \mid \exists e_{ki} \in E_c \wedge \nexists e_{ik'} \in E_c\}$$

So, a sink is a reachable node, but there is no possibility of escaping from it to a better node. If a node v_i is formed by a

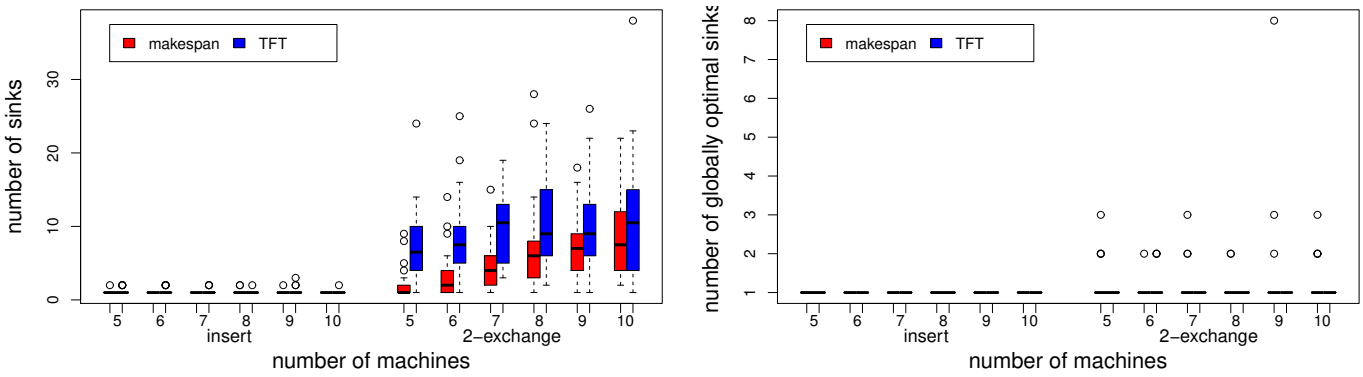


Fig. 2. Number of sinks (left) and globally optimal sinks (right) of the PFSP instances according to the number of machines, distinguishing between the makespan (red) and the total flow time (blue), and the insert and the 2-exchange neighbourhoods.

global optimum or a set of global optima, it is called a **globally optimal sink**.

In Figure 2, we show for both neighbourhoods the number of sinks and the number of globally optimal sinks. Although the number of local optima in both cases was considerably high, the number of sinks is low. Using the insert neighbourhood the majority of the instances have a single sink, and in all of them one globally optimal sink is observed. For the 2-exchange neighbourhood, the number of sinks is higher, and considering the minimisation of the makespan, a lower number of sinks is found than when minimising the TFT. Almost all of the instances have a single globally optimal sink, but in the case of the makespan, more instances with 2, 3 or even 8 globally optimal sinks are found.

According to these results, we could state that minimising the TFT is more difficult than minimising the makespan, at least when using the 2-exchange neighbourhood (for the insert neighbourhood, there are not palpable differences). Indeed, for the makespan, there is a lower number of sinks, and thus, lower number of nodes from which it becomes impossible to escape to a better one. At the same time, we find a considerably larger number of globally optimal sinks. The growth in the number of sinks confirms that the problem becomes harder as the number of machines increases.

The study of the connections between the different nodes and the globally optimal sinks is essential to understand whether the global optima are easily reachable or not. In general, starting from any node of the CLON, we can draw a path following the outgoing edges and moving from node to node, until a sink is found. The set composed of all the departure nodes of paths that finish at the same sink, is what we call the **funnel** of such sink. Notice that one node can have more than one outgoing edge. Thus, different sinks can be reached when departing from the same node. In this sense, we define the **unique nodes of a funnel** as the nodes inside this funnel which do not belong to any other different funnel.

We report in Figure 3, firstly, the average proportion of the size of the globally optimal funnels with respect to the sum of the sizes of all the funnels. Secondly, the average proportion of the number of the unique nodes of the globally optimal funnels with respect to the sum of the number of unique nodes of all the funnels. Finally, we show the Page Rank centrality score [19] of the globally optimal sinks.

Using the insert neighbourhood, the proportion of the size of the globally optimal funnel, as well as of the number of unique solutions in the globally optimal funnels, is one in almost all the instances. When minimising the TFT, more instances with a lower proportion appear than when minimising the makespan. For the 2-exchange neighbourhood, these proportions decrease with the number of machines. The proportions for the TFT are much lower than for the makespan. In fact, for the TFT, almost all the instances have a proportion of the size of the globally optimal funnels, or about 0.7, and a proportion of the number of unique solutions in the globally optimal funnels lower than 0.2. This is additional evidence of the TFT being more difficult to minimise than the makespan. Regarding the Page Rank of the globally optimal sink, a similar behaviour is observed for both neighbourhoods. It decreases with the number of machines and the Page Rank for the TFT is lower than for the makespan. The Page Rank is also lower when considering the 2-exchange.

D. Neutrality

Neutrality of the PFSP instances has been extensively studied in [20], in particular with respect to the proportion of solutions that have the same fitness value within the insertion neighbourhood. In this work, we focus instead on neutrality at the coarser level of the CLON.

As a first indicator, the proportion of unique fitness values to the number of nodes ($|V_c| = nv_c$) is evaluated. If there is no neutrality, this value is 1. In the opposite case, this value would be $1/nv_c$ when all the nodes of the CLON are global optima. Then, the average size of the nodes of the CLON is also analysed. If one node is composed of a single local optimum, its size is 1; while if it is a set of local optima, the size of the node is the cardinality of this set. So, this measure gives information about the number of local optima with the same fitness that are connected to each other.

These two properties are represented in Figure 4. For the 2-exchange, more nodes with the same fitness are found than

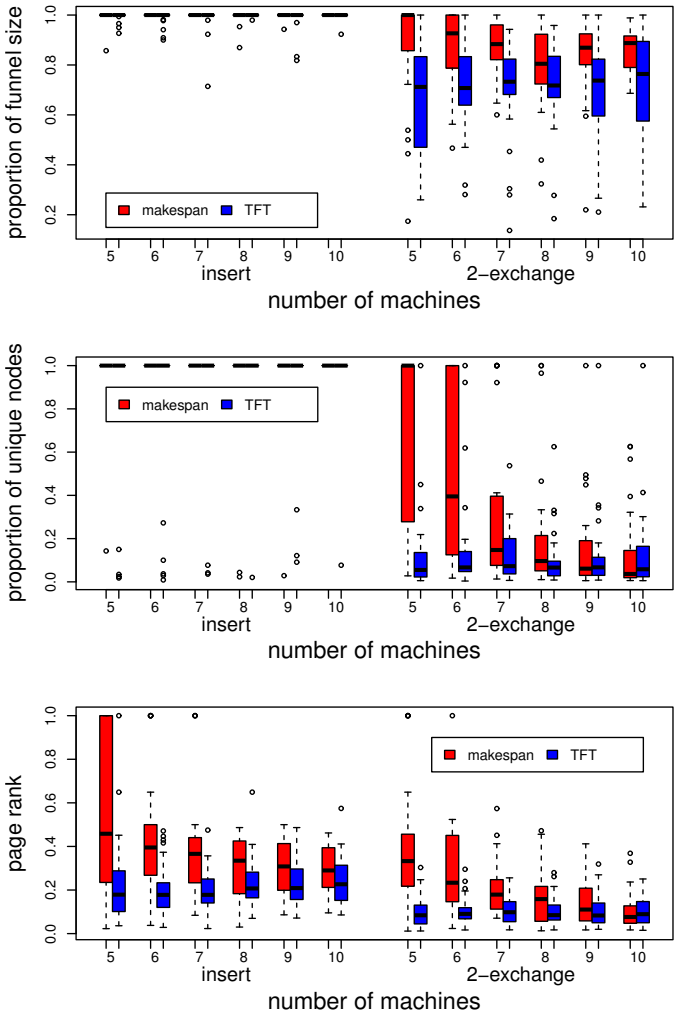


Fig. 3. Average proportion of the size of the globally optimal funnels with respect to the sum of the sizes of all the funnels (top), average proportion of the number of unique solutions in the globally optimal funnels with respect to the total number of unique solutions (centre), and Page Rank of the globally optimal sinks (bottom), of the PFSP instances according to the number of machines, distinguishing between the makespan (red) and the TFT (blue), and using the insert and the 2-exchange neighbourhoods.

for the insert neighbourhood (the proportion of unique fitness values is lower). Minimising the makespan, the proportion of unique fitness values decreases with the number of machines, so that more neutrality is present in the instances with a high number of machines. However, this behaviour is not observed for the TFT. In fact, the proportion is higher for the TFT. This reflects the fact that there is less neutrality at the nodes of the CLONs for the TFT than for the makespan. This correlates with the result about the average size of the nodes: for the TFT they are lower than for the makespan. Precisely, for the TFT, the average size of all instances is about 1.

V. Performance of the Iterated Local Search

A. Iterated Local Search

Iterated Local Search (ILS) algorithms are built to escape from local optima [21]. The general scheme is given in Algorithm 2. So, we need to specify the following steps:

(i) Generation of the Initial Solution
(ii) Local Search
(iii) Perturbation
(iv) Acceptance Criterion

In the present work, (i) the initial solutions are taken uniformly at random in the space of permutations of size $n = 10$. (ii) The Local Search is the same Stochastic Best-improvement Hill-Climber of Algorithm 1. We consider two different neighbourhoods for this heuristic: insert and 2-exchange. (iii) As the perturbation operator, we choose one 2-exchange movement. Finally, (iv) only those local optima that improve the current fitness are chosen. The search terminates at the global optimum, which is known a priori, or when reaching a pre-set limit of fitness evaluations: $FE_{max} = 0.1|\Omega| = 0.1 \cdot 10! = 362880$.

Algorithm 2 Iterated Local Search

$\pi_0 \leftarrow$ GenerateInitialSolution
$\pi^* \leftarrow$ LocalSearch(π_0)
repeat
  $\pi' \leftarrow$ Perturbation(π^*)
  $\pi'^* \leftarrow$ LocalSearch(π')
  $\pi^* \leftarrow$ AcceptanceCriterion(π^*, π'^*)
until *termination condition met*

B. Performance Evaluation

For the performance criterion, we use the expected number of function evaluations to reach the global optimum (success) after independent restarts of the ILS algorithm (Algorithm 2) [22]. This measure accounts for both the success rate ($p_s \in (0, 1]$) and the convergence speed. After $N-1$ unsuccessful runs stopped at T_{us} steps and the final successful one running for T_s steps, the total run-length would be $T = \sum_{k=1}^{N-1}\{T_{us}\}_k + T_s$. Taking the expectation and considering that N follows a geometric distribution (Bernoulli trials) with parameter p_s, gives:

$$\mathbb{E}(T) = \left(\frac{1-p_s}{p_s}\right)\mathbb{E}(T_{us}) + \mathbb{E}(T_s)$$

where $\mathbb{E}(T_{us}) = FE_{max}$, the ratio of successful to total runs is an estimator for p_s, and $\mathbb{E}(T_s)$ can be estimated by the average running time of those successful runs.

Figure 5 shows the performance of the ILS for the different objective functions and neighbourhoods. The results for the estimated probability of success and the estimated run-length (at logarithmic scale) are displayed. In general, under the insert neighbourhood, the estimated probability of success is lower for the makespan than for the TFT, and the estimated run-lengths are larger. However, for the 2-exchange neighbourhood, the estimated probability of success for the makespan decreases with the number of machines while the estimated run-length increases. Thus, for a small number of machines, the estimated probability of success is larger for the makespan than for the TFT and the estimated run-length is lower. As the

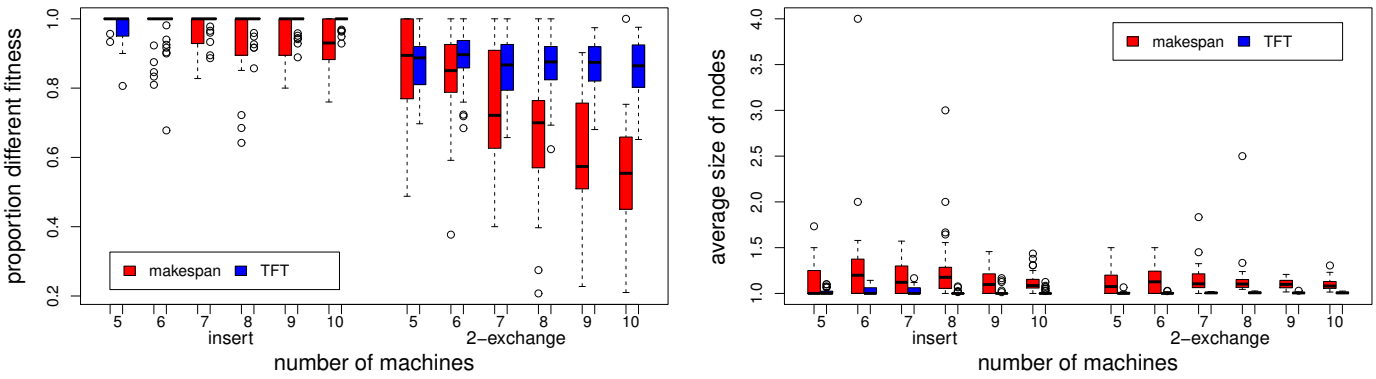


Fig. 4. Proportion of unique fitness values to number of nodes (left) and average size of the nodes (right) of the PFSP instances according to the number of machines, distinguishing between the makespan (red) and the TFT (blue), and the insert and the 2-exchange neighbourhoods.

number of machines increases, the values for both objective functions become quite similar.

C. Correlation with CLON features

This section explores the correlations between the CLON metrics analysed in Section IV and the ILS performance presented above. More precisely, Table I reports the rank-based Spearman's ρ statistic between each CLON metric and the ILS estimated run-length, considering the natural pairings of move operator and perturbation intensity between ILS variants and CLON models.

In all cases, the Page Rank of the global optima is the measure with the highest correlation: the lower the Page Rank of the global optima (*PR*) the longer it takes for the iterated search to solve an instance to optimality. Usually, the number of local optima ($\#LO$) have been considered as a complexity measure for algorithms based on local search. Here, we show that there are other metrics with a higher correlation with the ILS performance than the local optima. For example, the number of sinks ($\#sinks$), the proportional size of the globally optimal funnels (*GOfunnel*) and the proportion of the number of unique nodes (*#unique GOfunnel*) in the globally optimal funnels. In the case of the insert neighbourhood and minimising the makespan, the correlation for these three metrics is not so high. Note that, in this scenario, the number of sinks in almost all the instances was 1, and the proportion of the size and of the number of unique nodes in the globally optimal funnels were also 1. The same is observed for the number of globally optimal sinks ($\#GOsinks$): under the insert neighbourhood this value is 1 in all the instances. The proportion of unique fitness values (*#unique fitness*) has also a high influence on the ILS performance, being higher than that of the $\#LO$ for the TFT and the 2-exchange neighbourhood. The correlation with the size of the nodes ($|v_i|$) under the insert neighbourhood is higher than for the 2-exchange.

VI. Visualisation

One of the advantages of modelling an instance as a network is the possibility of visualising it. This section is devoted to provide an example of the visualisation of the CLONs. The graphs have been created using the igraph package in the R programming language [23]. As the CLON model indicates, the nodes are sets of local optima and the edges represent escape transitions according to one 2-exchange move. The size of nodes is proportional to the number of incoming edges, thus, it reflects the extent to which nodes attract the search dynamics. The colour of the nodes reflects their funnel membership. The palette follows a red to yellow gradient, where red identifies the global optima, and the yellow colour gradient reflects increased fitness. The nodes that belong to more than one funnel are in grey. We present both 2D and 3D images. The X and Y coordinates are determined by a graph layout algorithm, while in the 3D visualisation, the Z coordinate indicates the fitness values [15].

As a representative case, in Figure 6 we show the resultant CLON of one PFSP instance with 10 jobs and 7 machines, when minimising the TFT (6a) and the makespan (6b). In this example, the neighbourhood used in the local search is the 2-exchange. Clearly, the structure observed for the TFT is more complex than for the makespan. In the makespan, a lower number of nodes appear and just one funnel is found where all the nodes connect with the global optimum. However, in the TFT, there are 4 sinks and there is a large number of nodes belonging to more than one funnel. Although there is a very large number of nodes, only 3 of them connect uniquely with the global optimum.

VII. Conclusion

Local Optima Networks were proposed to understand the structure of combinatorial landscapes at the intermediate scale of local minima and their basins of attraction. In this paper, a variant of such model was presented: the Compressed Local Optima Network. The number of nodes and edges are reduced, yet this compressed network has features that are meaningful for the study of search difficulty. These properties were studied for different landscapes of the PFSP. The neighbourhoods considered were 2-exchange and insert. Regarding the objective functions, our analysis was focused on minimising makespan and total flow time. We examined the properties in order to

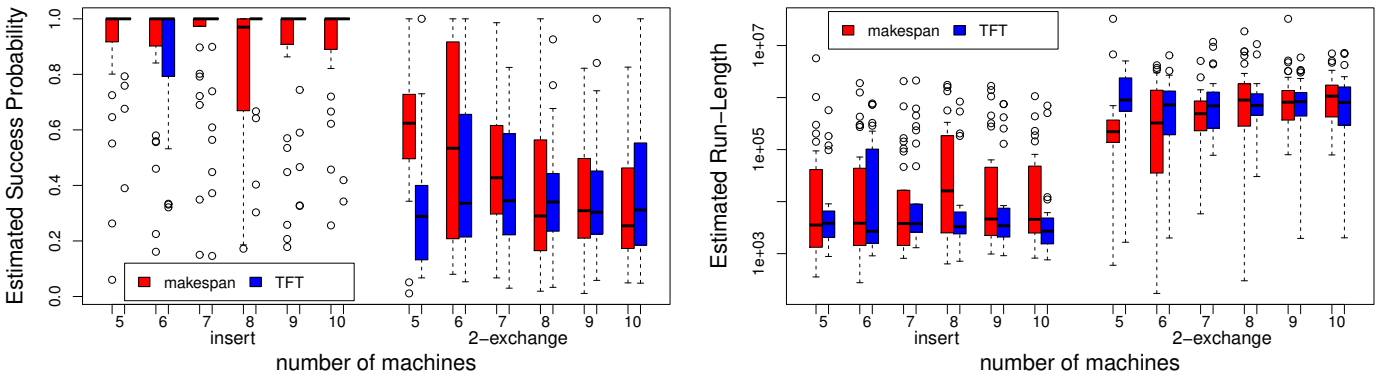


Fig. 5. Performance of Iterated Local Search according to the number of machines, distinguishing between the makespan (red) and the TFT (blue), and the insert and the 2-exchange neighbourhoods. Estimated success probability (left) and estimated run-length (right).

TABLE I
SPEARMAN'S ρ STATISTIC FOR THE CORRELATION BETWEEN THE ESTIMATED RUN-LENGTH OF ILS VARIANTS AND THE CLON METRICS BY THE RESPECTIVE NEIGHBOURHOOD AND OBJECTIVE FUNCTION.

$\mathcal{N}$	f	$\#LO$	$\#sinks$	$\#GOsinks$	*GOfunnel*	*#unique GOfunnel*	*PR*	*#unique fitness*	$\lvert v_i \rvert$
insert	makespan	0.4682	0.2314	———	−0.2316	−0.2311	−0.6507	−0.2785	0.3229
	total flow time	0.3846	0.4608	———	−0.4618	−0.4597	−0.5707	−0.3024	0.2670
2-exchange	makespan	0.5353	0.6593	−0.1266	−0.6159	−0.6549	−0.7889	−0.5606	−0.0022
	total flow time	0.4545	0.6075	−0.1048	−0.8516	−0.8436	−0.9405	−0.4451	0.1278

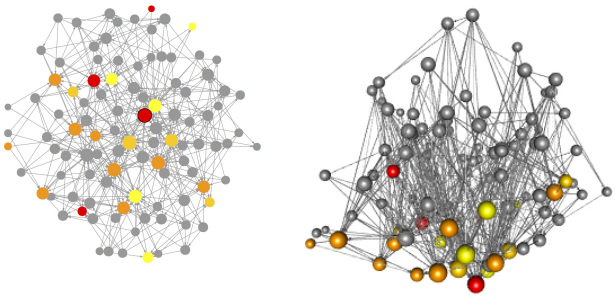
highlight differences between the landscapes, giving insights about features that impact algorithm performance.

The main conclusion obtained from the analysis of the network features is that minimising the TFT is more difficult than minimising the makespan. Indeed, in general, for the TFT, there is a larger number of local optima and a larger number of sinks, with just one of those sinks being the global optimum. Moreover, the proportion of the size of the globally optimal funnel and the proportion of the number of unique nodes inside this globally optimal funnel is lower for the TFT than for the makespan. The page rank of the globally optimal sink is also lower for the TFT. Especially, the differences between both objective functions are noticeable for the 2-exchange neighbourhood. This is due to the fact that the insert is a suitable operator for the PFSP for both objective functions, and therefore, similar properties are observed. However, for the insert neighbourhood, there is one feature for which we find the highest differences between the TFT and the makespan: the page rank of the globally optimal sink. Precisely, this is the feature for which we found the highest correlation with the ILS performance in all the landscapes.

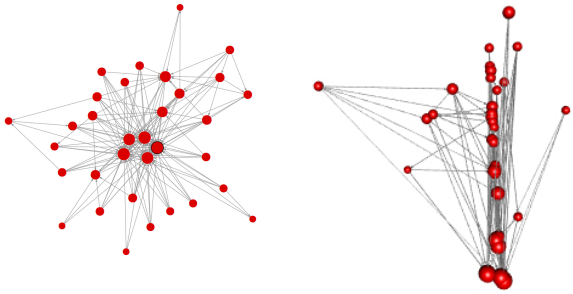
Other features, such as the number of sinks, the proportion of the size of the globally optimal funnel and the proportion of the number of unique nodes in the globally optimal funnel, also present a high correlation with the ILS performance. In fact, it is higher than that found for the number of local optima in all of the landscapes except for the makespan with the insert neighbourhood. Thus, another important conclusion derived from this work is that, although the number of local optima has usually been considered as a difficulty measure in the context of local search, the features presented here give more valuable information about it. Taking the makespan as the objective function, the presence of neutrality at the level of local optima is higher than when working with the TFT. In general, the proportion of the number of unique fitness values is lower for the makespan than for the TFT, with a larger size of the nodes. As observed, there is an inverse correlation between the presence of neutrality in the landscape and the ILS performance. Thus, this is another factor that confirms that minimising the makespan is easier than the TFT.

We showed two examples of the visualisation of the Compressed Local Optima Network for the same PFSP instance under the 2-exchange neighbourhood, but referring to the TFT and the makespan. Images were shown in 2D and 3D projection. In the 3D visualisation, the Z coordinate indicates the fitness values. So, the concepts of sink and funnel are easily interpreted. Clear differences in the structures of both landscapes were observed. While the network for the makespan is composed of just one funnel (the global optimum) and all the nodes connect with the globally optimal sink, a more complex network is obtained for the TFT.

All the information gathered in this work about the properties found for both objective functions helps in the development of approaches designed for the single-objective, as well as multi-objective optimisation. More objective functions for the PFSP and more instance classes (machine-correlated



(a) Total flow time – 2D (left) and 3D (right) projections



(b) Makespan – 2D (left) and 3D (right) projections

Fig. 6. CLON for a PFSP instance with 10 jobs and 7 machines, using the 2-exchange neighbourhood, considering the TFT (a) and the makespan (b). Images are shown in 2D and a 3D projection. Funnel structures are visualised in colours, with red indicating the globally optimal funnel, and the yellow gradient an increase in cost. Grey nodes are those belonging to more than one funnel. A black border in the 2D plots highlights the global optima. Node sizes are proportional to the number of incoming edges. The CLON for the TFT (a) shows a harder landscape than for the makespan (b), due to the multiple funnels and the more complex structure.

or mixed correlated) could be explored. In this article, the networks and their features were exhaustively calculated, so the whole search space needed to be evaluated. Thus, instances of size $n = 10$ were considered. Future work will study LONs in larger problems. This will require sampling to extract the LON models and the estimation of their features. The ultimate goal is to derive easy-to-compute landscape metrics that can predict the performance and guide the design of heuristic search algorithms when solving difficult combinatorial problems. This article is an additional step in this direction.

Acknowledgement

This work has been partially supported by IT-609-13 programs (Basque Government), TIN2016-78365-R (Spanish Ministry of Economy, Industry and Competitiveness) and the Leverhulme Trust (award number RPG-2015-395).

References

[1] E.-H. Aghezzaf and H. V. Landeghem, "An integrated model for inventory and production planning in a two-stage hybrid production system," *International Journal of Production Research*, vol. 40, no. 17, pp. 4323–4339, 2002.

[2] H. Boukef, M. Benrejep, and P. Borne, "A proposed genetic algorithm coding for flow-shop scheduling problems," *International Journal of Computers, Communications and Control*, vol. 2, no. 3, pp. 229–240, 2007.

[3] P. Chandra, P. Mehta, and D. Tirupati, "Permutation flow shop scheduling with earliness and tardiness penalties," *International Journal of Production Research*, vol. 47, no. 20, pp. 5591–5610, 2009.

[4] M. M. Yeniseya and B. Yagmahan, "Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends," *Omega*, vol. 45, no. C, pp. 119–135, 2014.

[5] G. Ochoa, M. Tomassini, S. Verel, and C. Darabos, "A study of nk landscapes' basins and local optima networks," in *Genetic and Evolutionary Computation Conference, GECCO 2008, Proceedings, Atlanta, GA, USA, July 12-16, 2008*, 2008, pp. 555–562.

[6] M. Tomassini, S. Vérel, and G. Ochoa, "Complex-network analysis of combinatorial spaces: The NK landscape case," *Physical Review E*, vol. 78, no. 6, pp. 66–114, 2008.

[7] F. Daolio, S. Verel, G. Ochoa, and M. Tomassini, "Local optima networks of the quadratic assignment problem," in *IEEE Congress on Evolutionary Computation (CEC 2010)*, 2010, pp. 2145–3152.

[8] ——, "Local optima networks of the permutation flow-shop problem," in *Artificial Evolution, EA 2013, Lecture Notes in Computer Science 8752*, 2014, pp. 41–52.

[9] S. Verel, G. Ochoa, and M. Tomassini, "Local optima networks of NK landscapes with neutrality," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 6, pp. 783–797, 2011.

[10] F. Chicano, F. Daolio, G. Ochoa, S. Verel, M. Tomassini, and E. Alba, "Local optima networks, landscape autocorrelation and heuristic search performance," in *Proceedings of Parallel Problem Solving from Nature, PPSN XII*, ser. Lecture Notes in Computer Science, vol. 7492. Springer, 2012, pp. 337–347.

[11] S. Verel, G. Ochoa, and M. Tomassini, "The connectivity of NK landscapes' basins: a network analysis," in *in Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, 2008, pp. 648–655.

[12] G. Ochoa, S. Verel, and M. Tomassini, "First-improvement vs. best-improvement local optima networks of NK landscapes," in *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I*, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds., 2010, pp. 104–113.

[13] F. Daolio, S. Verel, G. Ochoa, and M. Tomassini, "Local optima networks and the performance of iterated local search," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, 2012, pp. 369–376.

[14] G. Ochoa, S. Verel, F. Daolio, and M. Tomassini, "Local optima networks: A new model of combinatorial fitness landscapes," in *Recent Advances in the Theory and Application of Fitness Landscapes*, 2014, pp. 233–262.

[15] G. Ochoa and N. Veerapen, "Additional Dimensions to the Study of Funnels in Combinatorial Landscapes," in *Proceedings of the 2016 Genetic and Evolutionary Computation Conference, GECCO 2016*. New York, NY, USA: ACM, 2016, pp. 373–380.

[16] L. W. J. Watson, L. Barbulescu and A. Howe, "Contrasting structured and random permutation flow-shop scheduling problems: search-space topology and algorithm performance," *INFORMS Journal of Computing*, vol. 14, no. 2, pp. 98–123, 2002.

[17] E. Taillard, "Some efficient heuristic methods for the flow shop sequencing problem," *European Journal of Operational Research*, vol. 47, no. 1, pp. 65–74, 1990.

[18] ——, "Benchmarks for basic scheduling problems," *European Journal of Operational Research*, vol. 64, no. 2, pp. 278 – 285, 1993.

[19] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Seventh International World Wide Web Conference*, vol. 30, no. 1-7, 1998, pp. 107–117.

[20] M.-E. Marmion, C. Dhaenens, L. Jourdan, A. Liefooghe, and S. Verel, "On the neutrality of flowshop scheduling fitness landscapes," in *International Conference on Learning and Intelligent Optimization*. Springer Berlin Heidelberg, 2011, pp. 238–252.

[21] H. R. Lourenço, O. C. Martin, and T. Stützle, *Handbook of Metaheuristics*. Boston, MA: Springer US, 2010, ch. Iterated Local Search: Framework and Applications, pp. 363–397.

[22] A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm." in *IEEE Congress on Evolutionary Computation (CEC 2005)*, vol. 2, 2005, pp. 1777–1784.

[23] G. Csárdi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, no. 1695, pp. 1–9, 2006.