engineering proceedings

Proceeding Paper

Optimization of the Flow-Shop Scheduling Problem under Time Constraints with PSO Algorithm [†]

Milad Mansouri [*], Younes Bahmani and Hacene Smadi

Laboratory of Automation and Manufacturing Engineering, University of Batna 2, Batna 05000, Algeria; y.bahmani@univ-batna2.dz (Y.B.); h.smadi@univ-batna2.dz (H.S.)

[*] Correspondence: milad.mansouri@univ-batna2.dz

[†] Presented at the 4th International Electronic Conference on Applied Sciences, 27 October–10 November 2023; Available online: https://asec2023.sciforum.net/.

Abstract: This study aims to minimize the makespan objective function in a Flow-Shop environment, considering two crucial temporal constraints: "Waiting time" and "Release date". Given the NP-hardness of this scheduling problem, we employed an enhanced metaheuristic called particle swarm optimization (PSO) to find the optimal solution. Through a series of experiments conducted on a specific set of benchmark instances, we evaluated the performance of our approach by comparing the obtained results against the lower bound (LB). This comparison showcased the effectiveness of our proposed method in addressing this complex scheduling problem.

Keywords: flow-shop scheduling problem; release date; waiting time; makespan; particle swarm optimization

Citation: Mansouri, M.; Bahmani, Y.; Smadi, H. Optimization of the Flow-Shop Scheduling Problem under Time Constraints with PSO Algorithm. *Eng. Proc.* **2023**, *56*, 220. https://doi.org/10.3390/ASEC2023-15410

Academic Editor: Alessandro Bruno

Published: 27 October 2023

1. Introduction

In production management, the scheduling optimization process aims to determine a sequence of tasks, taking into consideration various constraints, to enhance productivity and establish a more efficient production system.

Our research stands out due to the integration of two time constraints which are the waiting time and release date, attempting to minimize the makespan criterion. Adding these constraints makes the problem more complex and closer to reality; however, it is well-known as NP-hard according to the literature, making it computationally challenging. In this case, we turn to advanced optimization techniques to find optimal or nearly optimal solutions. One such technique is particle swarm optimization (PSO), a metaheuristic known for its ability to handle complex optimization problems by using the collective intelligence of particles navigating the search space.

In evaluating our research's effectiveness, we employed our methodology on a series of benchmark tests and used the deviation index (DIV) for a more in-depth analysis of the results. The findings appear to be extremely encouraging.

2. Flow-Shop Scheduling under Waiting Time and Release Date Constraints

In accordance with Graham's proposed classification, the problem under consideration can be represented as $F \mid r, Wait - Time \mid C_{max}$.

2.1. Description of the Problem

We have a set of N jobs that need to be assigned and processed on a set of M machines. The objective is to find the optimal sequence of jobs on machines that optimizes the makespan within wait time and release date constraints to improve process efficiency.

Key attributes of a flow shop are:

- Each job follows a production line flow and remains confined to it, traversing all M machines sequentially (ranging from $M1$ to Mm).

- The production line can manufacture various products.
- Notably, each machine can handle only one task at a given moment, while each job can be processed by only one machine at a time.
- Importantly, task processing occurs continuously, without any interruptions
- The predetermined processing times $p_{\mathrm{i,j}}$ for each task, remain constant throughout.
- Equally, the waiting time and release date for tasks are predetermined and remain consistent.

2.1.1. Makespan Objective Function

In a scheduling problem, the makespan function measures the total duration required to conclude all tasks. Attaining an optimal makespan in the flow shop leads to improved productivity, optimized resource allocation, meeting deadlines, and reduced costs. Several approaches have been proposed to minimize makespan in flow-shop, such as; three metaheuristics (AIS, IG, AIS–IG), that offer improved solutions and scalability [1]. CDS heuristic compared with PSO metaheuristic and a proposed method that combines SPT and permutation rules [2]. A hybrid algorithm combining tabu search with genetic algorithm improves scheduling solutions [3], A novel tie-breaking rule enhances the NEH heuristic, improving scheduling in flow shops efficiently [4].

2.1.2. Wait Time and Release Date Constraints

- The wait time constraint in scheduling determines the maximum allowed time a task can wait after its release before starting the process. Mainly, it resolves conflicts by prioritizing tasks based on urgency or with shorter waiting times. It is used, for example, to wait until the temperature of the part or equipment decreases before further processing.
- The release date specifies the earliest time the task can start and thereby affects task star times. In addition, it contributes to coordinating tasks and ensures that tasks start on time to meet specific scheduling requirements. This constraint is usually used in the assembly process for task coordination.

To our knowledge, no study has yet combined these two constraints, although some research has explored them individually [5,6]. Adding these two constraints increases the problem's complexity, requiring the use of advanced algorithms, like particle swarm optimization (PSO).

2.2. Mathematical Problem Formulation

To establish the mathematical model for this scheduling problem, we introduce additional notation presented in Table 1.

$$C_{max} = min(max(c_{\pi_i,m})) \qquad \forall\, i \in n \tag{1}$$

$$c_{\pi_i,k+1} \geq c_{\pi_i,k} + w_{\pi_i,k+1} + p_{\pi_i,k+1} \qquad \forall\, i \in n\, ;\, k \in m-1 \tag{2}$$

$$c_{\pi_j,k} - c_{\pi_i,k} + G\Big(3 - \delta_{\pi_i,k} - \delta_{j,k} - \mu_{\pi_i,\pi_j,k}\Big) \geq p_{\pi_i,k} + w_{\pi_j,k} \qquad \forall\, i,j \in n\, , i \neq j\, ,\, k \in m \tag{3}$$

$$c_{\pi_i,k} - c_{\pi_j,k} + G\Big(2 - \delta_{\pi_i,k} - \delta_{\pi_j,k} - \mu_{\pi_j,\pi_i,k}\Big) \geq p_{\pi_i,k} + w_{\pi_i,k} \qquad \forall\, i,j \in n\, , i \neq j\, ,\, k \in m \tag{4}$$

$$c_{1,1} = r_1 + p_{1,1} \tag{5}$$

$$c_{\pi_i,1} = max\big(r_{\pi_i}, c_{\pi_{i-1},1}\big) + p_{\pi_i,1} \qquad \forall\, i \in n+1 \tag{6}$$

$$\sum_{1\leq k\leq m} \delta_{\pi_i,k} = m \qquad \forall\, i \in n \tag{7}$$

$$\sum_{1\leq i\leq n} \delta_{\pi_i,k} = n \qquad \forall\, k \in\ m \tag{8}$$

Table 1. Parameters and binary variables used in the mathematical model.

Parameter	Signification	Parameter	Signification
i, j	Index of jobs	c	Completion time
k	Index of machines	s	Starting time
π_i	ith job in the sequence π	w	waiting time
n	Number of jobs to be processed	r	Release date
m	Number of processing machines	G	Large positive number
p	Processing time	$\delta_{\pi_i,k}$	Take 1; if i is processing on machine k, and 0 otherwise
		μ_{ijk}	Take 1; if i is processed before j, $(i \prec j)$ on k and 0 otherwise

(1) The makespan objective function. (2) The succession of the same job on the machines. within a given sequence. (3) And (4) signifying one machine can treat one task at the same time in the same job's sequence, considering the wait time constraint. (5) Completion time of the first job in the first machine including the release date. (6) Completion time of all jobs in the first machine. (7) A job visits all machines once. (8) Each machine processes only one job at a time.

2.3. Complexity of the Problem

The complexity of the scheduling problem is widely acknowledged in research as being NP-hard beyond five machines m ≥ 5 [7–10], indicating their challenging nature to resolve. Approaches based on heuristic or metaheuristic methods are necessary to tackle this problem. Although optimality is not guaranteed with these methods, they do provide effective approximations

3. Particle Swarm Optimization (PSO)

The particle swarm optimization (PSO) algorithm, derived from the social dynamics observed in avian and aquatic species, employs particles that represent solutions to explore the multidimensional solution space to attain optimal or near-optimal results. PSO's efficacy lies in its capacity to strike a balance between exploration and exploitation of the search space, making it adept at addressing complex challenges, inclusive of NP-hard issues. This algorithm has proven its effectiveness in flow-shop scheduling, as showcased in several research studies [11–14].

4. Experimental Evaluation and Results

To conduct our experiment, we implemented a particle swarm optimization (PSO) algorithm using (MATLAB R2018b) in a computer: Intel(R) Core (TM) i5-7200U CPU, 2.50 GHz, 2.71 GHz,4.00 Go memory, and a database sourced from the following site: https://github.com/chneau/go-taillard/tree/master/instances (accessed on 16 April 2023). Table 2 represents a summary of the found results for twelve instances. The steps of the experiment are detailed as follows:

a. The experiment encompassed 12 benchmarks, each one comprising 10 distinct instances. We conducted 10 tests for each of these 12 benchmarks.
b. Each benchmark is characterized by two elements: the number of jobs requiring processing and the number of processing machines.
c. We focused on the following dimensions for small-sized problems: (20 $\times$ 05; 20 $\times$ 10; 20 $\times$ 20; 50 $\times$ 05; 50 $\times$ 10; 50 $\times$ 20).
d. For larger-sized problems, we utilized the following dimensions: (100 $\times$ 05, 100 $\times$ 10, 100 $\times$ 20, 200 $\times$ 10, 200 $\times$ 20, and 500 $\times$ 20).
e. Each trial yielded the following results: CMax, deviation (Div), Lower Bound, and the CPU time.
f. To enhance the result analysis, we selected the best 10 makespan values.
g. For a better interpretation of the results, we employed the relative "index of deviation".

$$\text{Div} = \frac{\text{Global fitness} - \text{Lower Bound}}{\text{Lower Bound}} \times 100 \tag{9}$$

Table 2. Summary table of the found results for all instances.

	Instance	CPU	DIV	Global Fitness	LB
01	20_05	0.9989923	4.491525424	1233	1180
02	20_10	1.198532533	3.875476493	1635	1574
03	20_20	1.469578899	28.16153029	2412	1882
04	50_05	2.479333432	5.095541401	2831	2983
05	50_10	2.824219993	10.24390244	3390	3075
06	50_20	3.489259796	18.55670103	4370	3686
07	100_05	4.850529547	−3.572082799	5264	5459
08	100_10	5.528670233	3.163265306	6066	5880
09	100_20	6.834969121	10.83619019	7436	6709
10	200_10	10.9573531	1.223295807	11,419	11,281
11	200_20	13.69698202	8.222240623	13,070	12,077
12	500_20	34.14775248	1.413757345	29,339	28,930

The values of this index range between 0 and 100, with lower values indicating better results.

Note: We made modifications to the lower bound values extracted from the database since we introduced new constraints, whereas the database values were originally unconstrained.

The following Figure 1 illustrates the values from Table 2. In Figure 1a, we observe a linear increase in the values of makespan and lower bound as the complexity of the problem grows. However, once we reach the (100 × 20) instance, we notice exponential growth in both. The two curves are almost identical showing that the values found by our approaches are very close to the optimal. Figure 1b, demonstrates that CPU time values increase proportionally with problem complexity.

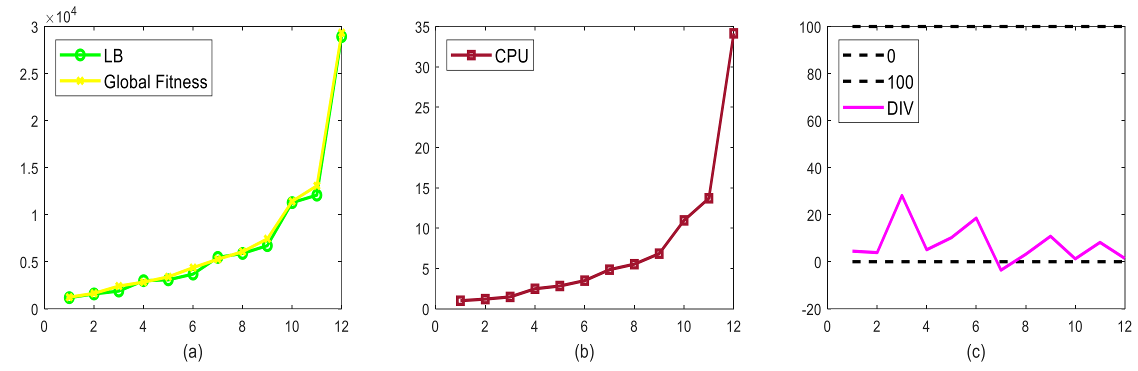

Figure 1. Results from applying PSO to a constrained flow-shop scheduling problem are presented as follows: (**a**) a comparison between the global fitness and lower bound (LB); (**b**) CPU time progression relative to problem complexity; (**c**) an analysis of the index of deviation (DIV).

To identify the factors influencing the complexity of the problem, we compared the results of one instance to those of a larger instance. Our observations indicate that as the number of jobs increases, the complexity of the problem increases significantly. However, when the number of machines increases, the complexity also rises, but to a lesser extent.

In summary, our study reveals that the increase in the number of jobs directly contributes to an increase in the complexity of the problem and discusses the impact of integrating waiting time and releasing date constraints on the scheduling decisions.

For better clarity, Figure 1c represents the deviation index (DIV), used to quantify the gap between the lower bound values and the makespan values obtained through our approach. This indicator ranges from 0 to 100, and the values obtained in our study are closer to 0 than 100. Furthermore, the negative value in line 7 of Table 2 indicates that the found solution is better than the optimal one provided by LB, thereby demonstrating the effectiveness of our method.

Figure 2 is an example illustrating a Gantt chart of a sequence of 5 jobs, each represented by different colors, showcasing the impact of the waiting time constraint. For instance, the first job, represented by the brown color, completed its processing on the first machine at time 4. However, it had to wait until time 7 before starting its processing on machine 2, despite the availability of machine 2.

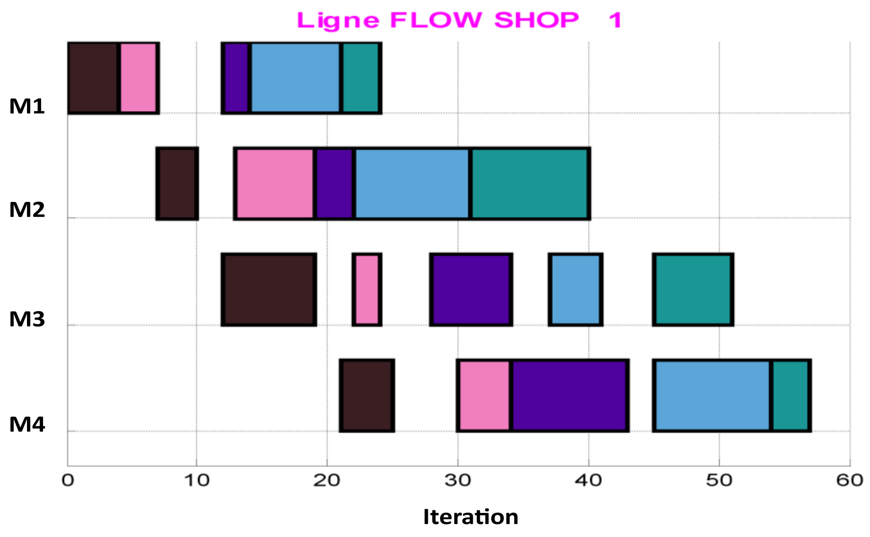

Figure 2. Gantt Chart representation of a 5-Job Sequence.

5. Conclusions

In conclusion, in this study we used the PSO algorithm to generate a sequence of jobs while minimizing the makespan function under the constraints of waiting time and release date in a flow-shop production workshop. These time constraints made the problem more complex. To our knowledge, there has been no prior work discussing these constraints simultaneously in a flow shop type workshop. We tested our approach on a set of benchmarks using the deviation index (DIV) to evaluate its effectiveness. Our approach produced values close to 0, thus indicating its performance. Our study also showed that the number of jobs has a great influence on the complexity of the system.

Author Contributions: Software development, M.M.; methodology, M.M.; validation, M.M. and Y.B.; results analysis, Y.B.; Investigation, M.M.; preparation and writing, M.M.; supervision, Y.B.; verification and correction, Y.B.; management and coordination, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Abdollahpour, S.; Rezaeian, J. Minimizing makespan for flow shop scheduling problem with intermediate buffers by using hybrid approach of artificial immune system. *Appl. Soft Comput.* **2015**, *28*, 44–56. [CrossRef]
2. Irman, A.; Febianti, E.; Khasanah, U. Minimizing makespan on flow shop scheduling using Campbel Dudek and Smith, particle swarm optimization, and proposed heuristic algorithm. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *673*, 012099. [CrossRef]

3. Umam, M.S.; Mustafid, M.; Suryono, S. A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 7459–7467. [CrossRef]
4. Sharma, M.; Sharma, M.; Sharma, S. An improved neh heuristic to minimize makespan for flow shop scheduling problems. *Decis. Sci. Lett.* **2021**, *10*, 311–322. [CrossRef]
5. Han, J.H.; Lee, J.Y. Scheduling for a flow shop with waiting time constraints and missing operations in semiconductor manufacturing. *Eng. Optim.* **2023**, *55*, 1742–1759. [CrossRef]
6. Grabowski, J.; Skubalska, E.; Smutnicki, C. On Flow Shop Scheduling with Release and Due Dates to Minimize Maximum Lateness. *J. Oper. Res. Soc.* **1983**, *34*, 615. [CrossRef]
7. Brucker, P.; Knust, S.; Wang, G. Complexity results for flow-shop problems with a single server. *Eur. J. Oper. Res.* **2005**, *165*, 398–407. [CrossRef]
8. Garey, M.R.; Johnson, D.S.; Sethi, R. Complexity of Flowshop and Jobshop Scheduling. *Math. Oper. Res.* **1976**, *1*, 117–129. [CrossRef]
9. Martinez, S.; Dauzère-Pérès, S.; Guéret, C.; Mati, Y.; Sauer, N. Complexity of flowshop scheduling problems with a new blocking constraint. *Eur. J. Oper. Res.* **2006**, *169*, 855–864. [CrossRef]
10. Soukhal, A.; Oulamara, A.; Martineau, P. Complexity of flow shop scheduling problems with transportation constraints. *Eur. J. Oper. Res.* **2005**, *161*, 32–41. [CrossRef]
11. Mansouri, M.; Bahmani, Y.; Smadi, H. A Hybrid Method for the Parallel-Flow Shop-Scheduling Problem. *Comput. Sci. Math. Forum* **2023**, *7*, 14413. [CrossRef]
12. Bank, M.; Ghomi, S.M.T.F.; Jolai, F.; Behnamian, J. Application of particle swarm optimization and simulated annealing algorithms in flow shop scheduling problem under linear deterioration. *Adv. Eng. Softw.* **2012**, *47*, 1–6. [CrossRef]
13. Mansouri, M.; Bahmani, Y.; Smadi, H. *Monocriterion Optimization in Parallel Flow Shop*; EasyChair: Manchester, UK, 2022.
14. Liao, C.J.; Tseng, C.-T.; Luarn, P. A discrete version of particle swarm optimization for flowshop scheduling problems. *Comput. Oper. Res.* **2007**, *34*, 3099–3111. [CrossRef]