Fitness Landscape Analysis of the Permutation Flowshop Scheduling Problem with Total Flow Time Criterion

Marco Baioletti and Valentino Santucci

Department of Mathematics and Computer Science
University of Perugia, Italy
{marco.baioletti,valentino.santucci}@unipg.it

Abstract. This paper provides a fitness landscape analysis of the Permutation Flowshop Scheduling Problem considering the Total Flow Time criterion (PFSP-TFT). Three different landscapes, based on three neighborhood relations, are considered. The experimental investigations analyze aspects such as the smoothness and the local optima structure of the landscapes. To the best of our knowledge, this is the first landscape analysis for PFSP-TFT.

1 Introduction and Related Work

The Permutation Flowshop Scheduling Problem (PFSP) is a scheduling problem widely encountered in areas such as manufacturing and large scale products fabrication [13]. The goal of PFSP is to determine the best permutation $\pi = \langle \pi[1], \ldots, \pi[n] \rangle$ of n jobs that have to be processed through a sequence of m machines. Preemption and job-passing are not allowed. Here we focus on the Total Flow Time (TFT) criterion which consists in minimizing the objective function

$$f(\pi) = \sum_{j=1}^{n} c(m, \pi[j]), \tag{1}$$

where $c(i, \pi[j])$ is the completion time of job $\pi[j]$ on machine i and is recursively calculated in terms of the processing times $p_{i,\pi[j]}$ as

$$c(i, \pi[j]) = \begin{cases} p_{i,\pi[j]} & \text{if } i = j = 1 \\ p_{i,\pi[j]} + c(i, \pi[j-1]) & \text{if } i = 1 \text{ and } j > 1 \\ p_{i,\pi[j]} + c(i-1, \pi[j]) & \text{if } i > 1 \text{ and } j = 1 \\ p_{i,\pi[j]} + \max\{c(i, \pi[j-1]), c(i-1, \pi[j])\} & \text{if } i > 1 \text{ and } j > 1 \end{cases} \tag{2}$$

The PFSP with the TFT criterion has been demonstrated to be NP hard for two or more machines [7][19]. Therefore, even due to its practical interest, many researches have been devoted to finding high quality and near optimal solutions by means of heuristic or meta-heuristic approaches, for instance with evolutionary algorithms [5][7][22].

Meta-heuristic techniques navigate the fitness landscape of the instance at hand. A fitness landscape is a triple $(S, \mathcal{N}, f)$ where: S is the set of solutions, $\mathcal{N}$ is a neighborhood relation among the solutions in S, and f is the objective/fitness function to optimize.

The fitness landscape can be analyzed by studying several features of interest [18][24][28]. These analyses are also important to understand the behavior of meta-heuristics and evolutionary algorithms [3][16] applied to combinatorial optimization problems [1][6].

A first aspect concerns the neutrality of the landscape and in general the classification of the solutions/points according to the fitness differences among the neighbors. There exists seven types of points: SLMIN, LMIN, IPLAT, SLOPE, LEDGE, LMAX and SLMAX. They are described in Figure 1 [15]. A point P is of type IPLAT if all its neighbors have the same fitness values as P. A point is of type SLMAX (SLMIN) if it is a strictly local maximum (minimum), while the types LMAX and LMIN are non strict local maxima (or minima). A point P is a SLOPE if some of its neighbors have a greater fitness value and some other ones have a lower value than P. Finally, a LEGDE point P has also some neighbors with the same fitness values as P. The distribution of point types gives a quantitative analysis about the neutrality of the fitness landscape. In particular, a fitness landscape is more or less neutral according to the higher or lower percentage of IPLAT, LEGDE, LMAX and LMIN.

A second feature is to study the landscape correlation analysis [27]. Its main tool is the autocorrelation coefficient $\rho(1)$ which quantifies how much the fitness values of neighbor solutions are related to each other. Higher values for $\rho(1)$ mean that the landscape is smooth, while a small value means a rugged landscape.

A third analysis tries to check the presence of the so called "big-valley" hypothesis [14], i.e. if local minima of good quality are clustered and surround the global minimum point. Some of the combinatorial optimization problems have this structure and this is a positive characteristic for search algorithms, because it is easier to search the global optimum.

Another point that has been studied is the distribution of local minima and their distance from the global minimum [17]. Again, the distribution and the distance from local and global minima is strictly related to the performances of meta-heuristic algorithms.

Finally, another aspect of interest is the concept of fitness barrier [24], i.e. the number of steps necessary for exiting from a local minimum and finding a better point. A search space with a low fitness barrier can be better explored by a meta-heuristic algorithm because there are less chances that it can be trapped in a local minima.

In our case, S is the set of all the n-length permutations of jobs, and f is the total flow time objective of equation (1). Moreover, three fitness landscapes generated by three different neighborhood relations are considered, namely: the adjacent swap (ASW), interchange (INT) and insert (INS) neighborhoods [2][23]. ASW and INT neighbors are obtained by swapping two items in a permutation. While for INT neighbors no restriction is considered on the items to swap, the

ASW neighborhood limits the swap to adjacent items. Finally, the INS neighborhood is obtained by removing an item and inserting back in a different position. To the best of our knowledge this is the first analysis conducted on PFSP-TFT.

The rest of the paper is organized as follows. Section 2 briefly describe the experimental setting used throughout the paper. A classification of the different type of solutions is provided in Section 3 together with an analysis of the landscape neutrality. Section 4 analyze the smoothness of the spaces. Sections 5 and 6 provide a study of the local minima distribution, while Section 7 discusses some aspects about the characteristics of the basins of attraction. Finally, conclusions are drawn in Section 8

2 Experimental Setting

The experiments were held over a suite of 120 PFSP-TFT instances: ten problem instances for each $n \times m$ configuration, with $n \in \{10, 20, 50, 100\}$ and $m \in \{5, 10, 20\}$. The $n = 20, 50, 100$ instances were taken from the widely known benchmark suite of Taillard [25], while the 10 jobs instances were randomly generated using the same scheme of [25].

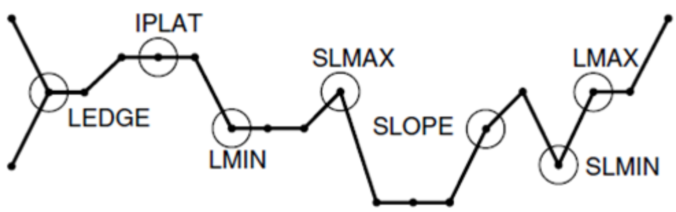
When the experimental analyses involve local minima, these have been exactly located through exhaustive enumeration for $n = 10$ instances, while for the larger instances they were collected by means of 2 000 local searches (each one starting from a randomly generated seed solution) performed using the "best improvement" strategy.

Finally, while the global optima of the 10 jobs instances were exactly identified, for the other instances we have considered the best known solutions taken from the state-of-the-art results reported in recent works [19][22].

3 Position Type Distributions and Neutrality Analysis

In this section we provide a classification of the position types [15] for the three search spaces.

Fig. 1: Position Types



The data in Table 1 have been obtained by a complete enumeration of the $n!$ solutions in the search space. Due to the cardinality, $n = 10$ has been considered.

Table 1: Position Type Distributions in Percentage Values

$n \times m$	Neigh.	SLMIN	LMIN	SLOPE	LEDGE	IPLAT	LMAX	SLMAX
	ASW	0.41	0.05	89.85	9.43	0.00	0.06	0.20
10×5	INT	< 0.01	< 0.01	80.28	19.71	0.00	< 0.01	< 0.01
	INS	< 0.01	< 0.01	76.87	23.13	0.00	< 0.01	< 0.01
	ASW	0.55	0.05	93.91	5.07	0.00	0.04	0.38
10×10	INT	< 0.01	< 0.01	87.93	12.07	0.00	< 0.01	< 0.01
	INS	< 0.01	< 0.01	81.27	18.72	0.00	< 0.01	< 0.01
	ASW	0.48	0.03	94.26	4.76	0.00	0.04	0.42
10×20	INT	< 0.01	< 0.01	89.72	10.28	0.00	< 0.01	< 0.01
	INS	< 0.01	< 0.01	84.38	15.62	0.00	< 0.01	< 0.01

Table 1 shows that the vast majority of the points are SLOPE and LEDGE points. Other types are very rare and, in particular, IPLAT points are not present in none of the three search spaces.

SLOPEs are more frequent in the ASW search space. Moreover, it looks that the quantity of SLOPEs increases with m. Therefore, the neutrality of the landscape is very likely to decrease when the number of machines increases.

The study of position types for $n > 10$ is possible by a random uniform sampling of the solutions in the search space. The results of this investigation are presented in Table 2. Note that, since the sampling have produced only SLOPE and LEDGE points, Table 2 only shows the percentage of SLOPEs. The remaining percentage are LEDGEs.

Table 2: Percentage of SLOPE points

$n \times m$	ASW	INT	INS
20×5	81.89	58.30	48.40
20×10	85.10	70.85	57.15
20×20	92.09	76.15	58.10
50×5	49.52	8.45	7.55
50×10	58.67	22.10	12.40
50×20	78.41	37.90	16.15
100×5	24.70	0.10	0.20
100×10	31.51	2.05	0.35
100×20	49.79	7.05	1.40

Under random sampling, Table 2 shows that the number of SLOPEs increases with m and decreases with n. For (relatively) small m and large n, LEDGE points are more frequent than SLOPEs, thus denoting that the neutrality degree is very likely to increase with n. This phenomenon is even stronger for the INT and INS neighborhoods. In these cases, as n increases, the number of SLOPEs drastically decreases. In particular, when $n = 100$, almost all the sampled points are LEDGEs.

In Table 3, we show the neutrality degrees [15], i.e. the number of the average percentage of neighbors with the same fitness. The percentage is computed with respect to the neighborhood size. For each configuration $n \times m$, we report the minimum and the maximum values. Data for $n = 10$ have been obtained by a complete enumeration, while for $n \in \{20, 50, 100\}$ a random uniform sampling of $N = 10\,000$ points has been drawn.

Table 3: Average Neutrality Degree per Solution in percentage values

$n \times m$	ASW	INT	INS
10×5	$[1.07, 1.10]$	$[0.44, 0.48]$	$[0.31, 0.36]$
10×10	$[0.59, 0.62]$	$[0.27, 0.30]$	$[0.24, 0.28]$
10×20	$[0.55, 0.59]$	$[0.22, 0.26]$	$[0.20, 0.22]$
20×5	$[5.34, 5.41]$	$[0.14, 0.43]$	$[0.34, 0.37]$
20×10	$[5.31, 5.35]$	$[0.15, 0.36]$	$[0.32, 0.34]$
20×20	$[5.30, 5.32]$	$[0.11, 0.17]$	$[0.31, 0.33]$
50×5	$[2.12, 2.18]$	$[0.17, 0.29]$	$[0.08, 0.09]$
50×10	$[2.09, 2.12]$	$[0.10, 0.15]$	$[0.07, 0.08]$
50×20	$[2.07, 2.08]$	$[0.074, 0.10]$	$[0.06, 0.07]$
100×5	$[1.09, 1.13]$	$[0.14, 0.22]$	$[0.03, 0.04]$
100×10	$[1.04, 1.08]$	$[0.12, 0.19]$	$[0.02, 0.03]$
100×20	$[0.99, 1.04]$	$[0.10, 0.15]$	$[0.01, 0.02]$

These results clearly confirm the previous indications about the neutrality of the search spaces.

4 Landscape Correlation

Here, we study the autocorrelation coefficients and the correlation lengths for the three search spaces [27].

For each problem instance considered, a random walk of $N = 500\,000$ steps has been performed and the N visited solutions, together with their fitness values, have been registered. Then, the autocorrelation $\rho(1)$ and the correlation length $l = (\ln(|\rho(1)|))^{-1}$ have been computed. Intuitively, $\rho(1)$ measures the statistical correlation between neighboring solutions, while the correlation length gives an indication of how far it is possible to go without incurring in a steep descend or ascend.

The results of the experiment have been aggregated for every $n \times m$ configurations and shown in Table 4.

Table 4 shows that the autocorrelation is very large and becomes close to 1 when n increases. As described in [27], this is an evidence that the three spaces are very smooth, i.e., neighboring solutions tend to have similar fitness values [8]. This aspect is very important because search algorithms [10][11][12] are, in general, positively affected by the smoothness of the underlying fitness landscape.

Table 4: Autocorrelations and Correlation Lengths

$n \times m$	**Autocorr.**			**Corr. Lengths**		
	ASW	**INT**	**INS**	**ASW**	**INT**	**INS**
10 × 5	0.92	0.74	0.81	12.49	3.26	4.77
10 × 10	0.92	0.73	0.81	11.79	3.25	4.68
10 × 20	0.93	0.74	0.82	15.74	3.40	5.03
20 × 5	0.97	0.86	0.91	37.16	6.52	10.63
20 × 10	0.96	0.85	0.90	25.87	6.31	9.79
20 × 20	0.96	0.85	0.90	22.76	6.15	9.41
50 × 5	0.99	0.94	0.96	160.34	15.74	27.54
50 × 10	0.99	0.93	0.96	97.75	14.20	24.01
50 × 20	0.99	0.93	0.96	70.49	14.16	22.94
100 × 5	1.00	0.97	0.98	640.95	29.88	53.67
100 × 10	1.00	0.96	0.98	272.42	26.92	47.28
100 × 20	0.99	0.96	0.98	144.18	24.46	41.60

5 Fitness-Distance Analysis of the Local Minima

As known in literature, evidence of the "big-valley" structure for a given fitness landscape can be inferred by showing that the fitness-distance correlation [26] of the local minima is large enough. A value larger than 0.15 is generally accepted as threshold [14][15].

For each $n \times m$ configuration and for each neighborhood relation, Table 5 provides the minimum and maximum correlation coefficients between the fitness and the distance of the local minima to the presumed global minimum. Moreover, also the number of instances presenting a correlation larger than 0.15 is shown.

Table 5: Fitness-Distance Correlations

$n \times m$	**ASW**		**INT**		**INS**	
	fdc	#corr.inst.	fdc	#corr.inst.	fdc	#corr.inst.
10 × 5	$[0.30, 0.85]$	10/10	$[0.33, 0.70]$	10/10	$[0.33, 0.97]$	10/10
10 × 10	$[0.32, 0.75]$	10/10	$[0.29, 0.69]$	10/10	$[0.49, 0.88]$	10/10
10 × 20	$[0.46, 0.84]$	10/10	$[0.33, 0.73]$	10/10	$[0.57, 1.00]$	10/10
20 × 5	$[0.39, 0.83]$	10/10	$[0.25, 0.52]$	10/10	$[0.19, 0.50]$	10/10
20 × 10	$[0.48, 0.71]$	10/10	$[0.25, 0.52]$	10/10	$[0.03, 0.39]$	**7/10**
20 × 20	$[0.45, 0.69]$	10/10	$[0.20, 0.51]$	10/10	$[0.17, 0.51]$	10/10
50 × 5	$[0.47, 0.68]$	10/10	$[0.10, 0.30]$	**7/10**	$[0.07, 0.19]$	**4/10**
50 × 10	$[0.46, 0.59]$	10/10	$[0.06, 0.36]$	**4/10**	$[0.05, 0.18]$	**1/10**
50 × 20	$[0.44, 0.55]$	10/10	$[0.09, 0.34]$	**5/10**	$[0.06, 0.25]$	**3/10**
100 × 5	$[0.50, 0.64]$	10/10	$[0.04, 0.21]$	**1/10**	$[-0.01, 0.13]$	**0/10**
100 × 10	$[0.46, 0.59]$	10/10	$[-0.02, 0.20]$	**3/10**	$[-0.06, 0.14]$	**0/10**
100 × 20	$[0.37, 0.54]$	10/10	$[0.00, 0.13]$	**0/10**	$[-0.09, 0.20]$	**1/10**

These results show that the "big-valley" hypothesis is clearly more evident for the ASW neighborhood than for INT and INS. Moreover, in the latter cases, the number of correlated instances seems to decrease when n increases[1]. Finally, the number of machines m does not seem to influence the correlation coefficients.

6 Distribution of the Local Minima

The distribution of the local minima has been investigated by means of two experiments. The first aims at estimate the centrality of the global minimum with respect to the other local minima, while the second goes in the opposite direction and analyzes the distances of the local minima from the global minimum.

The optimum centrality is defined as the percentage value oc defined as

$$oc = 100 \cdot \frac{d_{lmin} - d_{gmin}}{d_{gmin}}, \tag{3}$$

where d_{lmin} and d_{gmin} are, respectively, the average pairwise distance among the local minima and the average distance from the nearest global minimum. The measure oc estimates how much the global minimum is centrally located with respect to the other local minima [9]. A particular distinction has to be made between positive (i.e., $d_{lmin} > d_{gmin}$) and negative (i.e., $d_{lmin} < d_{gmin}$) values. For each $n \times m$ configuration and for each neighborhood relation, Table 6 reports the minimum and maximum oc values together with the number of instances where $oc > 0$.

Table 6: Global Optimum Centrality

$n \times m$	ASW		INT		INS	
	oc	$\# oc > 0$	oc	$\# oc > 0$	oc	$\# oc > 0$
10×5	$[5.55, 34.44]$	10/10	$[-28.97, -15.91]$	**0/10**	$[-6.92, 40.74]$	**9/10**
10×10	$[-5.44, 26.83]$	**9/10**	$[-34.89, -22.55]$	**0/10**	$[4.69, 23.38]$	10/10
10×20	$[15.21, 34.61]$	10/10	$[-37.89, -18.83]$	**0/10**	$[-0.79, 100.00]$	**9/10**
20×5	$[0.00, 7.20]$	10/10	$[1.18, 8.06]$	10/10	$[0.23, 10.10]$	10/10
20×10	$[-0.94, 4.41]$	**9/10**	$[0.30, 5.99]$	10/10	$[-1.28, 6.82]$	**6/10**
20×20	$[0.63, 4.83]$	10/10	$[1.92, 7.50]$	10/10	$[0.21, 5.70]$	10/10
50×5	$[0.64, 4.01]$	10/10	$[0.67, 2.03]$	10/10	$[-0.08, 1.59]$	**9/10**
50×10	$[-0.83, 1.89]$	**8/10**	$[0.48, 1.67]$	10/10	$[-0.18, 1.44]$	**9/10**
50×20	$[-0.28, 2.20]$	**8/10**	$[0.33, 1.61]$	10/10	$[-0.06, 1.26]$	**9/10**
100×5	$[0.83, 2.88]$	10/10	$[-0.01, 0.31]$	**9/10**	$[-0.58, 0.47]$	**7/10**
100×10	$[-0.76, 1.46]$	**6/10**	$[0.06, 0.74]$	10/10	$[-0.57, 0.65]$	**9/10**
100×20	$[-0.71, 0.79]$	**5/10**	$[-0.09, 1.05]$	**7/10**	$[-0.20, 0.52]$	**7/10**

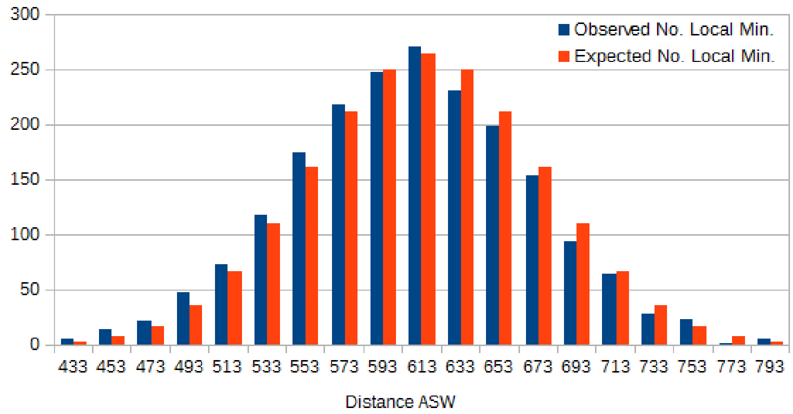
Table 6 shows that, except the INS neighborhood in the 10-jobs instances, only few oc values are negative. Moreover negative values have a very low magni-

[1] Note that the results of the INS neighborhood on the 10-jobs instances, due to the very small number of local minima, are not too much significant.

tude. This suggests us to conclude that the global optimum tends to be located inside the region of the local optima. Moreover, oc tends to decreases when n increases. Therefore, it is likely that the global optimum tends to “move” towards the borders of the local optima region when n increases.

The opposite analysis has been conducted by comparing the observed distribution of the local minima distances from the global minimum with respect to their expected distribution in the case that they were randomly sampled. Figures 2a, 2b, and 2c report the observed (blue bars) and expected (red bars) distributions for the first instance of the 50×10 problems, respectively, for ASW, INT, and INS (the other instances seems to have a similar behaviour).

Fig. 2: Local Minima Distribution w.r.t. Global Min. for ASW



These results show that the observed distribution is approximately similar to the expected one, thus local minima look like as they are randomly sampled from the set of all the permutations.

Furthermore, for ASW and INS the observed minima are a bit more than the estimated ones at the left of the distribution mode. The opposite happen for INT.

7 Fitness Barriers

In this section we study the fitness barriers [24], restricted to the ASW search space, which is the basis for the differential mutation operator described in [4][20][21].

For every instance ($n = 20, 50, 100$ and $m = 5, 10, 20$), we have found $N = 10\,000$ local minima by means of a simple local search based on adjacent swap

Fig. 3: Local Minima Distribution w.r.t. Global Min. for INT

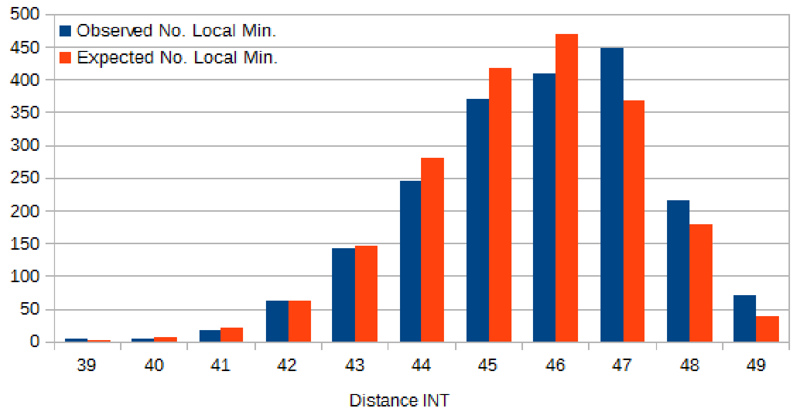
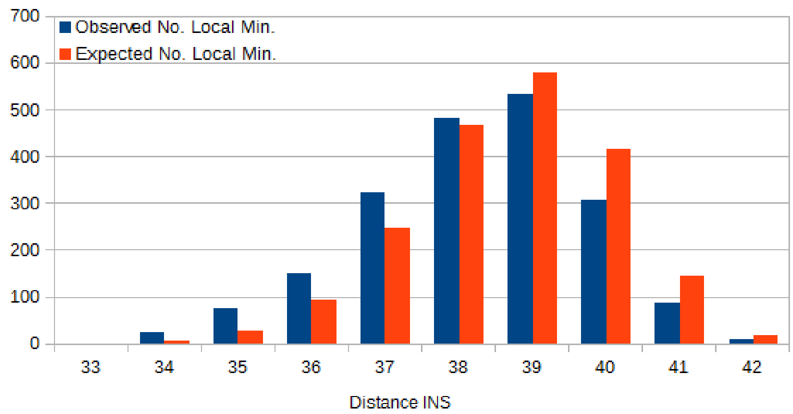


Fig. 4: Local Minima Distribution w.r.t. Global Min. for INS



moves. For each local minimum x, we have found, through a breadth-first search, the closest point y which has a smaller fitness, i.e., $f(y) < f(x)$. We have then considered two variables: the length L of the shortest path from x to y, and the largest fitness value H of the points in the path. Starting from H, we have computed the relative variation $R = \frac{H-f(x)}{f(x)}$.

L indicates how many steps one must go away from a local minimum in order to escape from its attraction basin. H is the barrier height and denotes which is the highest point one must pass through in order to find a better local minimum.

L has very low variability and, in all the experiments, presents an average value very close to 3. This means that very often it is sufficient to walk for three steps from a local minimum in order to find a better point.

The analysis of R shows that, in order to escape from a local minimum, it is sufficient to accept values which are around 1% (for $n = 20$), around 0.4% (for $n = 50$) and around 0.2% (for $n = 100$) worse than the minimum.

8 Conclusion

In this paper we have investigated the structure of PFSP-TFT instances. Three different landscapes have been considered by analyzing the three most popular neighborhood relations for PFSP solutions.

The experiments provide indications on the smoothness and the local optima structure of the landscapes. The main result regards the evidence of a "big valley" structure on the distribution of the local optima. With this regard, also a new measure of optimum centrality has been introduced. Moreover, depending on the neighborhood considered, different indications have emerged.

References

1. Baioletti, M., Milani, A., Santucci, V.: Algebraic particle swarm optimization for the permutations search space. In: IEEE Congress on Evolutionary Computation CEC 2017 (in press)
2. Baioletti, M., Busanello, G., Vantaggi, B.: Acyclic directed graphs representing independence models. International Journal of Approximate Reasoning 52(1), 2 – 18 (2011), http://dx.doi.org/10.1016/j.ijar.2010.09.005
3. Baioletti, M., Chiancone, A., Poggioni, V., Santucci, V.: Towards a new generation aco-based planner. In: LNCS - Proc. of ICCSA 2014. pp. 798–807 (2014), http://dx.doi.org/10.1007/978-3-319-09153-2_59
4. Baioletti, M., Milani, A., Santucci, V.: Linear ordering optimization with a combinatorial differential evolution. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics, Kowloon Tong, Hong Kong, October 9-12, 2015. pp. 2135–2140 (2015), http://dx.doi.org/10.1109/SMC.2015.373
5. Baioletti, M., Milani, A., Santucci, V.: A discrete differential evolution algorithm for multi-objective permutation flowshop scheduling. Intelligenza Artificiale 10(2), 81–95 (2016), http://dx.doi.org/10.3233/IA-160097

6. Baioletti, M., Milani, A., Santucci, V.: An extension of algebraic differential evolution for the linear ordering problem with cumulative costs. In: Parallel Problem Solving from Nature - PPSN XIV - 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings. pp. 123–133 (2016), http://dx.doi.org/10.1007/978-3-319-45823-6_12
7. Ceberio, J., Irurozki, E., Mendiburu, A., Lozano, J.A.: A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem. IEEE Transactions on Evolutionary Computation 18(2), 286–300 (2014), http://dx.doi.org/10.1109/TEVC.2013.2260548
8. Chiancone, A., Franzoni, V., Niyogi, R., Milani, A.: Improving link ranking quality by quasi-common neighbourhood. In: Proc. of 15th International Conference on Computational Science and Its Applications, ICCSA 2015. pp. 21–26 (June 2015), doi:10.1109/ICCSA.2015.19
9. Chiancone, A., et al.: A multistrain bacterial diffusion model for link prediction. International Journal of Pattern Recognition and Artificial Intelligence 31(8), 201–233 (2017), doi:10.1142/S0218001417590248
10. Franzoni, V., Mencacci, M., Mengoni, P., Milani, A.: Heuristics for semantic path search in wikipedia. LNCS - Proc. of ICCSA 2014 8584, 327–340 (2014), doi:10.1007/978-3-319-09153-2_25
11. Franzoni, V., Milani, A.: Semantic context extraction from collaborative networks. In: 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 131–136 (May 2015), doi:10.1109/CSCWD.2015.7230946
12. Franzoni, V., Milani, A.: A semantic comparison of clustering algorithms for the evaluation of web-based similarity measures. LNCS - Proc. of ICCSA 2016 9790, 438–452 (2016), doi:10.1007/978-3-319-42092-9_34
13. Gupta, J.N., Jr., E.F.S.: Flowshop scheduling research after five decades. European Journal of Operational Research 169(3), 699 – 711 (2006), http://dx.doi.org/10.1016/j.ejor.2005.02.001
14. Hains, D.R., Whitley, L.D., Howe, A.E.: Revisiting the big valley search space structure in the tsp. Journal of the Operational Research Society 62(2), 305–312 (2011), http://dx.doi.org/10.1057/jors.2010.116
15. Hoos, H.H., Stützle, T.: Stochastic local search: Foundations and applications. Elsevier (2004)
16. Milani, A., Santucci, V.: Community of scientist optimization: An autonomy oriented approach to distributed optimization. AI Communications 25(2), 157–172 (2012), http://dx.doi.org/10.3233/AIC-2012-0526
17. Ochoa, G., Verel, S., Daolio, F., Tomassini, M.: Local Optima Networks: A New Model of Combinatorial Fitness Landscapes, pp. 233–262. Springer Berlin Heidelberg, Berlin, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-41888-4_9
18. Pitzer, E., Affenzeller, M.: A Comprehensive Survey on Fitness Landscape Analysis, pp. 161–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-23229-9_8
19. Santucci, V., Baioletti, M., Milani, A.: Algebraic differential evolution algorithm for the permutation flowshop scheduling problem with total flowtime criterion. IEEE Transactions on Evolutionary Computation 20(5), 682–694 (Oct 2016), http://dx.doi.org/10.1109/TEVC.2015.2507785
20. Santucci, V., Baioletti, M., Milani, A.: A differential evolution algorithm for the permutation flowshop scheduling problem with total flow time criterion. In: Proc.

of Parallel Problem Solving from Nature – PPSN XIII - LNCS. pp. 161–170. Springer International Publishing, Cham (2014), http://dx.doi.org/10.1007/978-3-319-10762-2_16
21. Santucci, V., Baioletti, M., Milani, A.: An algebraic differential evolution for the linear ordering problem. In: Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015, Companion Material Proceedings. pp. 1479–1480 (2015), http://doi.acm.org/10.1145/2739482.2764693
22. Santucci, V., Baioletti, M., Milani, A.: Solving permutation flowshop scheduling problems with a discrete differential evolution algorithm. AI Communications 29(2), 269–286 (2016), http://dx.doi.org/10.3233/AIC-2012-0526
23. Schiavinotto, T., Stützle, T.: A review of metrics on permutations for search landscape analysis. Computers & Operations Research 34(10), 3143 – 3153 (2007), http://dx.doi.org/10.1016/j.cor.2005.11.022
24. Stadler, P.F.: Fitness landscapes, pp. 183–204. Springer Berlin Heidelberg, Berlin, Heidelberg (2002), http://dx.doi.org/10.1007/3-540-45692-9_10
25. Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research 64(2), 278 – 285 (1993), http://dx.doi.org/10.1016/0377-2217(93)90182-M
26. Tomassini, M., Vanneschi, L., Collard, P., Clergue, M.: A study of fitness distance correlation as a difficulty measure in genetic programming. Evolutionary Computation 13(2), 213–239 (2005), http://dx.doi.org/10.1162/1063656054088549
27. Weinberger, E.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cybernetics 63(5), 325–336 (1990), http://dx.doi.org/10.1007/BF00202749
28. Wu, Y., McCall, J.A.W., Corne, D.: Fitness landscape analysis of bayesian network structure learning. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2011, New Orleans, LA, USA, 5-8 June, 2011. pp. 981–988 (2011), http://dx.doi.org/10.1109/CEC.2011.5949724