A fitness landscape analysis of Pareto local search on bi-objective permutation flowshop scheduling problems

Arnaud Liefooghe, Bilel Derbel, Sebastien Verel, Hernan Aguirre, Kiyoshi Tanaka

A Fitness Landscape Analysis of Pareto Local Search on Bi-objective Permutation Flowshop Scheduling Problems

Arnaud Liefooghe[1,2], Bilel Derbel[1,2], Sébastien Verel[3], Hernán Aguirre[4], and Kiyoshi Tanaka[4]

[1] Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL, F-59000 Lille, France
[2] Inria Lille – Nord Europe, F-59650 Villeneuve d'Ascq, France
[3] Univ. Littoral Côte d'Opale, LISIC, F-62100 Calais, France
[4] Shinshu University, Faculty of Engineering, Nagano, Japan

Abstract. We study the difficulty of solving different bi-objective formulations of the permutation flowshop scheduling problem by adopting a fitness landscape analysis perspective. Our main goal is to shed the light on how different problem features can impact the performance of Pareto local search algorithms. Specifically, we conduct an empirical analysis addressing the challenging question of quantifying the individual effect and the joint impact of different problem features on the success rate of the considered approaches. Our findings support that multi-objective fitness landscapes enable to devise sound general-purpose features for assessing the expected difficulty in solving permutation flowshop scheduling problems, hence pushing a step towards a better understanding of the challenges that multi-objective randomized search heuristics have to face.

1 Introduction

The multi-objective optimization community has spent a lot of efforts in the design of general-purpose techniques allowing to tackle hard optimization problems. Despite the number of available algorithms, their skillful design and their flexibility when applied to a large spectrum of problems, a key ingredient to make them efficient and effective lies in the choice of their components in order to be specifically adapted to the multi-objective optimization problem (MOP) being tackled. This might even depend on the intrinsic properties of the problem instance being considered. In this respect, there is evidence that new tools dedicated to the understanding of the behavior of existing algorithms in light of the properties of the MOP(s) under study are needed. We in fact argue that it is timely to set up new methodological tools allowing to systematically investigate the properties of multi-objective heuristics and to accordingly study and relate their performance and their effectiveness to a given MOP. Such tools exist in the single-objective optimization literature, where a number of paradigms from the so-called fitness landscape analysis [14] have proved to be extremely helpful in attaining such a goal. Motivated by their success and their accuracy, there was

recently several studies leveraging the single-objective case and pushing fitness landscape analysis a step toward the development of new statistical methodologies and the identification of general-purpose characteristics and features that fit the multi-objective nature of a given optimization problem; see e.g. [1, 5, 17]. However, still a relatively huge gap remains in bridging fitness landscape analysis with the design of multi-objective randomized search heuristics.

In this paper, we continue the efforts from the community in this direction by conducting a comprehensive fitness landscape analysis with respect to the co-called permutation flowshop scheduling problem (PFSP). Our interest in this problem class stems from the fact that scheduling problems in general, and the PFSP in particular, are ubiquitous to countless real-world applications and constitute one of the most important and challenging problem class from combinatorial optimization [10, 16]. Nonetheless, in the multi-objective setting and from a fitness landscape perspective, this problem class was only studied to a small extent. Given that a number of different objectives (e.g., makespan, tardiness) can be considered, one can naturally consider several multi-objective formulations, which makes the PFSP particularly appealing and challenging to investigate. In particular, we already know that optimizing each objective separately leads to single-objective problems with different degrees of difficulty [16]. However, extending such a knowledge to the multi-objective setting is far from being straightforward, given that different objective combinations can be considered. It is to notice that previous studies on the subject exist [2, 8]. We actually extend them in different perspectives, as will be described in the following. In particular, we restrict ourselves to the bi-objective PFSP. In fact, multiple proposals considering bi-objective PFSP formulations and instances exist, but it is not clear what are their differences and similarities, neither is clear their impact on the performance of multi-objective algorithms. We are here specifically interested in measuring such an impact and eliciting in an explicit and comprehensive manner the characteristics that makes a problem formulation and the characteristics of problem instances more difficult to solve than another for Pareto local search algorithms. Our contributions can be summarized as follows:

- By conducting an empirical analysis, we are able to study and characterize small-size bi-objective PFSP instances in terms of the correlation between the objective values, the number of Pareto optimal solutions, the number of ranks in non-dominated sorting, and the number of Pareto local optimal solutions for two widely-used neighborhood operators (namely exchange and insert).
- By plugging these operators into the framework of the so-called Pareto Local Search (PLS) algorithm [13], we conduct a comprehensive study on the behavior of the so-obtained variants of PLS when applied to different bi-objective PFSP formulations and instance types. It is to notice that PLS is reported to be one of the state-of-the-art approach for bi-objective PFSP, especially for small-size instances [12].
- As a byproduct, we are able to measure the individual effect of problem features on the performance of PLS variants in terms of success rate, that is the

probability to identify the whole Pareto set. More importantly, we report the joint impact of these features by considering two scenarios: (i) predicting the degree of difficulty of each instance for each algorithm variant, (ii) predicting which algorithm variant performs better for a given instance. For each scenario, a machine learning classification model based on random forests [4] is considered and analyzed in order to measure and hence to quantify how important problem features are when solving a particular instance. Our findings reveal important knowledge on the impact of multimodality, in terms of the proportion of Pareto local optimal solutions, in order to explain the difficulty faced by PLS when solving bi-objective PFSP instances.

In the remainder, we first describe in Section 2 the considered PFSP, the corresponding pairs of objectives that we shall study, and the different types of instances. In Section 3, we provide a comprehensive analysis and correlation study for the considered problem characteristics. In Section 4, the individual effect and the joint impact of problem features on algorithm performance are analyzed. In Section 5, we conclude the paper and discuss some open questions.

2 Bi-objective permutation flowshop scheduling problems

The Permutation Flowshop Scheduling Problem (PFSP) is one of the most popular optimization problem from scheduling [16]. Most research typically deals with a single-objective formulation aiming at minimizing the *makespan*, i.e. the final completion time. However, many other criteria can be formalized, and multi-objective PFSP formulations have also been investigated in the literature; see e.g. [16] for on overview. This section introduces some necessary definitions and presents different benchmark instances taken from the specialized literature.

Problem formulation and objectives. The PFSP consists in scheduling N jobs $\{J_1, J_2, \ldots, J_N\}$ on M machines $\{M_1, M_2, \ldots, M_M\}$. Machines are critical resources, i.e. two jobs cannot be assigned to the same machine at the same time. A job J_i is composed of M consecutive tasks $\{t_{i1}, t_{i2}, \ldots, t_{iM}\}$, where t_{ij} is the j^{th} task of the job J_i, requiring the machine M_j. A processing time p_{ij} is associated with each task t_{ij}, and a due date d_i is assigned to each job J_i. We here focus on the permutation PFSP, where the operating sequences of the jobs are identical and unidirectional on every machine. A candidate solution can be represented by a permutation of size N. Let X denote the set of all feasible solutions (permutations). For an instance of N jobs, there exists $|X| = N!$ feasible solutions. To evaluate the schedule of a PFSP instance, many criteria may be defined. In this study, we focus on minimizing two objectives (f_1, f_2) simultaneously from a panel of five criteria as listed in Table 1, and selected among the most widely investigated ones from the literature [16]. Following the notations from [16], this problem class is actually denoted as $F/prmu, d_i/\#(f_1, f_2)$. Given that the single-objective PFSP (minimizing each of these objectives separately) is known to be NP-hard [16] for instances with more than two machines ($M > 2$), bi-objective instances can typically not be solved to optimality.

Table 1. PFSP objectives ($C_i(x)$ is the completion time of Job J_i in schedule x).

name	formulation	description
$C_{\max}(x)$	$= \max_{i \in [\![1,N]\!]} C_i(x)$	maximum completion time (makespan)
$C_{\text{sum}}(x)$	$= \sum_{i \in [\![1,N]\!]} C_i(x)$	sum of completion times
$T_{\max}(x)$	$= \max_{i \in [\![1,N]\!]} \max(0, C_i(x) - d_i)$	maximum tardiness
$T_{\text{sum}}(x)$	$= \sum_{i \in [\![1,N]\!]} \max(0, C_i(x) - d_i)$	sum of tardiness
$T_{\text{card}}(x)$	$= \left\vert \{i \mid C_i(x) > d_i, i \in [\![1,N]\!]\} \right\vert$	number of late jobs

Definitions. Let $Z \subseteq \mathbb{R}^2$ denote the image of feasible solutions in the *objective space* when using the function vector $f = (f_1, f_2)$ such that $Z = f(X)$. The Pareto dominance relation is defined as follows. A solution $x \in X$ is dominated by a solution $x' \in X$, denoted as $x \prec x'$, if $f_k(x') \leqslant f_k(x)$ for all $k \in \{1, 2\}$, with at least one strict inequality. A solution $x \in X$ is *a Pareto optimal solution* (POS) if there does not exist any other solution $x' \in X$ such that $x \prec x'$. The set of all POS is the *Pareto set*, and its mapping in the objective space is the *Pareto front*. One of the most challenging issues in multi-objective optimization is to identify a minimal complete Pareto set, i.e. one Pareto optimal solution mapping to each point from the Pareto front. Since the PFSP is NP-hard [16], heuristics appear to be well suited to identify a *Pareto set approximation*.

Problem instances. The most common and challenging set of single-objective PFSP benchmark instances is due to Taillard [15]. Each instance file provides a processing time p_{ij} for each task t_{ij}. Each processing time is generated following a discrete uniform distribution: $p_{ij} \sim [\![\ 1\ ,\ 99\ ]\!]$. These instances are restricted to the objectives dealing with the completion time (i.e. $C_{\max}, C_{\text{sum}}$). Indeed, given that no due date is provided, tardiness-related objectives (i.e. $T_{\max}, T_{\text{sum}}, T_{\text{card}}$) cannot be considered. For this reason, multiple researchers have been interested in extending these single-objective instances by adding a due date d_i for every job J_i. Three due date generation techniques are described below:

1. Inspired by Basseur et al. [3], we propose to generate the due dates following the discrete uniform distribution: $d_i \sim [\![\ 50 \cdot M\ ,\ \texttt{lb\_cmax}\ ]\!]$, where `lb_cmax` is a lower bound of the optimal makespan for the instance under consideration, as defined in [15]. In [3], the authors used an upper bound instead of a lower bound. Using a lower bound avoids us to use any prior external knowledge (about the optimal solution) and likely results in tighter due dates. Hence, a due date d_i roughly lies between the average completion date of the first scheduled job and an optimistic estimate of the best completion time.
2. Liefooghe et al. [11] propose to generate the due dates as follows: $d_i \sim [\![\ \overline{p} \cdot M\ ,\ \overline{p} \cdot (N + M - 1)\ ]\!]$, such that $\overline{p}$ is the average-value of processing times. As a consequence, if all processing times are the same, a due date d_i would lie between the completion date of the first and of the last scheduled job.
3. Unlike the previous approaches, Minella et al. [12] generate the due dates as: $d_i \sim [\![\ p_i^\star\ ,\ p_i^\star \cdot 4\ ]\!]$, such that $p_i^\star = \sum_{j \in [\![1,M]\!]} p_{ij}$ is the sum of the processing times over all machines for job J_i. The authors argue that this method generates due dates that range from very tight to relatively tight values.

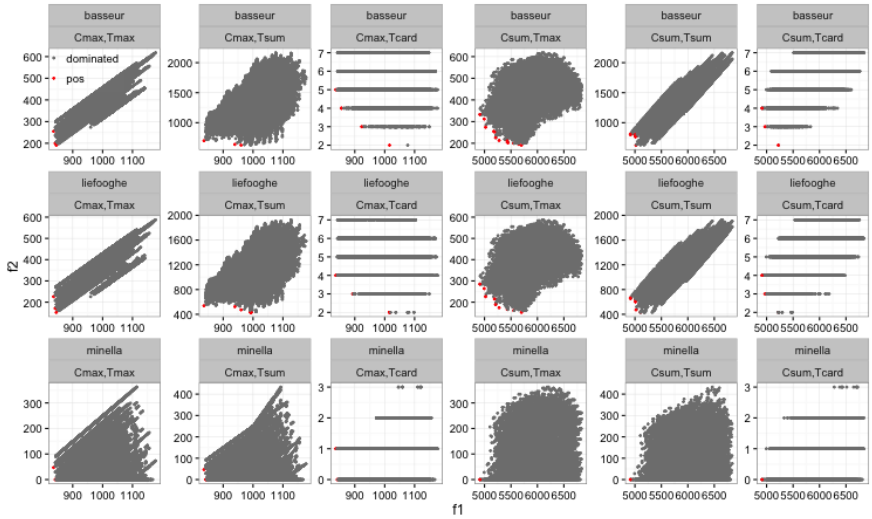

Fig. 1. Objective space ($N = 8$, $M = 8$, one instance).

We intuitively hypothesize that the way the due dates are generated, together with the pair of objectives to be minimized, have a high impact on the bi-objective PFSP instance to be solved. For this reason, we investigate the impact of the choice of these parameters on the fitness landscape characteristics and difficulty of small-size instances. More particularly, we consider the following parameter setting. The problem size is set to a small value of $N \in \{5, 6, 7, 8\}$ in order to enumerate the solution space exhaustively, from $|X| = 120$ for $N = 5$ up to $|X| = 40\,320$ for $N = 8$. The number of machines is $M \in \{5, 6, 7, 8\}$. For each value of `type`, N and M, 10 independently generated PFSP instances are considered. We investigate the following objective pairs: $(f_1, f_2) \in \{(C_{\max}, T_{\max}), (C_{\max}, T_{\text{sum}}), (C_{\max}, T_{\text{card}}), (C_{\text{sum}}, T_{\max}), (C_{\text{sum}}, T_{\text{sum}}), (C_{\text{sum}}, T_{\text{card}})\}$.

3 Problem features

In this section, we explore the characteristics of bi-objective PFSP instances in terms of visualization of the objective space, correlation between the objective values, number of Pareto optimal solutions, number of non-dominated fronts, and number of Pareto local optimal solutions. The section ends with a correlation analysis between each pair of those features and of the instance parameters.

Objective space. Following [2], in order to give a clear intuition of how the different objectives relate one to the other, we show in Fig. 1 a simple inspection of the objective space for one instance with $N = 8$ and $M = 8$. We can basically see that different objective pairs results in different objective space and Pareto front shapes. The objective space actually looks similar for the instances of type `basseur` and `liefooghe`, which is clearly to contrast to `minella`. This is clearly attributed to the differences of due date generation policies. Let us also observe that the pair of objectives $(C_{\max}, \star)$ seems to provide the same objective space

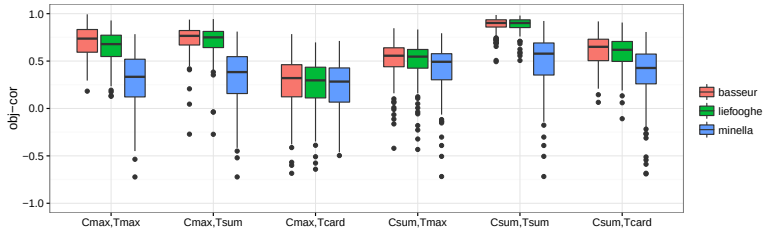

Fig. 2. Spearman rank correlation between the objective function values over the whole solution space for all the instances.

shape than its counterpart $(C_{\text{sum}}, \star)$ However, it is not clear that the convexity of the Pareto front is also similar for different pairs of objectives. Notice also that the degree of conflict between the two pairs of objectives $(C_{\max}, T_{\text{sum}})$ and $(C_{\text{sum}}, T_{\max})$ for both instance types `liefooghe` and `basseur` is less perceivable than for the two pairs $(C_{\max}, T_{\max})$ and $(C_{\text{sum}}, T_{\text{sum}})$, for which even the size of the Pareto front seems to be relatively smaller. This informal observation is to be extended in the next two paragraphs where the objective correlation and the Pareto front cardinality are quantified and analyzed more explicitly.

Objective correlation. In Fig. 2, we summarize the degree of conflict between the objectives by showing the box-plots obtained when computing the nonparametric Spearman rank correlation coefficient between the objective function values over the whole solution space and for all the instances of each class. Recall that the extreme value of 1 indicates a perfect correlation between the objective values, whereas -1 indicates that they are highly conflicting. A value around 0 indicates that both objectives are uncorrelated. The first notable observation is that the instances of type `minella` have a significantly smaller objective correlation than the two other types, except for $(C_{\max}, T_{\text{card}})$. The second notable observation is that all pairs of objectives are positively correlated, which suggests that by improving f_1, one has a high chance to also improve f_2. This is not surprising for PFSP because lowering the completion time naturally tends to lower the value of other objectives, but this rather uncommon when compared against other MOPs [7]. Notice that there exists a stronger correlation between the objectives dealing with completion time and the sum/maximum tardiness compared to the objective dealing with the number of late jobs, which might be attributed to the high number of plateaus for this function. The lowest correlation is for $(C_{\max}, T_{\text{card}})$ which is validated by Fig. 1, where the convex envelope of the objective space resembles a circle. The relatively high correlation between the objectives suggests that the Pareto set cardinality shall be limited.

Pareto optimal solutions. In Fig. 3, we report the proportion of Pareto optimal solutions (POS) in the solution space, computed and aggregated over all the instances of the same type for $N = 8$ and $M = 8$. This is related to the notion of intractability, which arises when the Pareto set cannot be enumerated in a polynomial amount of time [7]. For those instances, the number of POS goes from 0.001% up to 0.1% of the solution space. Besides observing that the objective pair has an impact on the proportion of POS, the most notable observation is

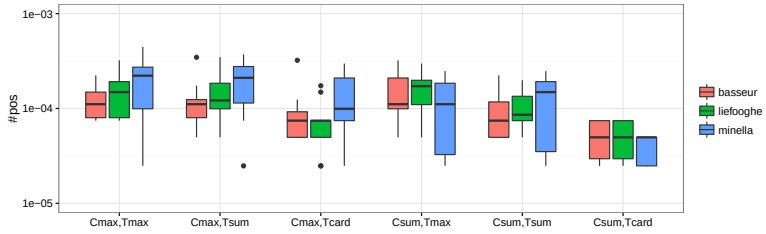

Fig. 3. Proportion of POS in the solution space ($N = 8$, $M = 8$).

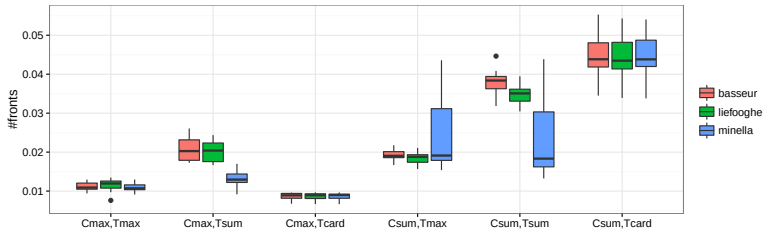

Fig. 4. Proportion of non-dominated fronts in the solution space ($N = 8$, $M = 8$).

that the instances of type `minella` have the highest proportion of POS for $C_{\max}$. This is more mitigated when analyzing the proportion of POS involving C_{sum}. At last, the instances of type `liefooghe` comes in second position, and typically exhibit more POS than `basseur`, which again confirms that the distribution of due date plays an important role in characterizing PFSP instances.

Non-dominated fronts. The previous feature provides a descriptive statistic about the *optimal* Pareto set. In this section, we examine another measure about how *non*-optimal solutions are structured. We use the same measure as [1] by looking at the impact of PFSP instance parameters on the number of non-dominated fronts. This is done by ranking all feasible solutions and layering them into fronts based on non-dominated sorting [9]. In Fig. 4, we report the proportion of non-dominated fronts, with respect to the size of the solution space, computed and aggregated over all the instances of the same type, for size $N = 8$ and $M = 8$, and for each objective pair. This measure can be interpreted as follows. When it is close to 1, the landscape has many fronts which contain few solutions. As this measure goes down to 0, all solutions tend to gather into the same front. From 1 to 0, we then except fewer, but denser, non-dominated fronts. For example, for $N = 8$, when the value is 0.01, we have around 400 fronts, each one containing 100 solutions (i.e. 0.25% of the solution space) in average. When the value is 0.05, we have around 2 000 fronts, each one containing 20 solutions (i.e. 0.5% of the solution space) in average. Roughly speaking, when an instance has less POS, it tends to have more fronts. This means that what happens for POS tend to uniformly generalize to the whole solution space. We can observe that the objective pairs $(C_{\max}, \star)$ implies less fronts than the objective pairs $(C_{\text{sum}}, \star)$. We also can notice that the instance type does not have a significant impact on the number of fronts, except for the instances of type `minella` which have less fronts than the instances of type `basseur` and `liefooghe` when considering the objective pairs $(\star, T_{\text{sum}})$.

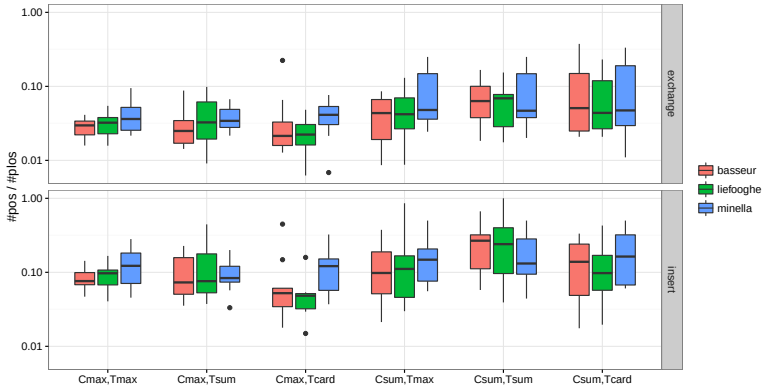

Fig. 5. Proportion of POS *per* PLOS ($N = 8$, $M = 8$).

Pareto local optimal solutions. The previous considered features are algorithm-independent in the sense that they do not rely on any particular optimization technique or operator. However, we are also interested in the multimodality, in terms of the number of local optima, of the multi-objective landscape induced by a given PFSP instance and a given neighborhood operator. Hence, we consider a feature that shall directly relate to the search process implied by a local search algorithm for solving the PFSP. Since we are dealing with permutations, we consider the following conventional neighborhood operators [10]:

- The *exchange* neighborhood, for which two solutions are neighbors iff one can be obtained from the other by exchanging two jobs at arbitrary positions.
- The *insert* neighborhood, for which two solutions are neighbors iff one can be obtained from the other by removing a job at a given position and inserting it at another position, hence shifting all other jobs in between.

For both operators, the neighborhood size is quadratic in N, although the insert neighborhood is larger by a factor of about two. Considering these two neighborhood operators, we investigate the number of Pareto local optimal solutions (PLOS) with respect to the different formulations and problem instances. A PLOS is simply a solution which is not dominated by any of its neighbors [13]. In fact, POS are also PLOS, and they are taken into account in our analysis. In Fig. 5, we show a statistic on the proportion of PLOS, computed over all the instances of the same type, for size $N = 8$ and $M = 8$. To be more precise, we report the proportion of POS divided by the proportion of PLOS, which relates global and local optima and which can be interpreted as follows. When this measure is 1, every local optima is a global optima. On the contrary, as this measure goes down to 0, the proportion of PLOS increase substantially, which eventually can be interpreted as having a problem instance which is more challenging to solve for a local search procedure. In fact, when the number of PLOS increases, the local search procedure is more likely to be trapped into local optima, and it is then eventually more difficult for the algorithm to find a new POS. The most notable observation from Fig. 5 is that, when comparing the two neighborhood operators, it appears that the exchange operator induces more difficult

landscapes, whatever the instance type and the objective pair. In fact, the insert neighborhood, which is slightly larger, induces less PLOS. This means that we expect the induced landscape to be easier to search. We can also remark that, overall, the objective pairs $(C_{\max}, \star)$ induce more PLOS than the objective pairs $(C_{\text{sum}}, \star)$, especially for instances of type `minella`. This indicates that the instances corresponding to the former shall be more difficult to solve than the ones corresponding to the latter. In particular, for the objective pairs $(C_{\max}, \star)$, instances of type `minella` have typically less PLOS, whereas there is not much difference between the instance types for $(C_{\text{sum}}, \star)$. Finally, we can observe that, apart from a few exceptions, no significant difference in the range of the number of PLOS can be reported when comparing the two other instance types.

A correlation analysis on problem features. To conclude this section, we provide in Fig. 6 a correlation analysis between the different considered features. The goal here is to study and to quantify the association between features. Notice that we consider six numerical features (N, M, objective correlation, number of POS, of fronts, of PLOS) and three categorical ones (instance types, neighborhood structures, and objective pairs). When comparing two numerical features, a Spearman nonparametric correlation coefficient is computed and reported together with a scatter plot and a simple linear regression. When the association between a numerical feature and a categorical one is considered, the density of feature values in every category is shown, as well as box-plots. Notice that, depending on the pair of features considered, the instances are mixed accordingly in Fig. 6 in order to provide a big picture summarizing and generalizing all our findings so far, of course in a more coarse-grained fashion.

Let us start with the problem size N (3rd line, 3rd column). We observe that it has no impact on the correlation between the objectives. However, when N grows, the proportional number of POS decreases, the proportional number of fronts decreases, and the proportional number of PLOS increases. More surprisingly, M (4th line, 4th column) does not show any correlation with the other features. Next, when the objective correlation grows (5th line, 5th column), the proportional number of POS decrease, the proportional number of fronts increase, and the number of PLOS decreases. When the number of POS grows (6th line, 6th column), the proportional number of fronts grows. Although this might seem surprising at first sight, we actually argue that this is an artifact due of mixing instances with different $N-$values, as one can guess by observing the clustering effect in the scatter plot. Actually, when considering each $N-$value separately, the proportional number of fronts slightly decreases. In addition, we can see that the larger the number of POS, the proportionally fewer the number of PLOS. As for the proportional number of non-dominated fronts (7th line, 7th column), it is negatively correlated with the number of PLOS.

This correlation analysis indicates that there is a relatively high interaction between some of the features. We hypothesize that they are actually impactful in terms of instance difficulty. In the following, we consider to effectively study the association between problem features and instance difficulty by experimentally appreciating the ability of a dedicated algorithm class to solve PFSP instances.

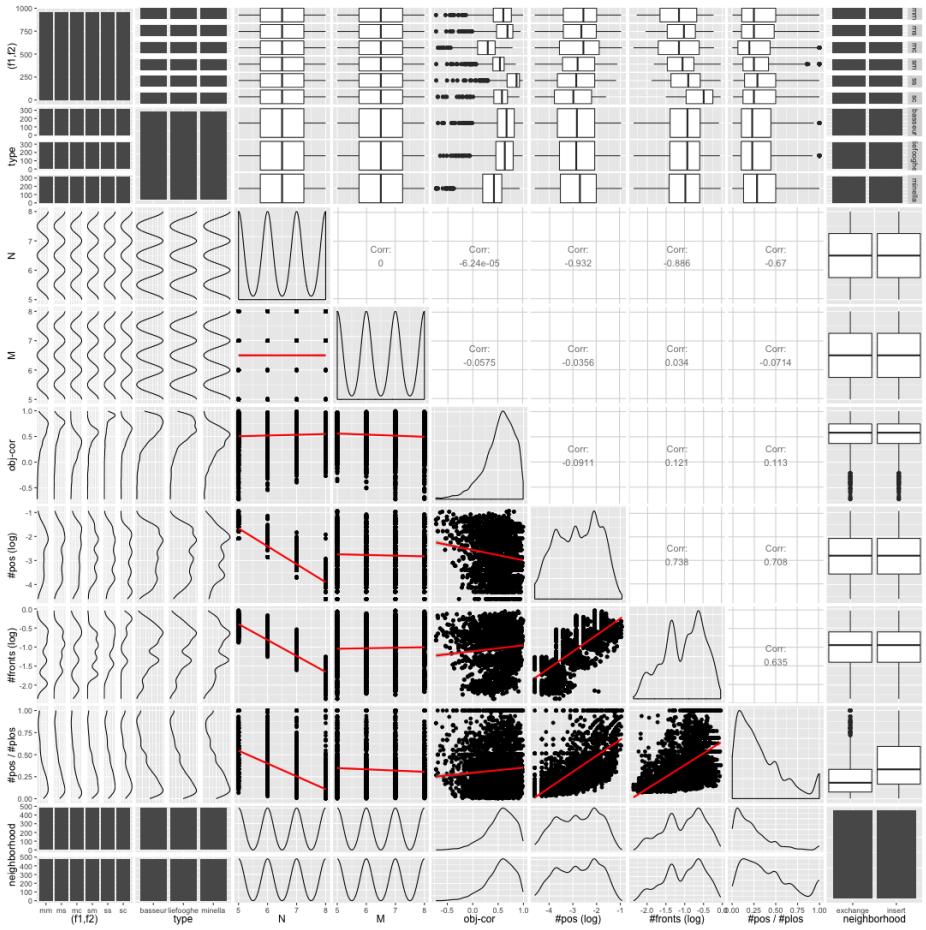

Fig. 6. Association matrix between each pair of problem features. For the objective pair (f_1, f_2), the notation `mm` (resp. `ms`, `mc`, `sm`, `ss` and `sc`) stands for $(C_{\max}, T_{\max})$ (resp. $(C_{\max}, T_{\text{sum}})$, $(C_{\max}, T_{\text{card}})$, $(C_{\text{sum}}, T_{\max})$, $(C_{\text{sum}}, T_{\text{sum}})$ and $(C_{\text{sum}}, T_{\text{card}})$).

4 Problem features vs. algorithm performance

In order to elicit the impact of the previous features on the performance of solving a given PFSP instance, we consider the so-called Pareto Local Search (PLS) algorithm [13]. PLS has proved extremely relevant to solve different combinatorial MOPS, including the PFSP [6, 8, 12]. We first recall its algorithmic components and the way it is instantiated and experimented for the PFSP.

Algorithm description and performance assessment. PLS maintains an unbounded archive A of mutually non-dominated solutions. This archive is initialized with one (random) initial solution from the solution space. At each iteration, one (unvisited) solution is selected at random from the archive $x \in A$. All neighboring solutions $\mathcal{N}(x)$ from x are then exhaustively evaluated. This of course assumes given a neighborhood relation. The non-dominated solutions

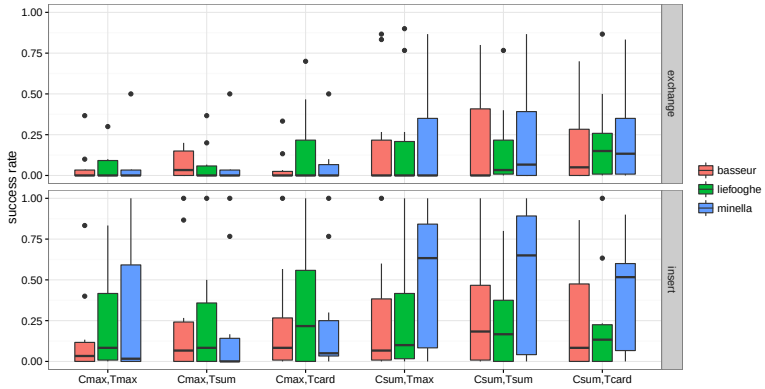

Fig. 7. Success rate of PLS ($N = 8$, $M = 8$).

from $A \cup \mathcal{N}(x)$ are stored in the archive, and the current solution x is then flagged as `visited` to avoid a useless revaluation of its neighborhood. The algorithm naturally stops once all solutions from the archive are flagged as `visited`. PLS always terminates and returns a maximal Pareto local optimum set [13]. In this work, we experiment PLS using the two previously-defined neighborhood operators, namely *exchange* and *insert*, hence ending with two PLS variants. To experiment the so-obtained variants and their relative performance, we consider to run them on the considered instances (30 independent runs for each instance), and to compute the success rate, that is the proportion of runs where PLS is able to identify the Pareto set. The success rate can be interpreted as the empirical probability of solving a given instance to optimality. Notice that, since the considered instances are of small size, this measure is fully accurate for the purpose of our study by avoiding the biases that other performance indicators and their corresponding parameters could introduce. Before studying the impact of problem features, we start by summarizing the main empirical observations with respect to the success rate of PLS on the different instance classes.

Exploratory analysis. In Fig. 7, we report the success rate of both PLS variants for instances of size $N = 8$ and $M = 8$. The first notable observation is that, independently of the instance type or neighborhood, the success rate is lower for the objective pairs $(C_{\max}, \star)$ than for $(C_{\text{sum}}, \star)$. This is in accordance with our observations on the number of PLOS, where the former objective pairs were intuited as more difficult to solve. Similarly, we can observe that the considered instances are overall less difficult to solve for PLS when using the insert neighborhood than when using the exchange neighborhood, which we can directly relate to our observation about PLOS. Actually, the objective pairs $(C_{\text{sum}}, \star)$ for the instances of type `minella` are relatively easy to solve when using the insert neighborhood, with a median success rate above 50%. On the contrary, the instances of type `basseur` seem to provide the overall lowest success rate, which indicates that they are the most challenging for PLS. Over the objective pairs $(C_{\max}, \star)$, the instances of type `liefooghe` are found to be relatively easier to solve than `minella`, whereas this is not the case for $(C_{\text{sum}}, \star)$.

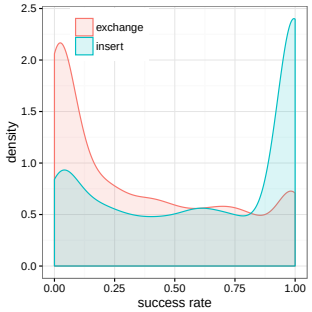

Fig. 8. Density of success rate values over all the instances.

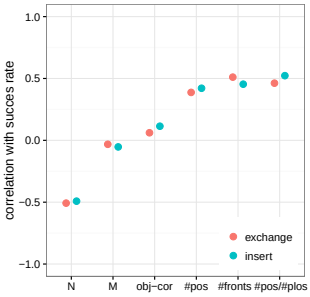

Fig. 9. Non-parametric correlation between problem features and success rate.

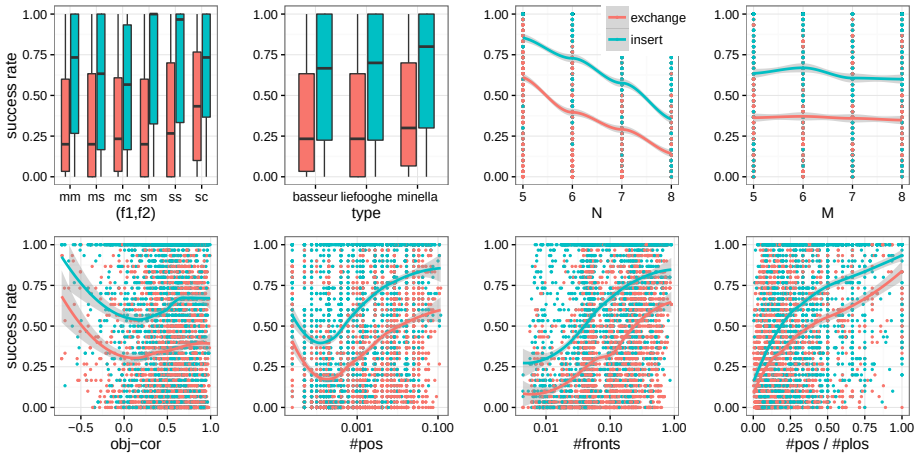

Fig. 10. Scatterplot and locally-fitted polynomial function (LOESS) between problem features and success rate.

Correlation analysis. In the following, we want to understand the relation between the problem features and the success rate of PLS over all the instances. We here focus on this association *across* instance classes. In order to measure this association *within* instance classes, we rather suggest to follow a mixed model analysis such as [5]. In Fig. 8, we can first appreciate the distribution of success rates for both PLS variants over all the instances. Basically, we can observe many failures when using the exchange neighborhood, and relatively more success when using the insert neighborhood. In Figs. 9–10, we report the correlation between the success rate and the considered features. Although the two operators have different behavior in terms of the distribution of their success rate (Fig. 8), we can now observe that their performance is correlated in the same manner with the considered features. In particular, we found a relatively high correlation of success rate with the problem size N, as well as with the number of fronts, of POS, and of PLOS. Interestingly, the correlation with the pair of objectives is rather low, which indicates that the performance of PLS is more impacted by the underlying structure of the PFSP fitness landscape, indepen-

dently of the nature of the objectives to be optimized. Similarly, the success rate is not impacted by the instance type, which could seem surprising at a first sight. We attribute this to the relatively high variance of success rate across all the instances of same type. Unsurprisingly, the instance difficulty increase with N, since we can see that the success rate decreases with N for both operators. However, the difficulty stays constant whatever the value of M, which is to be related to the very low correlation that was found previously between M and the other considered features. When examining how the performance relates with the objective correlation, we can see that the expected success rate estimation follows a more complicated trend. This is also the reason why the correlation between objective correlation and success rate is low in Fig. 9, i.e. the local regression curve does not follow a monotonic function but rather has a U-shape. Notice also that PFSP instances tend to be easier when the objectives are conflicting (although we only observed few instances in such cases, so that the confidence interval is large), or on the contrary highly correlated. However, when the objective correlation is close to zero, the success rate is smaller for both PLS variants. As for the correlation with the proportion of POS, the general trend of the local regression curve clearly indicates that the success rate increases as the number of POS grows (this also holds when the proportion of POS is very low, that is when the absolute number of POS is 1). This might be explained as follows. When the number of non-dominated solutions is large, the archive maintained by PLS is typically large. Hence, the probability to get trapped in a Pareto local optimum set [13] decreases, so that PLS is able to run longer and to find better approximation sets. This means that it is easier for PLS to find the whole Pareto front when it is larger. This is to be mitigated by the runtime required to effectively find all optimal solutions, which we do not report in this paper due to space restriction. Actually, we observed that the runtime of PLS is much larger when the number of POS is large. At last, as expected, when the proportional number of PLOS increases, PFSP instances are typically more difficult, and then the success rate of PLS decreases, independently of the neighborhood operator.

Relative importance. The correlation analysis provided in the previous section allowed to us to highlight the individual impact and effect of each feature on the performance of PLS. However, this does not tell us the combined effect of those features. In the following, we investigate this issue by considering a machine learning perspective for classification. More specifically, we consider the two following complementary scenarios: (i) predicting the degree of difficulty of an instance for each algorithm variant, and (ii) predicting which algorithm variant (PLS with insert or PLS with exchange) performs better for a given instance. For each scenario, a classification model based on random forests (RF) is first constructed using the different considered features as input variables. RF is a state-of-the-art ensemble learning method based on the construction of multiple decision trees that outputs the mode of the classes of the individual trees [4]. In the first scenario, the output of the RF model is the class of difficulty of a given instance. Given the distribution of success rates depicted in Fig. 8, we divided PFSP instances into three classes: easy instances for which the success

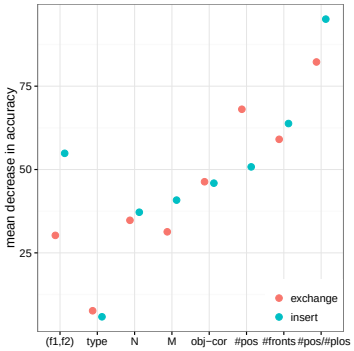

Fig. 11. Variable importance from RF classification model for scenario #1.

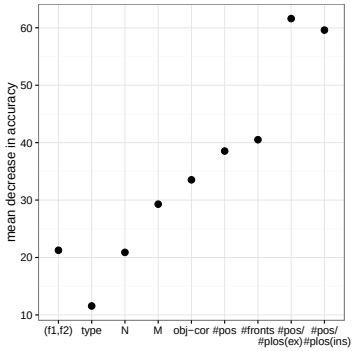

Fig. 12. Variable importance from RF classification model for scenario #2.

rate is 1 (265 for exchange, 1134 for insert), hard instances for which the success rate is 0 (657 for exchange, 243 for insert), moderate instances for which the success rate is in-between 0 and 1 (1958 for exchange, 1503 for insert). The estimate of error rate of the corresponding RF model is 22.33% for exchange, and 26.56% for insert. In the second scenario, the output class of the RF model is simply whether PLS with insert has a better success rate than PLS with exchange for a given instance. In case both variants have the same success rate, the one which evaluated the smallest number of solutions in average is said to be better. PFSP instances are then divided into two classes: 808 instances for which PLS-exchange is better, and 2072 instances for which PLS-insert is better. The estimate of error rate of the corresponding RF model is 23.40%.

Interestingly, RF has the ability to render the relative importance of each input variable (here, features), given in terms of the mean decrease in accuracy [4]. This measure is depicted in Fig. 11 for the first scenario and in Fig. 12 for the second one. We clearly see that the most important feature relates to the proportional number of PLOS. This emphasizes the high impact of multimodality on the difficulty of PFSP instances and on the performance of multi-objective local search. It is also interesting to remark that the next important features are those of general-purpose nature which were specifically investigated in the paper. In fact, the number of POS, the number of fronts and the objective correlation are all found to be more important than the other features that relates directly to the benchmark parameters, namely N, M, the pair of objectives (with the exception of scenario 1 for the insert neighborhood), and the instance type.

5 Conclusions

The multi-objective fitness landscape analysis conducted in this paper allowed us to comprehensively relate the characteristics of bi-objective PFSP instances with the sources of difficulty faced by Pareto local search. From the methodological point of view, we argue that the investigated features and the statistical tools involved all along the paper revealed to be highly valuable in order to accurately harness the complexity of a multi-objective search process, in particular by emphasizing the crucial importance of multimodality. It is our hope that the

approach adopted in this paper, as well as our findings regarding the PFSP, will enable to tackle other challenging issues, both from the algorithm design perspective and the more practical problem solving perspective. In fact, leveraging our analysis to other multi-objective scheduling problems and algorithms is still an open challenge in terms of scalability (e.g., large-size instances with more than two objectives, low-cost feature computation or estimation). Tightly related to this issue, a particularly challenging research path would be to set up a sound and complete methodology for algorithm performance prediction, with the ultimate goal of selecting the most appropriate ones for a given instance.

References

1. Aguirre, H., Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes. Eur J Oper Res 181(3), 1670–1690 (2007)
2. Basseur, M., Liefooghe, A.: Metaheuristics for biobjective flow shop scheduling. In: Metaheuristics for Production Scheduling, chap. 9, pp. 225–252. Wiley (2013)
3. Basseur, M., Seynhaeve, F., Talbi, E.G.: Design of multi-objective evolutionary algorithms: application to the flow shop scheduling problem. In: CEC 2002. pp. 1151–1156. NJ, USA (2002)
4. Breiman, L.: Random forests. Mach Learn 45(1), 5–32 (2001)
5. Daolio, F., Liefooghe, A., Verel, S., Aguirre, H., Tanaka, K.: Problem features vs. algorithm performance on rugged multi-objective combinatorial fitness landscapes. Evolut Comput (to appear)
6. Dubois-Lacoste, J., López-Ibáñez, M., Stützle, T.: A hybrid TP+PLS algorithm for bi-objective flow-shop scheduling problems. Comput Oper Res 38(8), 1219–1236 (2011)
7. Ehrgott, M.: Multicriteria optimization. Springer (2005)
8. Geiger, M.: On operators and search space topology in multi-objective flow shop scheduling. Eur J Oper Res 181(1), 195–206 (2007)
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
10. Hoos, H., Stützle, T.: Stochastic Local Search: Foundations and Applications. Morgan Kaufmann (2004)
11. Liefooghe, A., Basseur, M., Jourdan, L., Talbi, E.G.: Combinatorial optimization of stochastic multi-objective problems: an application to the flow-shop scheduling problem. In: EMO 2007. LNCS, vol. 4403, pp. 457–471. Matsushima, Japan (2007)
12. Minella, G., Ruiz, R., Ciavotta, M.: A review and evaluation of multiobjective algorithms for the flowshop scheduling problem. INFORMS J Comput 20(3), 451–471 (2008)
13. Paquete, L., Schiavinotto, T., Stützle, T.: On local optima in multiobjective combinatorial optimization problems. Ann Oper Res 156(1), 83–97 (2007)
14. Richter, H., Engelbrecht, A. (eds.): Recent Advances in the Theory and Application of Fitness Landscapes. Springer (2014)
15. Taillard, E.D.: Benchmarks for basic scheduling problems. Eur J Oper Res 64, 278–285 (1993)
16. T'Kindt, V., Billaut, J.C.: Multicriteria Scheduling: Theory, Models and Algorithms. Springer (2005)
17. Verel, S., Liefooghe, A., Jourdan, L., Dhaenens, C.: On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. Eur J Oper Res 227(2), 331–342 (2013)