



# Local Optima Networks of the Permutation Flow-Shop Problem

Fabio Daolio, Sébastien Verel, Gabriela Ochoa, Marco Tomassini

## ► To cite this version:

Fabio Daolio, Sébastien Verel, Gabriela Ochoa, Marco Tomassini. Local Optima Networks of the Permutation Flow-Shop Problem. International Conference on Artificial Evolution (EA 2013), Oct 2013, Bordeaux, France. hal-00945475

**HAL Id: hal-00945475**

**<https://hal.science/hal-00945475>**

Submitted on 9 Sep 2021

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

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

# Local Optima Networks of the Permutation flow-shop Problem

Fabio Daolio<sup>1</sup>, Sébastien Verel<sup>2</sup>, Gabriela Ochoa<sup>3</sup>, Marco Tomassini<sup>1</sup>

<sup>1</sup> Information Systems Department, University of Lausanne, Lausanne, Switzerland.  
{fabio.daolio, marco.tomassini}@unil.ch

<sup>2</sup> Université Nice Sophia Antipolis and INRIA Lille - Nord Europe, France.  
verel@i3s.unice.fr

<sup>3</sup> Computing Science and Mathematics, University of Stirling, Scotland, UK.  
gabriela.ochoa@cs.stir.ac.uk

**Abstract.** This article extracts and analyzes local optima networks for the permutation flow-shop problem. Two widely used move operators for permutation representations, namely, swap and insertion, are incorporated into the network landscape model. The performance of a heuristic search algorithm on this problem is also analyzed. In particular, we study the correlation between local optima network features and the performance of an iterated local search heuristic. Our analysis reveals that network features can explain and predict problem difficulty. The evidence confirms the superiority of the insertion operator for this problem.

## 1 Introduction

The number and distribution of local optima in a combinatorial search space are known to impact the search difficulty on the corresponding landscape. Understating these features can also inform the design of efficient search algorithms. For example, it has been observed in many combinatorial landscapes that local optima are not randomly distributed, rather they tend to be relatively close to each other (in terms of a plausible metric) and to the known global optimum; clustered in a “central massif” (or “big valley” if we are minimizing) [4, 11, 18]. Search algorithms exploiting this globally convex structure have been proposed [4, 18].

A recently proposed model of combinatorial fitness landscape *local optima networks*, captures in detail the distribution and topology of local optima in a landscape. The model was adapted from the study of energy landscapes in physics, which exist in continuous space [21]. In this network view of energy surfaces, vertices are energy minima and there is an edge between two minima if the system can jump from one to the other with an energy cost of the order of the thermal energies. In the combinatorial counterpart, vertices correspond to solutions that are minima or maxima of the associated combinatorial problem, but edges are defined differently, and are oriented and weighted. In a first version, the weights represent an approximation to the probability of transition between the respective basins in a given direction [6, 16, 23, 25]. This definition, although

informative, produced densely connected networks and required exhaustive sampling of the basins of attraction. A second version, *escape edges* was proposed in [24], which does not require a full computation of the basins. Instead, these edges account for the chances of escaping a local optimum after a controlled mutation (e.g. 1 or 2 bit-flips in binary space) followed by hill-climbing. This second type of edges has, up to now, only been explored for binary spaces [24]. Also, previous work on networks with both basin and escape edges considered a single move operator on the corresponding search space.

This article extracts, analyzes and compares local optima networks of the *Permutation Flow-shop Problem* considering two types of move operators commonly used for permutation representation, namely, *insertion* and *exchange*. The article goes further and studies correlations among network features and the performance of an iterated local search heuristic.

## 2 Methods

### 2.1 Permutation Flow-Shop Problem

This section describes the optimization problem, solution representation, and move operators considered in this study.

**Problem formulation.** In the *Permutation Flow-shop Problem* (PFSP), a flow of  $n$  jobs has to be scheduled for processing on  $m$  different machines in sequential order. Each of the  $n$  jobs will start at machine 1 and end at machine  $m$ . Concurrency and preemption are not allowed. In other words, job  $i$  can not start on machine  $j + 1$  until machine  $j$  has completed it, and execution must run to completion once started. For any operation, job  $i$  will require a given processing time  $d_{ij}$  on machine  $j$ . Hence, a solution to the PFSP is a job processing order  $\pi$ , i.e. a permutation of the sequence of  $n$  jobs, where  $\pi(i)$  denotes the  $i^{th}$  job in the sequence. The objective is to find the permutation  $\pi_{best}$  yielding the minimum *makespan*,  $C_{max}$ , which is defined as the earliest completion time of its last job,  $\pi_{best}(n)$ , on the last machine  $m$ .

**Search operators.** Several methods for solving the PFSP have been proposed [19], many of which are based on local search heuristics. For those, the choice of a move operator determines the topology of the search space [10]. We consider here two widely used operators for permutation representation. Namely, the *swap* (or *exchange*) operator, and the *shift* (or *insertion*) operator. *Exchange*( $x, y$ ) simply swaps the job at positions  $x$  and  $y$ , while *Insert*( $x, y$ ) selects a job at position  $x$  and inserts it into position  $y$ , shifting all others jobs; this operator is known to work well on the PFSP [22].

### 2.2 Local Optima Networks

This section overviews relevant definitions for building Local Optima Networks with Escape Edges in the presence of a neutral fitness landscapes.

A **fitness landscape** [20] is a triplet  $(S, V, f)$  where  $S$ , a search space, is the set of all admissible solutions,  $V : S \rightarrow 2^{|S|}$ , a neighborhood structure, is the function that assigns to every  $s \in S$  a set of neighbors  $V(s)$ , and  $f : S \rightarrow \mathbb{R}$  is a fitness function that maps the quality of the corresponding solutions.

Given a fitness landscape  $(S, V, f)$ , a **local optimum** ( $LO$ ), which is taken to be a maximum here, is a solution  $s^*$  such that  $\forall s \in V(s), f(s) \leq f(s^*)$ .

In our study, the search space is composed of job sequences  $\pi$  of length  $n$ , therefore  $|S| = n!$ . The neighborhood is defined by the two selected move operators, consequently  $|V(\pi)| = n(n-1)/2$  under the *exchange* operator and  $|V(\pi)| = (n-1)(n-1)$  under the *insertion* operator. Finally,  $f(\pi) = -C_{max}(\pi)$  that is to be maximized.

A **neutral neighbor** of  $s$  is a configuration  $x \in V(s)$  with the same fitness value  $f(x) = f(s)$ ; the size of the set  $V_n(s) = \{x \in V(s) \mid f(x) = f(s)\}$  gives the **neutral degree** of a solution, i.e. how many neutral neighbors it has. When this number is high, the landscape can be thought of as composed of several sub-graphs of configurations with the same fitness value. This is the case for the fitness landscape of PFSP [14].

A neutral network (connected sub-graph whose vertices are neutral neighbors), also called a plateau, is a **local optimum neutral network** if all of its vertices are local optima.

---

**Algorithm 1:** Stochastic Best-Improvement Hill-Climber

---

```

Choose initial solution  $s \in S$  ;
repeat
    randomly choose  $s'$  from  $\{z \in V(s) \mid f(z) = \max_{x \in V(s)} f(x)\}$ ;
    if  $f(s') \geq f(s)$  then
         $s \leftarrow s'$ ;
until  $s$  is in a Local Optimum Neutral Network;

```

---

Since the size of the landscape is finite, we can mark the local optima neutral networks as  $LONN_1, LONN_2, \dots, LONN_n$ . These are the vertices of the *local optima network* in the neutral case. In other words, we have a network whose nodes are themselves networks.

Algorithm 1 finds the local optima and defines their basins of attraction [16]. The connections among optima represent the chances of escaping from a  $LONN$  and jumping into another basin after a controlled move [24]. But in a neutral landscape, the partition of solutions into basins of attraction is not sharp: Algorithm 1 is a stochastic operator  $h$  and  $\forall s \in S$  there is a probability  $p_i(s) = P(h(s) \in LONN_i)$ . Therefore, the **basin of attraction** of  $LONN_i$  is the set  $b_i = \{s \in S \mid p_i(s) > 0\}$  and its size is  $\sum_{s \in S} p_i(s)$  [25]. If we perturb a solution  $s \in LONN_i$  by applying  $D$  random moves, we obtain a solution  $s'$  that will belong to another basin  $b_j$  with probability  $p_j$ , i.e. with probability  $p_j$ ,  $h(s')$  will eventually climb to  $LONN_j$ . The probability to go from  $s$  to  $b_j$  is then

$p(s \rightarrow b_j) = \sum_{s' \in b_j} p(s \rightarrow s')p_j(s')$ , where  $p(s \rightarrow s') = P(s' \in \{z \mid d(z, s) \leq D\})$  is the probability for  $s'$  to be within  $D$  moves from  $s$  and can be evaluated in terms of relative frequency. Escaping from  $LONN_i$  to the basin  $b_j$  after such a perturbation thus happens with probability  $w_{ij} = p(LONN_i \rightarrow b_j) = \frac{1}{\#LONN_i} \sum_{s \in LONN_i} p_i(s)p(s \rightarrow b_j)$ . Notice that  $w_{ij}$  might be different from  $w_{ji}$ .

The **Local Optima Network** (LON) is the weighted graph  $G = (N, E)$  where the nodes are the local optima neutral networks, and there is an arc  $e_{ij} \in E$  with weight  $w_{ij} = p(LONN_i \rightarrow b_j)$  between two nodes  $i$  and  $j$  if  $p(LONN_i \rightarrow b_j) > 0$ .

### 3 Local Optima Network Analysis

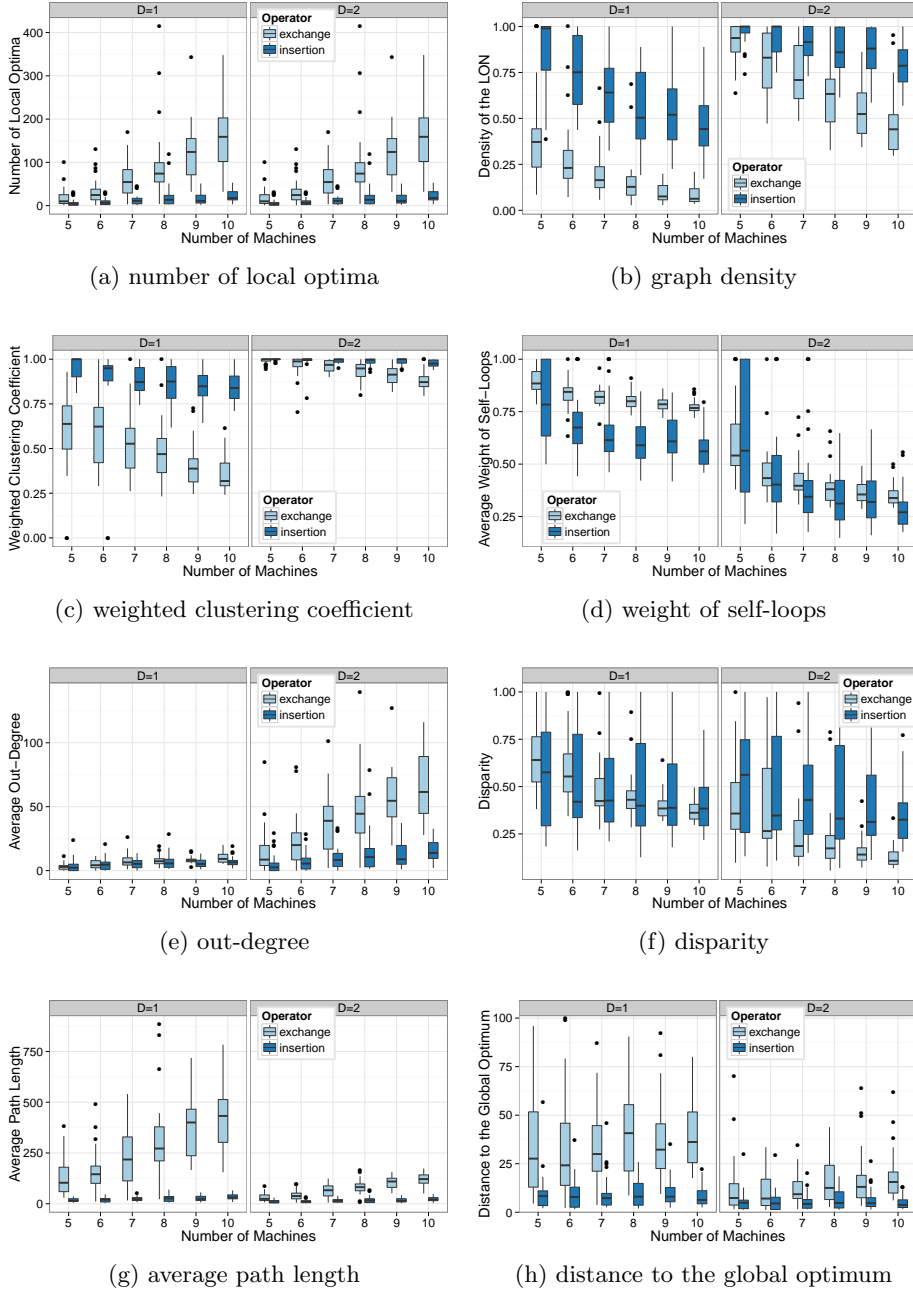
This section overviews the main topological features of the permutation flow-shop local optima networks. Networks were extracted for instances with  $n = 10$  jobs and  $m \in \{5, 6, 7, 8, 9, 10\}$  number of machines. Instances of the unstructured (random) class were generated using the problem generator proposed by Watson et al [26], which is based on the well-known Taillard benchmark [22]. For each combination of  $n$  and  $m$ , 30 instances were generated and results are presented through box-and-whiskers plots, to illustrate the distribution of the different metrics.

Four LON models are considered, namely, combining two neighborhoods: exchange and insertion, with two values of edge-escape distances:  $D = 1$  and  $D = 2$ . For building the models, local optima are obtained using Algorithm 1 with, respectively, exchange and insertion moves, whereas the escape-edges consider the exchange move for the 4 models. The Algorithms were implemented in C++ using the “ParadisEO” library [5]; data analysis and visualization use R [17] with the appropriate packages for network analysis and statistical computing.

**Network size:** Figure 1a shows that the number of local optima for all LON models increases with the number of machines. This is consistent with the observation that increasing the number of machines (number of constraints) makes the problem harder to solve. The number of optima does not depend on the edges model ( $D = 1$ ,  $D = 2$ ), therefore, the two subplots in Figure 1a are exactly the same. Figure 1a also indicates that the *exchange* LON model has a larger number of nodes as compared with the *insertion* model, which confirms that insertion is a better operator for the PFSP.

Figure 1b shows the density of edges, defined as the ratio of the LON number of edges to such number in a complete graph. As expected, the LON models with  $D = 2$  are more dense. The density decreases with the number of machines for all models, and it is higher for the insertion LONs.

**Clustering coefficient:** the *clustering coefficient* of a network is the average probability that two neighbors of a given node are also neighbors of each other. In the language of social networks, the friend of your friend is likely also to be your friend. The standard clustering coefficient [15] does not consider weighted



**Fig. 1.** Box-and-whiskers plots giving the distribution of LON features. Boxes comprise the 0.25 and 0.75-quantiles, with a thick black line at the median value (i.e. the 0.50-quantile). Whiskers extend for 1.5 times the inter-quantile range and define “outliers” values, depicted as black dots.

edges. We thus use the *weighted clustering* coefficient,  $CC^w$  proposed by [3]. In a random graph, the probability for links to form transitive closures would be the same as the probability to draw any link. Therefore, the clustering coefficient ( $CC$ ) would be the same as the graph link density ( $D_e$ ) [15]. By comparing the LONs weighted clustering coefficients  $CC^w$  on Figure 1c with their density of edges on Fig. 1b, we see that  $CC^w$  is higher on average than  $D_e$ . This suggests that the LONs have a local structure. Moreover, by looking at the clustering coefficient of un-weighted graphs (not presented here to save place), we notice that the weighted clustering coefficient  $CC^w$  is higher than the un-weighted coefficient, an evidence that high-probability transitions are more likely to form triangular closures than low-probability transitions.

**Transitions between optima:** Figure 1d reports the average transition probabilities of self-loops ( $w_{ii}$ ) within the networks. For all LON models, this metric decreases with the number of machines, and it is higher for the exchange operator. For all LON models,  $w_{ii}$  is on average higher than  $w_{ij, j \neq i}$  (not presented here). This suggests that a hill-climber after a perturbation from  $LONN_i$  is more likely to remain in the same basin than to escape it and reach another basin.

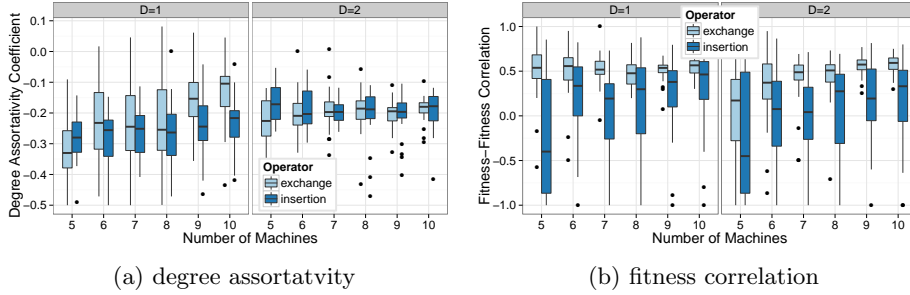
**Link heterogeneity:** Figure 1e shows the LON's average out-degree  $k_{out}$ , i.e. the average number of edges  $e_{ij}$  leaving a node  $i$ . As expected, the more dense LON models (with  $D = 2$ ) have higher out degree. For all models, this metric increases with the number of machines.

Figure 1e shows the *disparity* measure  $Y_2(i)$ , which gauges the weight heterogeneity of the arcs leaving a node [3]. For all models, this metric deviates from what would be expected in a random network, suggesting that the LON out-going edges are not equiprobable, but instead have predominant directions.

**Path lengths:** Figure 1g reports the LON's *average path length*. The length associated to a single edge  $e_{ij}$ , is  $d_{ij} = 1/w_{ij}$ , which can be interpreted as the expected number of random perturbations for escaping  $LONN_i$  and entering exactly the basin of  $LONN_j$ . The length of a path, then, is simply the sum of all the edge lengths in it. For all models, the path length increases with the number of machines. Path lengths are longer for the exchange LON with  $D = 1$ . The other LON models show short path lengths, specially for the insertion operator. Additional evidence supporting the advantage of this operator.

Figure 1h shows the average length of shortest paths that reach the global optimum starting from any other local optimum. This measure is clearly relevant to search difficulty. Shortest paths to the optimum reveal easy to search landscapes. Again, the insertion operator induces shortest distances, specially when coupled with an escape intensity  $D = 2$ .

**Mixing patterns:** Figure 2a reports on the tendency of LON nodes to connect to nodes with similar degree. Specifically, figure shows the Newman's  $r$  coefficient, a common measure of *assortativity* roughly equivalent the Pearson correlation between the endpoints degree of all the existing links [15]. Degree



**Fig. 2.** (a) Newman’s  $r$  coefficient of assortativity; (b) Spearman  $\rho$  correlation between the fitness of a node and the weighted average of its neighbors’ fitness.

mixing is known to have strong influence on the dynamical processes happening on complex network [2].

More interesting is to investigate mixing patterns with respect to the nodes fitness values. Figure 2b shows the Spearman correlation coefficient between the fitness of a  $LONN_i$  and the average value of its neighbors  $LONN_j$  fitnesses, weighted by the respective transition probabilities  $w_{ij,j \neq i}$ . This measure is less reliable on the small and dense LONs extracted from the *insertion* landscape, but on the *exchange* LONs, it suggests a positive fitness-fitness correlation that tends to increase with the number of machines. This might suggest that good solutions tend to be clustered within the search space.

More general and more pronounced is the positive correlation, measured by Spearman’s  $\rho$  statistic, between the fitness value of a node and the sum of the weights of its incoming transitions. Considering all instances,  $\rho$  is in the 95% confidence interval (0.78, 0.81), indicating that the higher the fitness of a  $LONN$ , the easier it is to reach it. This is consistent with results on other combinatorial spaces displaying a positive correlation between fitness and basin size [7].

## 4 The Performance of Iterated Local Search

The network metrics studied in the previous section, suggest that the insertion operator is preferable over the exchange operator, and that an escape distance of 2 ( $D = 2$ ) induces an easier to search landscape. In order to corroborate these predictions, this section studies the performance of a heuristic search algorithm, specifically, iterated local search, when running on the modeled PFSP instances. Moreover, we show that it is possible to predict the running time of ILS using multi-linear regression model based on LON features.

Iterated local search is a relatively simple but powerful strategy, which operates by alternating between a perturbation stage and an improvement stage. This search principle has been rediscovered multiple times, within different research communities and with different names. The term *iterated local search* (ILS) was proposed in [12]. Algorithm 2 outlines the procedure.



**Algorithm 2:** Iterated Local Search

---

```

 $s_0 \leftarrow$  Choose random initial solution  $s \in S$ ;
 $s^* \leftarrow \text{LocalSearch}(s_0, op)$ ;           // hill-climber using move operator  $op$ 
repeat
   $s' \leftarrow \text{Perturbation}(s^*, D)$ ;           //  $D$ -moves of random swap
   $s'^* \leftarrow \text{LocalSearch}(s', op)$ ;       // hill-climber using move operator  $op$ 
  if  $f(s'^*) > f(s^*)$  then
     $s^* \leftarrow s'^*$ ;                       // accept if better
until  $FE \leq FE_{max}$ ;

```

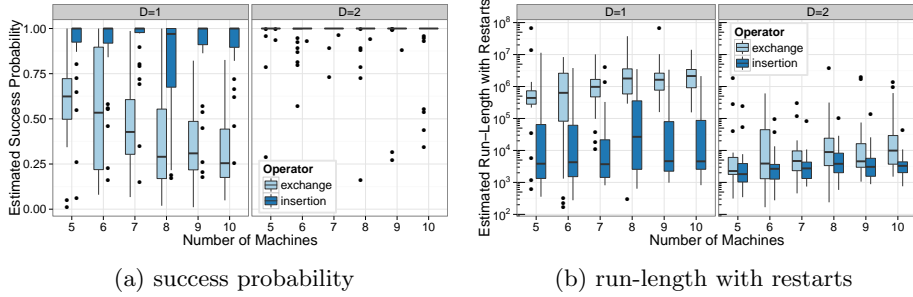
---

The *LocalSearch* procedure in Algorithm 2, corresponds to the stochastic hill-climber given in Alg. 1. In our implementation, the two operators studied: insertion and exchange can be used in this stage. The perturbation stage uses only the exchange operator but with two different intensities of one or two operator applications. Notice that Alg. 2 follows closely the structure of basins of the search space, and thus, the LON models should explain the performance of such ILS. Specifically, four ILS implementations are tested, namely, using insertion and exchange in the local stage, and using one or two applications of the exchange operator in the perturbation stage, which we denote  $D = 1$  and  $D = 2$ .

**Experimental setup:** The same instances studied in Section 3 are considered, i.e. unstructured (random) instances with  $n = 10$  jobs and  $m \in \{5, 6, 7, 8, 9, 10\}$  number of machines. The four variants of ILS (Algorithm 2 described above) are tested. The maximum running time is set to  $FE_{max} = 0.2|S| = 0.2 \cdot 10! = 725760$  function evaluations. On each instance, independent runs are randomly restarted 1000 times upon termination, which occurs either on finding the global optimum or on exhausting the  $FE$  budget.

For assessing the algorithms' performance, we use the expected number of function evaluations to reach the global optimum (*Run-Length* [9]), considering independent restarts of the ILS algorithms [1]. This 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  $p_s$ , it 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 successful runs.

**Comparing the performance of ILS variants:** Figure 3 compares the performance of the four ILS variants. Figure 3a reports the estimated probability of success, which is clearly superior for ILS variants with perturbation strength of 2 ( $D = 2$ ). In this case the ILS algorithm solves all instances to optimality in the median. For one perturbation ( $D = 1$ , in Fig. 3a), success rates are much lower, specially for the exchange operator, where they decrease with increasing number



**Fig. 3.** Performance of Iterated Local Search: (a) success probability, (b) run-length.

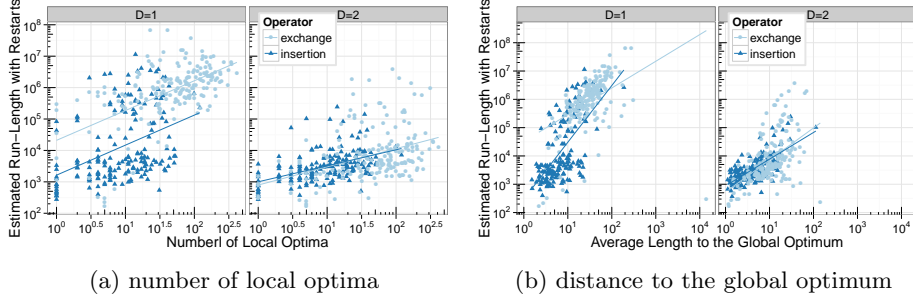
of machines. A closer look at the performance of the ILS variants is appreciated in Figure 3b, showing the estimated run-lengths. Run-lengths are much higher for ILS variants with a single exchange ( $D = 1$ ). For both  $D = 1$  and  $D = 2$ , the insertion operator produce shorter running lengths, although differences are greater when a single perturbation is used. Finally, for all ILS variants, the running length tends to increase with the number of machines. These performance observations, are consistent with the search difficulty predicted by the LON metrics in Section 3.

**Table 1.** Spearman’s  $\rho$  statistic for the correlation between the estimated run-length of ILS variants and the LON metrics by the respective move and perturbation.  $N_v$  nb of local optima,  $CC^w$  avg weighted clustering coeff.,  $F_{nn}$  neighboring nodes fitness-fitness corr.,  $k_{nn}$  neighboring nodes degree-degree corr.,  $r$  Newman’s assortativity,  $L_{opt}$  avg shortest distance to the global optimum,  $L_v$  avg path length,  $Fs_{in}$  fitness-strength(in) corr.,  $w_{ii}$  avg weight of self-loops,  $Y_2$  avg disparity of (out)links,  $k_{out}$  avg (out)degree.

ILS/LON	$N_v$	$CC^w$	$F_{nn}$	$k_{nn}$	$r$	$L_{opt}$	$L_v$	$Fs_{\rho}$	$w_{ii}$	$Y_2$	$k_{out}$
insertion $D1$	0.46	-0.221	0.199	0.078	0.238	0.634	0.40	-0.101	-0.31	-0.41	0.479
insertion $D2$	0.54	-0.209	0.316	-0.165	0.117	0.691	0.45	-0.167	-0.476	-0.46	0.55
exchange $D1$	0.535	-0.506	-0.004	0.142	0.353	0.624	0.536	-0.102	-0.235	-0.473	0.448
exchange $D2$	0.408	-0.255	0.22	-0.111	0.165	0.527	0.353	-0.035	-0.272	-0.434	0.409

**Performance prediction:** This section explores the correlations between the LON metrics from Section 3 and the ILS performance presented above. More precisely, Table 1 reports the rank-based Spearman’s  $\rho$  statistic between each LON metric and the ILS estimated run-length, considering the natural pairings of move operator and perturbation intensity between ILS variants and LON models. In all cases, the higher the number of local optima ( $N_v$ ) and, even more importantly, the longer the average lengths of paths to the global optimum ( $L_{opt}$ ), the longer it takes for the iterated search to solve an instance to opti-

mality. Figure 4 shows such correlations, which are the highest observed. Other scatter plots are less clear and are left out for reasons of space, but admittedly, their interpretation would also be less straightforward.



**Fig. 4.** Scatter plots of the estimated run-length versus different network metrics.

Finally, in order to investigate how the LON features could be used to predict the search difficulty on the whole set of explored landscapes, we propose a set of linear regression models having the estimated run-length as a dependent variable, log-transformed after a preliminary analysis (log-likelihood of Box-Cox’s power parameter). We perform an exhaustive search in the set of all possible regressors subsets [13] and for each subset size we retain the best model according to Mallows’s  $C_p$  statistic [8]. Results are given in Table 2. Interestingly, the number of local optima  $N_v$  is never chosen; instead, the best single predictor is the average length of the shortest paths to the global optimum  $L_{opt}$ , log-transformed, which alone accounts for more than 57% of the observed run-length variance across the PFSP instances under study.

**Table 2.** Exhaustive search among all regressors subsets for the multiple linear regression predicting the logarithm of estimated run-length as a function of the LON metrics. For each number of predictors  $\sharp P$ , the best model in terms of Mallows’s  $C_p$  statistic is given, along with its estimated regression coefficients and the resulting adjusted  $R^2$ .

$\sharp P$	$\log(N_v)$	$CC^w$	$F_{nn}$	$k_{nn}$	$r$	$\log(L_{opt})$	$\log(L_v)$	$Fs_\rho$	$w_{ii}$	$Y_2$	$k_{out}$	$C_p$	$adjR^2$
1						2.13						265.54	0.574
2		-5.18				1.43						64.06	0.675
3						1.481	0.895				-0.042	16.48	0.700
4		-2.079				1.473	0.540				-0.032	8.75	0.704
5		-2.388			-1.633	1.470	0.528				-0.030	5.97	0.706
6		-2.532			-1.722	1.469	0.472	-1.405			-0.028	3.75	0.707
7		-2.772			-1.986	1.461	0.427	-1.497		-0.408	-0.029	5.02	0.707
8		-2.748	-0.188		-2.078	1.464	0.452	-1.579		-0.515	-0.029	6.39	0.707

## 5 Conclusions

This article extracts and analyzes, for the first time, the local optima networks of the permutation flow-shop problem. The LON model with the so-called escape-edges, which account for the chances of escaping a local optimum after a controlled perturbation (1 or 2 random exchanges in our implementation), is extended to landscapes with neutrality. Two move operators, widely used for permutation representations (exchange and insert), are considered and contrasted.

LONs induced by the insertion operator present fewer nodes (i.e. fewer local optima), and shortest distances both among nodes and from any node to the global optimum. This evidence supports the superior performance of the insertion over the exchange move as reported in the literature. The LON models with  $D = 2$  produce shortest distances among nodes, and from any node to the global optimum, compared to models with  $D = 1$ . Therefore a local search heuristic using the insertion operator for adaptive walks and several kicks of the exchange operator to escape local optima, should perform well on these PFSP instances.

Indeed, four iterated local search variants were implemented and tested, which resemble the considered LON models. Among these, the ILS with insertion in the improvement stage and two exchanges in the perturbation stage, produced the best performance. This confirms the intuitions from the LON model metrics. Actually, not only the LON metrics correlate with the search performance, but also the ILS running time can be estimated using the LON features.

Future work will explore larger problems, which requires sampling to extract the LON models, and additional permutation flow-shop instance classes, such as machine-correlated and mixed-correlated instances [26]. 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.

## References

1. Auger, A., Hansen, N.: Performance evaluation of an advanced local search evolutionary algorithm. In: *Evolutionary Computation*, 2005. The 2005 IEEE Congress on. vol. 2, pp. 1777–1784. IEEE (2005)
2. Barrat, A., Barthlémy, M., Vespignani, A.: *Dynamical processes on complex networks*. Cambridge University Press (2008)
3. Barthélemy, M., Barrat, A., Pastor-Satorras, R., Vespignani, A.: Characterization and modeling of weighted networks. *Physica A: Statistical Mechanics and its Applications* 346(1), 34–43 (2005)
4. Boese, K.D., Kahng, A.B., Muddu, S.: A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters* 16, 101–113 (1994)
5. Cahon, S., Melab, N., Talbi, E.G.: Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics. *Journal of Heuristics* 10, 357–380 (2004)
6. Daolio, F., Tomassini, M., Vérel, S., Ochoa, G.: Communities of minima in local optima networks of combinatorial spaces. *Physica A Statistical Mechanics and its Applications* 390, 1684–1694 (2011)

7. Daolio, F., Verel, S., Ochoa, G., Tomassini, M.: Local Optima Networks of the Quadratic Assignment Problem. In: Evolutionary Computation (CEC), 2010 IEEE Congress on. pp. 1–8. IEEE (2010)
8. Gilmour, S.G.: The interpretation of mallows’s c.p-statistic. *The Statistician* pp. 49–56 (1996)
9. Hoos, H., Stützle, T.: Stochastic local search: Foundations and applications. Morgan Kaufmann (2005)
10. Jones, T.: Evolutionary Algorithms, Fitness Landscapes and Search. Ph.D. thesis, The University of New Mexico (1995)
11. Kauffman, S., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology* 128, 11–45 (1987)
12. Lourenço, H.R., Martin, O., Stützle, T.: Iterated local search. In: Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 57, pp. 321–353. Kluwer Academic Publishers (2002)
13. Lumley, T., Miller, A.: Leaps: Regression subset selection (2009), <http://CRAN.R-project.org/package=leaps>
14. Marmion, M., Dhaenens, C., Jourdan, L., Liefvooghe, A., Verel, S.: On the neutrality of flowshop scheduling fitness landscapes. *Learning and Intelligent Optimization* pp. 238–252 (2011)
15. Newman, M.: The structure and function of complex networks. *SIAM review* pp. 167–256 (2003)
16. Ochoa, G., Verel, S., Tomassini, M.: First-improvement vs. best-improvement local optima networks of nk landscapes. In: Parallel Problem Solving from Nature - PPSN XI. Lecture Notes in Computer Science, vol. 6238, pp. 104–113. Springer (2010)
17. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013), <http://www.R-project.org/>, ISBN 3-900051-07-0
18. Reeves, C.R.: Landscapes, operators and heuristic search. *Annals of Operations Research* 86, 473–490 (1999)
19. Ruiz, R., Maroto, C.: A comprehensive review and evaluation of permutation flow-shop heuristics. *European Journal of Operational Research* 165(2), 479–494 (2005)
20. Stadler, P.: Fitness landscapes. *Biological evolution and statistical physics* pp. 183–204 (2002)
21. Stillinger, F.: A topographic view of supercooled liquids and glass formation. *Science* 267, 1935–1939 (1995)
22. Taillard, E.: Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research* 47(1), 65–74 (1990)
23. Tomassini, M., Verel, S., Ochoa, G.: Complex-network analysis of combinatorial spaces: The NK landscape case. *Phys. Rev. E* 78(6), 066114 (2008)
24. Verel, S., Daolio, F., Ochoa, G., Tomassini, M.: Local Optima Networks with Escape Edges. In: Proceedings of International Conference on Artificial Evolution (EA-2011). pp. 10 – 23. Angers, France (Oct 2011)
25. Verel, S., Ochoa, G., Tomassini, M.: Local optima networks of nk landscapes with neutrality. *Evolutionary Computation, IEEE Transactions on* 15(6), 783–797 (2011)
26. Watson, J., Barbulescu, L., Whitley, L., Howe, A.: Contrasting structured and random permutation flow-shop scheduling problems: search-space topology and algorithm performance. *INFORMS Journal on Computing* 14(2), 98–123 (2002)