Solving an Integration Process Planning and Scheduling in a Flexible Job Shop Using a Hybrid Approach

Nassima Keddari[1(✉)] , Nasser Mebarki[2] , Atif Shahzad[3],
and Zaki Sari[1]

[1] Manufacturing Engineering Laboratory of Tlemcen MELT,
University of Tlemcen, Tlemcen, Algeria
keddarinassima@yahoo.fr, z_sari@mail.univ-tlemcen.dz
[2] LUNAM, Université de Nantes, LS2N, Laboratoire des Sciences
du Numérique de Nantes, UMR CNRS 6004, Nantes, France
nasser.mebarki@univ-nantes.fr
[3] Department of Industrial Engineering,
King Abdulaziz University, Jeddah, Saudi Arabia
atifshahzad@gmail.com

Abstract. Traditionally, process planning and scheduling functions are performed sequentially, where scheduling is implemented after process plans has been generated. Recent research works have shown that the integration of these two manufacturing system functions can significantly improve scheduling objectives. In this paper, we present a new hybrid method that integrates the two functions in order to minimize the makespan. This method is made up of a Shifting Bottleneck Heuristic as a starting solution, Tabu Search (TS) and the Kangaroo Algorithm metaheuristics as a global search. The performance of this newly hybrid method has been evaluated and compared with an integrated approach based on a Genetic Algorithm. Thereby, the characteristics and merits of the proposed method are highlighted.

Keywords: Integration · Process planning · Scheduling · Metaheuristics
Shifting bottleneck heuristic · Tabu search · Kangaroo algorithm

1 Introduction

The concept of Industry 4.0 promotes the integration of all aspects of production for greater efficiency. The factory of the future will be a smart manufacturing, holistic and flexible, where Internet of Things, augmented reality, automation and artificial intelligence will enable to adapt quickly the production system to a constantly changing environment. It necessitates a high level of enterprise integration. Yet, in most of manufacturing systems, two distinct functions are still handled independently to manage the production: the process planning function and the scheduling function.

Process planning determines how a product will be manufactured from its initial to a finished product. In other words, which sequence to use and which resource to select [1]. The scheduling is another manufacturing function that finds a mapping between

jobs and resources to achieve some relevant criteria. The output of process planning is an input of scheduling. Therefore, scheduling is based on a fixed process planning. Moreover, process planning doesn't consider the current capacity of the resources while it is a strict constraint for the scheduling function. This sequential organization doesn't permit to take fully advantage of the flexibility provided by modern manufacturing systems. Research on scheduling has focused primarily in the construction of efficient algorithms to solve different types of scheduling problems: flow shop, job shop, open shop, and so on. However, research works show that the integration of the process planning function and the scheduling function permits to gain valuable insights [2–4].

The Flexible Job Shop Scheduling Problem (FJSSP) is an extension of the classical job shop scheduling problem (JSSP), where each operation Oj can be processed by a set of alternative machines, subset of the set of machines M. In that case, pjk denotes the processing time of operation Oj on machine Mk. Machines are always available and while operations are being processed, preemption is not allowed. The aim is to find a schedule for processing these n jobs on the m machines. The quality of a schedule is given by a performance measure objective function f, based on the jobs completion times denoted Ci. One of the most classical objective functions is to minimize the makespan, which measures the total length of the schedule [5]. The makespan, denoted Cmax is computed as Cmax = $\max_{i=1..n}$Ci. Following the Lawler notation [6], this problem is noted J∥Cmax.

The integration of process planning and scheduling in a flexible job shop system (FJSSP-PPF) considers alternative machines for the operations, called operation flexibility (OF), and alternative operations' sequence, called sequencing flexibility (SF).

– In this paper, we address the problem of integrating process planning and scheduling on a flexible job shop. The main contribution of this paper is to propose a very effective hybrid method to evaluate and explore the search space intelligently in a reasonable time frame in order to solve the FJSSP-PPF problem. This method has been evaluated on a manufacturing model from the literature, with objective to find a schedule which minimizes the makespan. This hybrid method, described more precisely in Sect. 3, is based on three stages: Generation of an initial solution using an innovative local search procedure called Shifting Bottleneck Heuristic (SBH).
– The second stage is based on an exploration of the solutions' space using a Tabu Search method. The solution found by the SBH procedure becomes the initial solution of the Tabu Search. An efficient initialization of the solution is an essential aspect of a metaheuristic's performance in terms of solution quality and computing time. As the SBH is one of the most successful heuristics for the J∥Cmax problem, it is of the greatest interest to use it as initial solution for the Tabu Search method.
– Tabu Search gives very effective results to solve the JSSSP, but it is a neighborhood method that may be trapped in local optimum. To avoid this, the solution obtained from the Tabu Search becomes the initial solution of an iterated solution improvement metaheuristic called Kangaroo Algorithm. The Kangaroo method is a combination of local and global search. The algorithm tries to improve the current solution by exploring its neighborhood using an iterative stochastic descent

procedure. When a new improvement is no longer possible a "jump" procedure is performed in order to escape from the attraction of a local minimum [7].

Within this hybrid approach, the SBH to provide an efficient solution, tabu search is used to define an effective neighborhood around the initial solution as a local search method and the kangaroo algorithm is used to perform the global search among neighborhood. The main positive effect of this hybridization is the convergence speed to local optimum and to intensify the local search and the intensification ability of the Kangaroo algorithm (KA) the of global optimum.

The remainder of this paper is organized as follows: the second section proposes a literature review of hybrid methods for flexible scheduling job shop problems, section three describes the case study and the framework of our hybrid approach. In section four experimental results are reported and then discussed in section five. The last section concludes this work and proposes different research perspectives.

2 Literature Review

To solve the FJSSP-PPF problems, which are harder than FJSSP, hybrid methods give promising results. [2] Has developed a linear mixed-integer programming model (LMIPM) for integration problem that relates a Tabu Search (TS) heuristic with branch-and-bound method. The TS algorithm is employed as a fast heuristic to find an initial solution for a branch-and-bound procedure in an LMIPM environment. The feasible initial solution is generated using a specially designed dispatching rule earliest completion time first. [8] Have proposed a symbiotic Genetic Algorithm, which uses an artificial intelligent search technique, to handle, at the same time, the process planning and the scheduling functions. [1] Have analyzed the effect of changing flexible process plan of a part-type in a production order. They use a simulation-based Genetic Algorithm (GA) in order to select the key part-type so that a performance measure can be further improved. This approach generates near-optimal performance for the makespan. [9] Have proposed a new hybrid algorithm to solve the FJSSP-PPF problem with stochastic processing time. This approach combines simulated annealing and Tabu Search heuristic. More recently, [10] have compared the performance of two different heuristics based on genetic algorithms and simulated annealing algorithm, for the FJSSP-PPF problem, in order to minimize the total completion time.

One of the key factors of hybrid methods is the synergetic between the different methods used. Another key factor is the quality of the initial solution. Due to its great efficiency to solve the J‖Cmax problem, the Shifting Bottleneck Heuristic has been largely used in hybrid methods. [11] Have planned a hybrid method which combines a Shifting Bottleneck Heuristic and a Tabu Search algorithm. In their method, the re-optimization step in the Shifting Bottleneck algorithm is replaced by the Tabu search. [12] Have combined a Shifting Bottleneck Heuristic with an iterated local search method for re-optimizing already scheduled machines. Computational experiments show that this hybrid method improve existing results for benchmark instances. Moreover this combination of SBH with an iterated local search method is applicable to large instances of job shop (more than 100 jobs and 20 machines). [13] Have developed a hybrid

method, combining the genetic algorithm with the SBH for the flexible job shop scheduling problem. They generate the initial population randomly in order to maintain the diversity of individuals.

Most of works related to application of Kangaroo Algorithm use it as a level of a hybridization method. [7] Have proposed a hybrid method formed by an Ant Colony System (ACS) and a Kangaroo Algorithm (KA) to solve the single machine scheduling problem. This work is based on the collaborative power of the ACS and the intensification ability of KA.

3   Framework of the Approach

3.1   Case Study

In order to evaluate our method and compare its performances, we have chosen a manufacturing model already used by [14] (see Table 1). [14] Have used a genetic algorithm to solve it and their results are presented in Sect. 4.

Table 1.   Representation of the integration problem

Jobi	Oj	Set of alternatives machines (pj, k)	Process plan number (operation sequence)
J1	1	{M1(6), M2(6)}	1: (1-2-3)
	2	{M2(5), M1(6), M3(6)}	2: (1-3-2)
	3	{M3(4)}	
J2	4	{M1(3), M3(4)}	1: (4-5-6)
	5	{M2(7)}	2: (4-6-5)
	6	{M3(6), M1(5), M2(7)}	3: (6-4-5)
J3	7	{M1(7), M3(8)}	1: (7-8-9)
	8	{M3(5), M1(5), M2(6)}	2: (7-9-8)
	9	{M2(4)}	

This model presents two difficulties. The first one is to choose the order, in which the operations of each job are sequenced (sequencing flexibility), and the second is to assign each operation Oj to a machine Mk selected from the set of alternative machines (operation flexibility). For instance, job J2 has three process plans, each one with a fixed sequence; furthermore, operation O8 may be processed by M3, M1 or M2, with processing time 5, 5 and 6, respectively, but it must be processed after operation O7.

3.2   General Framework

As the FJSSP-PPF problem is a highly combinatorial optimization problem we propose a hybrid procedure where the first stage (the Shifting Bottleneck heuristic) is dedicated to solve a part of the problem as shown in Fig. 1.

In our hybrid procedure, as the SBH doesn't take into account alternative machines and alternative sequences as well, it is used only once, with an instance of a JSSP
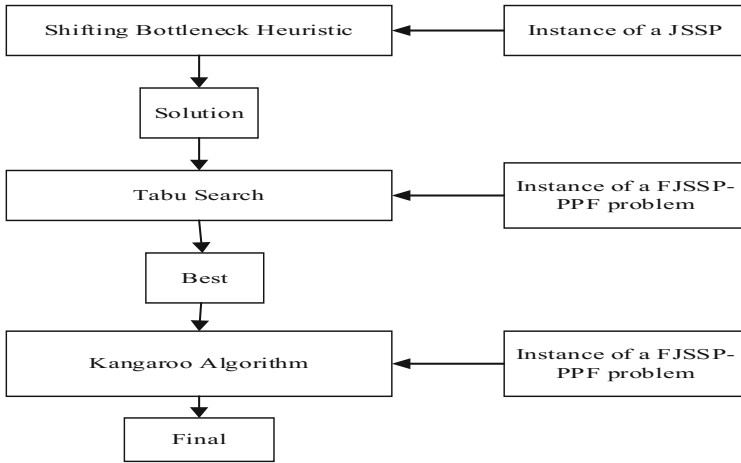
Fig. 1. General framework of our approach

derived from the FSP-PPF problem. Thus, at this stage, for each job a process plan is chosen arbitrarily and for each operation a machine is assigned arbitrarily. The Tabu Search (TS) and the Kangaroo Algorithm take into account both alternative sequences and alternative machines. The idea is that, at each stage, very good quality solutions are obtained permitting to improve this solution at the next stage.

3.3   Shifting Bottleneck Heuristic (SBH)

The Shifting Bottleneck Heuristic (SBH) was proposed by [15] to solve the minimum makespan problem for the job shop problem. The principle of this heuristic is to iteratively determine a machine considered as bottleneck and to optimally schedule this machine only. For each machine, a single machine scheduling problem 1 |rj| Lmax is solved using the branch and bound technique proposed by [16]. The bottleneck machine is the one with the highest Lmax. Then, using a disjunctive graph, the machines already scheduled are re-sequenced to include the optimal sequence of the current bottleneck machine. For each operation Oj, a release date rj and a due date dj are computed iteratively: rj is the earliest beginning date of operation Oj, computed from its predecessors already scheduled; dj is the latest completion time of operation Oj. For a more complete presentation of the SBH, the reader may refer to [17].

3.4   Tabu Search

Tabu Search (TS) algorithms are among the most effective approaches for solving JSSP [18]. They use a memory function to avoid being trapped in a local optimum [19]. Neighborhood structures and move evaluation strategies play the central role in the effectiveness and efficiency of the Tabu Search for the JSSP [20]. Most of the TS algorithms related to job shop scheduling are based on the pair-exchange method for generating neighborhood solutions. It depends on a permutation of operations placed

on position i and $i + 1$ on a given machine [2]. During the Tabu Search exploratory, two types of neighborhood structures are added to take into account the two distinct decisions of the FJSSP-PPF problem: process plans' selection and machine's affectation. Finally, it gives the following movement operators:

- For a machine, permute the order of two sequential operations (operation exchange) when these two operations are on the critical path of the solution.
- For a critical operation, assign an alternative machine (machine exchange).
- For a job with at least a critical operation, change the process plan (sequence exchange).

These three movement operators will also be used during the stochastic descent of the Kangaroo Algorithm, described in the next section.

3.5   Kangaroo Algorithm

As most heuristics, there is no theoretical results ensuring the convergence of a Tabu Search procedure to a global optimum. Further progress is achieved by using a jump procedure to potentially move away from the previous local optimum. A very promising heuristic that includes a jump mechanism is the Kangaroo Algorithm proposed by [21]. It is described in Fig. 4. The principle of the Kangaroo method is similar to the simulated annealing algorithm but with a different research strategy. It generates a solution by using an iterative procedure that contains two parts: the stochastic descent procedure and the jump procedure [7].

During the stochastic descent, Kangaroo Algorithm seeks a solution that minimizes a function f(S) in a neighborhood N(S) of the current solution S using a local uniform mutation η1. If the new solution S' is better than the previous solution, it is stored and a new solution is explored in the same neighborhood N(S'). The algorithm tries to improve the current solution A times, A being the maximum number of iterations in improving the current solution before a jump. If it is not possible to obtain a further improvement, the algorithm moves to another neighborhood with a jump procedure using a global uniform mutation η2.

3.6   Solution Representation

To solve FJSSP-PPF using TS and KA, we first need to represent the solution of our problem as a chromosome. Each individual in FJSSP-PPF consists of three vectors with the same length, as shown in Fig. 2.

7	4	6	1	9	5	2	8	3	→	Operation number vector
2	3	3	1	2	3	1	2	1	→	Process plan number vector
3	3	1	2	2	2	1	1	3	→	Machine assignment vector

Fig. 2. Solution structure

With this representation, by selecting a chromosome, the FJSSP-PPF problem is transformed into a JSSP problem formulation. Depending on the selection of each job's sequence and each operation's machine it produces different instances of a JSSP. For example, the chromosome presented in Fig. 2 gives the following schedule (Fig. 3):
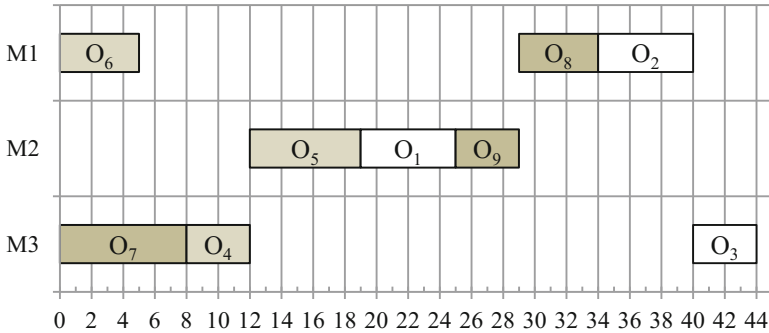

Fig. 3. Gantt chart of the solution derived from the chromosome selected in Fig. 2.

It can be noticed that the chromosome representation gives a semi-active schedule and not necessarily an active schedule. Active schedules are dominant for the makespan. For example, the example in Fig. 3 gives a semi-active schedule but not an active schedule.

4   Results

Our hybrid approach was coded in JAVA software Intel(R), Core (TM) i7 with a 2.2 GHz CPU. First, we present the solution obtained by the genetic algorithm developed by [14] in Fig. 4.
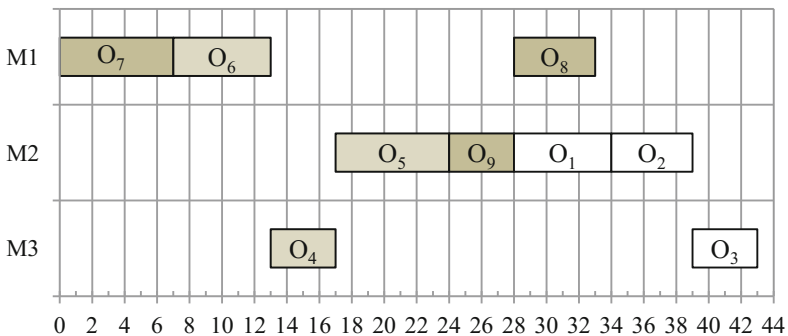

Fig. 4. Gantt chart of the solution proposed by [14]

The solution proposed by [14] is not an active schedule. For example, operations O1, O2, O3, O9 and O8 could begin earlier, on the same machines, without delaying any other operation, giving an active schedule. However, their solution gives a makespan of 43. In the following, we present the different solutions, on the same case study, obtained at each stage of our method. The SBH runs with the selections shown in Table 2: for each job the first process plan is chosen and for each operation the first machine is assigned.

Table 2. Solution for the SBH (critical operations are shaded)

4	5	7	8	6	1	2	3	9	Operation number vector	S(SBH)
1	1	1	1	1	1	1	1	1	Assigned process plan vector	
1	2	1	3	3	1	2	3	2	Assigned machines vector	

The outcome of the SBH gives the following critical operations: O4, O7, O1, O2 and O9. Gantt chart of the SBH solution is shown in Fig. 5. This solution is an active schedule.
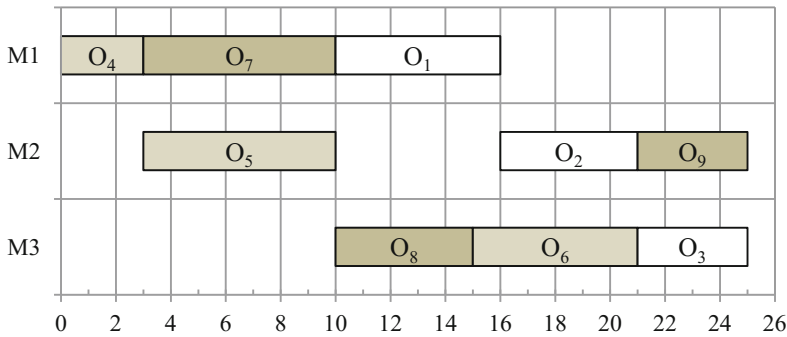

Fig. 5. Gantt chart of the SBH solution

Table 3 shows the results obtained after the Tabu Search procedure. Only job J1 has been modified: its process plan number is now the second (sequence exchange) and there is a machine exchange on operations O1 and O2. The critical operations become: O4, O7, O8 and O6. Figure 6 gives the Gantt chart of the TS solution.

Table 3. Presentation of the obtained SBH+TS solution

4	1	5	7	2	3	8	9	6	Operation number vector	S(SBH+TS)
1	2	1	1	2	2	1	1	1	Assigned process plan vector	
1	2	2	1	1	3	3	2	3	Assigned machines vector	

Finally, Table 4 shows the outcome after the Kangaroo Algorithm. This time, the modifications are on job J2 (sequence exchange and machine exchange for operations O4 and O6). The critical operations become O7, O6, O5. Figure 7 gives the Gantt chart obtained after the KA procedure.
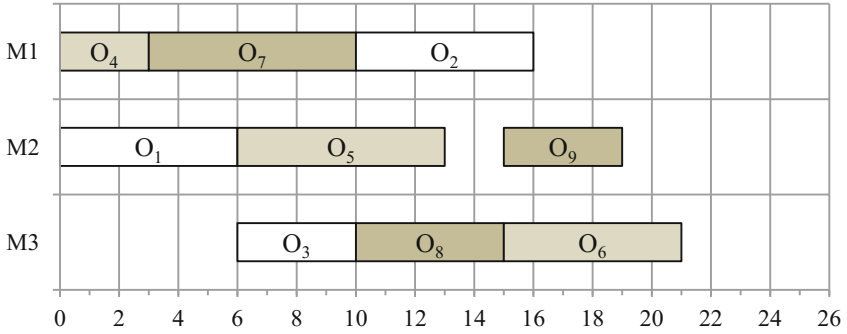
Fig. 6. Gantt chart of the SBH+TS solution

Table 4. Presentation of the obtained SBH+TS+KA solution

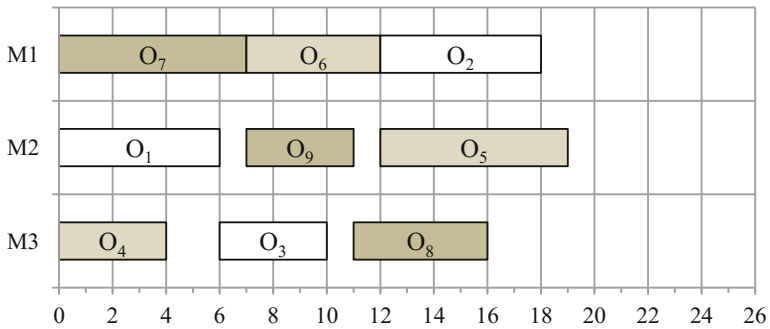| 4	1	7	6	3	2	9	8	5	Operation number vector	S(SBH+TS+KA)
2	2	2	2	2	2	2	2	2	Assigned process plan vector	
3	2	1	1	3	1	2	3	2	Assigned machines vector	

Fig. 7. Gantt chart of the SBH+TS+KA solution

Table 5 gives the number of generations, computational time and Cmax value obtained at each stage of our hybrid approach. The lowest Cmax value is found in 20 s by the hybrid method (SBH+TS+KA).

Table 5. Computational results

Method	Generations number	Comp. Time (seconds)	C_{max}	Improvement of the solution
GA [14]	Not provided	Not provided	44	
SBH	1	<1	25	43% (w.r.t. to GA)
SBH+TS	100	11	21	21% (w.r.t. to SBH)
SBH+TS +KA	100	20	19	5% (w.r.t. to SBH+TS)

5   Discussion

The solution provided by the GA is not well balanced (see Fig. 4). Among nine operations, four of them are assigned to machine M2, which gives a workload of 22 for this machine, while machine M3, with only two operations, has a workload of 8. Two successive operations of job J1:O1 and O2 are performed on the same machine M2. There is another cycle, with job J3 on machine M1. These cycles are one of the causes of the poor performance of the GA procedure proposed by [14].

The solution provided by the SBH is the most balanced solution of all obtained solutions (a workload of 16, 16 and 15 for machines M1, M2 and M3 respectively). As we can see in Table 5, the makespan is significantly improved by SBH w.r.t. to the solution obtained by [14]. This shows the importance of the initial solution when using metaheuristic. Moreover, as the SBH deals only with one JSSP instance, the solution is obtained immediately. Thanks to the high quality of the initial solution of the TS, the TS exploratory permits to improve again, and largely, the makespan (21%). With regards to the SBH solution, the new solution has permuted the machines of operations O1 and O2 (M1 and M2) and has changed the process plan of J1. Thus, during the TS search, only the operation and sequence flexibility of J1 has been updated.

Inside the hybrid system, SBH and TS methods are used to achieve local exploitation around, while the KA is used to achieve global exploration. Although the TS has already well explored the search space and has given a high quality solution, the KA achieves to improve this solution. The KA's final solution has reassigned operations O4 and O6 to alternative machines M3 and M1 respectively. Thanks to its jump mechanism, the KA can explore globally the search space and finally the process plan flexibility is used for all jobs and permits to improve the solution. It can be noticed that at the end of the hybrid procedure (SBH+TS+KA), the final solution is improved by 24% w.r.t. the solution obtained with only the SBH.

6   Conclusion and Future Work

In this paper, we developed a new approach that simultaneously integrates process planning and scheduling functions in a flexible job shop in order to minimize the makespan. This approach consists of hybridizing three heuristics: Shifting Bottleneck Heuristic (SBH), Tabu Search (TS) and Kangaroo Algorithm (KA). This approach has been compared with an integrated approach based on a Genetic Algorithm.

Our strategy is to start from a feasible initial solution created by the SBH. This heuristic gives a very good performance for the makespan in less than one second. This solution becomes a starting point for the Tabu Search. TS improves the solution obtained from the SBH by integrating operation flexibility and sequence flexibility. Again, the obtained solution becomes the initial solution for the next phase. In this phase, the search area is expanded by a global search algorithm called Kangaroo Algorithm, which improves the current solution by performing a jump procedure. The best makespan is produced by the hybridation SBH+TS+KA and outperforms the Genetic Algorithm by a factor of 2.3.

Although the scheduling problem studied is in this paper has only three machines and three jobs, due to its high flexibility, it is hard to get an optimal solution. However, it will be of the greatest interest to evaluate the efficiency (in terms of quality and speed) of our approach with a more complex problem. Comparisons of this approach with various optimization algorithms, such as exact methods or other metaheuristics, should also be very interesting. Another future work is to propose a multi-objective framework for solving the FSSP-PPF problem in order to minimize simultaneously the makespan and the maximum lateness.

References

1. Kumar, M., Rajotia, S.: Integration of process planning and scheduling in a job shop environment. Int. J. Adv. Manuf. Technol. **28**(1–2), 109–116 (2006)
2. Tan, W.: Integration of process planning and scheduling. A mathematical programming approach. Ph.D. Dissertation, University of Southern California (1998)
3. Tan, W., Khoshnevis, B.: Integration of process planning and scheduling – a review. J. Intell. Manuf. **11**(1), 51–63 (2000)
4. Li, W.D., McMahon, C.A.: A simulated annealing-based optimization approach for integrated process planning and scheduling. Int. J. Comput. Integr. Manuf. **20**(1), 80–95 (2007)
5. Samur, S., Bulkan, S.: An evolutionary solution to a multi-objective scheduling problem. In: Proceedings of the World Congress on Engineering 2010, WCE, vol. III (2010)
6. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.: Sequencing and scheduling: algorithms and complexity. Handb. Oper. Res. Manag. Sci. **4**, 445–522 (1993)
7. Serbencu, A., Minzu, V., Serbencu, A.: An ant colony system based metaheuristic for solving single machine scheduling problem. Ann. Dunarea De Jos Univ. Galati Fascicle III **3**(1), 19–24 (2007). ISSN 1221-454X
8. Kim, Y.K., Park, K., Ko, J.: A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. Comput. Oper. Res. **30**, 1151–1171 (2003)
9. Haddadzade, M., Razfar, M.R., Zarandi, M.H.F.: Integration of process planning and job shop scheduling with stochastic processing time. Int. J. Adv. Manuf. Technol. **71**, 241–252 (2014)
10. Botsali, A.R.: Comparison of simulated annealing and genetic algorithm approaches on integrated process routing and scheduling problem. Int. J. Intell. Syst. Appl. Eng. Adv. Technol. Sci. **4**, 101–104 (2016). ISSN 2147-67992
11. Bulbul, K.: A hybrid shifting bottleneck-tabu search heuristic for the job shop total weighted tardiness problem. Comput. Oper. Res. **38**(6), 967–983 (2011)
12. Braune, R., Zapfe, G.: Shifting Bottleneck scheduling for total weighted tardiness minimization: a computational evaluation of subproblem and re-optimization heuristics. Comput. Oper. Res. **66**, 130–140 (2016)
13. Gen, M., Gao, J., Lin, L.: Multistage-based genetic algorithm for flexible job-shop scheduling problem. In: Gen, M., et al. (eds.) Intelligent and Evolutionary Systems. Studies in Computational Intelligence, vol. 187, pp. 183–196. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-95978-6_13
14. Park, B.J., Choi, H.R.: A genetic algorithm for integration of process planning and scheduling in a job shop. In: Sattar, A., Kang, B.-H. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 647–657. Springer, Heidelberg (2006). https://doi.org/10.1007/11941439_69

15. Adams, J., Balas, E., Zawac, D.: The shifting bottleneck procedure for job shop scheduling. Manag. Sci. **34**(3), 391–401 (1988)
16. Carlier, J.: The one-machine sequencing problem. Eur. J. Oper. Res. **11**(1), 42–47 (1982)
17. Pinedo, M.: Scheduling Theory, Algorithms, and Systems. Springer, New York (2012). https://doi.org/10.1007/978-1-4614-2361-4
18. Jain, A.S., Meeran, S.: Deterministic job-shop scheduling: Past, present and future. Eur. J. Oper. Res. **113**, 390–434 (1998)
19. Zhang, C.Y., Li, P., Rao, Y., Guan, Z.: A very fast TS/SA algorithm for the job shop scheduling problem. Comput. Oper. Res. **35**(1), 282–294 (2008)
20. Jain, A.S., Rangaswamy, B., Meeran, S.: New and stronger job-shop neighborhoods: a focus on the method of Nowicki and Smutnicki. J. Heuristics **6**(4), 457–480 (2000)
21. Fleury, G.: Application des méthodes stochastiques inspirées du recuit simulé à des problèmes d'ordonnancement. Automatique Productique Informatique Industrielle **29**(4–5), 445–470 (1995)