Vol. 13, No. 1, 2026, Pages 86-96

JIEMS
Journal of Industrial Engineering and Management Studies

journal homepage: www.jiems.icms.ac.ir

A flow shop scheduling model considering no-wait constraints between sequential tasks

Fatemeh Zahmatkesh[1], Maryam Daneshmand-Mehr[1]*, Milad Abolghasemian[2]

[1] Department of Industrial Engineering, La. C., Islamic Azad University, Lahijan, Iran.

[2] Department of Industrial Engineering, Ayandegan University, Tonekabon, Iran.

Received: Oct 2025-24 / Revised: Feb 2026-19 / Accepted: Jun 2026-15

Abstract

Flow shop scheduling has traditionally focused on minimizing completion time; however, growing energy costs and sustainability concerns necessitate the simultaneous consideration of energy consumption as an optimization objective. In addition, in many real-world manufacturing systems such as chemical, food, and continuous production industries jobs must be processed without waiting between successive operations, a constraint that significantly increases problem complexity. Despite its practical relevance, the combined consideration of no wait flow shop scheduling and energy-aware optimization has received limited attention in the literature. This study develops a bi-objective no wait flow shop scheduling model that simultaneously minimizes makespan and total energy consumption, accounting for both active and idle energy usage of machines. A mathematical formulation is proposed, and due to the NP-hard nature of the problem, the NSGA II metaheuristic algorithm is employed to obtain high-quality Pareto-optimal solutions. Model validity is verified using small-scale instances solved optimally, followed by large-scale computational experiments. Furthermore, a comprehensive parametric sensitivity analysis is conducted to examine the influence of key factors including processing time, number of tasks, and machine energy consumption in active and idle states on both objectives. The results reveal that idle energy consumption has a disproportionately high impact on total energy usage and completion time, underscoring its critical role in sustainable scheduling decisions. The findings provide both theoretical insights and practical guidelines for designing energy-efficient schedules in no wait production environments.

Keywords: Flow shop scheduling, Mathematical modeling, Energy consumption, NSGA II

Paper Type: Original Research

1. Introduction

The flow shop scheduling problem is one of the most important combinatorial optimization problems, in which a set of jobs must be processed on a set of machines in the same order. The problem can be formulated as permutation or non-permutation, depending on whether the job sequence is identical across all machines (Khatami et al., 2020). The primary objective in classical flow shop scheduling is the minimization of makespan. Flow shop scheduling is related to scheduling jobs so that various machines can do various jobs and this can be considered as one of the methodologies for manufacturing planning (Han et al, 2023). The basic structure of this problem is permutation. So that each job is assigned to a machine and other machines cannot do that job and it would be continued until all jobs are assigned to all machines. The procedure should be so that the makespan is minimized. There are several assumptions that can be considered in this problem. For example, machine elitism, sequence-dependent setup time, and many other assumptions can influence the problem and can make it more complex (Want et al, 2023). In addition to the classic flow shop scheduling problem, there are several researches in which there a various conditions of processing. One of these assumptions is no wait flow shop problem in which all tasks of a job should be done with no delay. It means when the processing tasks begin there will be no wait time between tasks processing (Khatami et al, 2023). No wait scheduling is considered a special mode of continuous operation scheduling in which the operation of tasks should be done with an exact time delay between them. Therefore, no waiting time is allowed between consecutive operations of a job (Perez Gonzalez et al 2024). Among researchers in the field of flow shop, scheduling is focused on minimization of completion time, and this goal is considered a classic one in this field but the fact is that the problem can be converted to a multi-objective problem. One of the main objectives can be the energy consumption of machines so that task assignments to machines to be done while minimizing task completion time, and the energy consumption is decreased too. This subject is important since the sequential operation for an operation without wait can lead to energy consumption and as a result, it can reduce the optimization of energy. The current research problem seeks to provide a model that can lead to a reduction of completion

*Corresponding Author: mdaneshmand@iau.ac.ir

time and energy consumption. On the other hand, the model is designed based on the assumption that a set of operations of a job should be done without stop that can make the problem more complex. In this research, there are 2 objectives one tries to minimize completion time and the second one aims to minimize energy consumption. In fact, the model aims to minimize both energy and completion time. This will be done based on the fact that the procedure of doing tasks is so that they should be done without waiting. And no step is permissible in this model. So it is expected that the assignment of tasks to be so that it can accomplish the cited objectives. The nature of tasks is so that there should be no wait in doing work therefore based on this mode both makespan and energy should be optimized. By assigning tasks to machines based on machine characteristics optimization can be achieved. For solving the model NSGA-II algorithm is used. The algorithm is selected in this research because its efficiency is proved in similar research, especially flow shop scheduling therefore it is indicated as an important algorithm that can be helpful in this field. Finally, the researcher seeks to respond to the question of how the flow shop scheduling model works by considering no wait between sequential operations of each job. The paper is organized so that in the next section literature review is provided and then the model is introduced. After the model solution method is explained the finding analysis is done and finally conclusion is provided.

2. Literature Review

Exploration of some articles in the field of flow shop scheduling indicates that this problem is paid attention to by many researchers. Ayough and Khorshidvand (2019) in this study, a novel model is proposed for the design of a Cellular Manufacturing System (CMS) with the objective of minimizing costs, while considering a limited number of cells and workforce allocation. Due to the NP-hard nature of the problem, the metaheuristic algorithms of Simulated Annealing (SA) and Particle Swarm Optimization (PSO) are employed. Fernandez Viagas (2022) designed an acceleration procedure in order to minimize completion time. The procedure is included in the traditional frequent greedy algorithm. Fernandez Viagas et al (2022) consider the assembly flow shop scheduling problem that is an extension of 2 popular scheduling problems in literature. Wang et al (2023) focused on distributed com binational flow shop scheduling problem. In order to solve the problem a linear programming model of mixed integers is used. Newfeld et al (2023) categorize and describe literature about multi-objective mixed flow shop scheduling problems. Karimi Mamaghan et al (2022) seek to combine machine learning techniques with metaheuristics for the solution of the flow shop problem. Khatami et al (2023) indicate the mode of the 2-machine permutation flow shop is solvable. While the mode with more than 2 machines is considered as NPHARD. Han et al (2022) studied distributed flow shop scheduling problems with sequence depending on setup time and sought to minimize the energy consumption cost of the factory under resource balancing. Tasgetiren et al (2024) try to solve the blocking flow shop scheduling problem with makespan criterion using q-learning-based iterated greedy algorithms. Khurshid et al (2024) aim to hybrid evolution strategies with an iterated greedy algorithm for no-wait flow shop scheduling problems. Zhang et al (2024) provide effective social spider optimization algorithms for distributed assembly permutation flow shop scheduling problems in the automobile manufacturing supply chain. Perez Gonzalez and Framinan (2024) provide a review and classification of distributed permutation flow shop scheduling problems. Ayough et al. (2025) presented a nonlinear programming model for the job rotation problem in the Divisional Seru Production System (DSPS) and effectively optimizes flow time and workforce requirements using the Invasive Weed Optimization (IWO) metaheuristic. The results show that, compared with GAMS, IWO achieves significantly lower computational time with comparable accuracy, and that optimizing rotation periods reduces both cell imbalance and the required number of workers.

Table 1. Literature review

References	Year	Objective	Flow shop scheduling	No wait between the sequential operation of each job	completion time	Energy consumption
Vigas et al	2022	minimization of total delay in assembly flow shop scheduling	✓		✓	
Want et al	2023	minimization of completion time by focusing on distributed mixed flow shop scheduling problems by blocking constraints based on real production condition	✓		✓	
Newfled et al	2023	Categorization and description of literature about multi-objective mixed flow shop scheduling problem	✓			
Karimi Mamaghan et al	2022	Combination of machine learning and metaheuristics for solution of mixed optimization problem	✓		✓	
Perez Gonzalez and Framinan	2024	A review on distributed permutation flow shop scheduling review	✓			
Khatami et al	2022	2 machines permutation flow shop scheduling model	✓		✓	
Han et al	2022	Distributed flow shop scheduling problem with sequence depended on setup time.	✓			
Current research		Flow shop scheduling model by considering no wait between sequential tasks of each job	✓	✓	✓	✓

As we see in the above table there is no research that both study no wait between sequential tasks of each job in flow shop scheduling problem and aim to optimize completion time and energy consumption. Thus, we can say that current research since of innovation provision based on the gap includes innovation because the current research problem designs a flow shop scheduling model by considering no wait.

3. Model

In this section a flow shop scheduling model by considering no wait between sequential tasks is designed. The model has 2 objectives the first objective is to minimize completion time and the second one is to minimize energy. Next, the mode is explained and then the index, parameters, and variables are introduced finally objective function and constraints are described and finally solution method is explained. The production scheduling problem is considered as the set of production planning that is very important in various fields. The problems are changed during the time that it causes its existing conditions. In flow shop scheduling some tasks are done by some machines. Each job has some tasks and there are several assumptions for that. For example, sequence-dependent setup time, process time uncertainty or unavailability, and elitism of the machine are considered as some assumptions. But the main other assumption that should be serious is no wait or no stop in doing sequential tasks of each job in some industries and products is a requirement and is unavoidable. Energy consumption as an important subject in flow shop scheduling can be considered. The combination of both concepts meaning no wait and energy consumption seldom is considered in the literature review thus it can be considered as a research gap.

The fact that previous research doesn't pay attention to the combination of energy consumption and no wait and on the other hand there are some products whose production process follows a continuous procedure therefore combination of them can contribute to the study gap and is applicable in practice. In current research multi multi-objective model can be designable based on the above objectives. Model assumptions are as follows:

1. Tasks of each job should be continuous without stopping and continuously
2. The first objective is to minimize the completion time
3. The second objective is to minimize energy consumption
4. Energy consumption due to production time and idle time of the machine is considered

Table 2. Signature of sets

Signature	Sets
$i. i$	Machine
$j. j'$	Tasks
k	Position
n	Number of tasks that should be planned
m	The number of machines is no wait flow shop.
Signature	Parameters
$p_{j.i}$	Processing time of task j on machine i
$o_{j.i}$	Operation of task j on machine i
PE_i	Energy consumption indicator of machine i
LE_i	Energy consumption indicator of machine I in idle time
π	A Feasible scheduling
π_k	The task in position k of a feasible scheduling
$d_{\pi_{k-1}.\pi}$	Minimal delay in machine one between starting 2 sequential tasks in position k and k-1 based on no wait constraints
C_j	Completion time of task j
$C_{k.i.j}$	Completion time of task j in position k on machine i
C_{max}	Completion time
$X_{j.k}$	If task j occupy position k 1 and else zero
$C_{max}(\pi)$	Completion time of a certain scheduling
B_i	Total time of task for machine i
L_i	Total idle time for machine i
TEC	Energy consumption
Signature	Variable
$X_{j.k}$	Binary variable, if task j assigned to position k is equal 1 O.W is 0.

Objective function:

In this section, both objective functions are defined.

$$\min C_{max} = \sum_k^K \sum_i^m C_{k.i} \quad (1)$$

$$\min TEC = \sum_i^m B_i\, PE_i + L_i LE_i \quad (2)$$

The objective function (1) tries to minimize the completion time. The second objective in equation (2) aims to minimize energy consumption.

Constraints:

$$C_{k.i} = C_{k.i-1} + \sum_{j=1}^n X_{j.k}\, p_{j.i}. \forall k. i > 1 \quad (3)$$

$$C_{k.i} \geq \sum_{j=1}^n X_{j.k}\ \ p_{j.i} \forall k. i = 1 \quad (4)$$

$$C_{k.i} = C_{k-1.i} + \sum_{j=1}^n X_{j.k}\, p_{j.i}. \quad \forall k > 1,\ i \in m \quad (5)$$

$$C_{max} = C_{k.m}\ \forall k \quad (6)$$

$$\sum_{k=1}^n X_{j.k} = 1\ \forall k \quad (7)$$

$$B_i = \sum_j^n P_{i.j} \ \forall i \quad (8)$$
$$L_i = C_{max} - B_i \quad (9)$$
$$TEC = \sum_i^m B_i \, PE_i + L_i LE_i \quad (10)$$
$$C_{k.i} \geq 0 \ \ \forall k.i \quad (11)$$
$$TEC \geq 0 \quad (12)$$
$$B_i \geq 0 \quad (13)$$
$$L_i \geq 0 \quad (14)$$
$$C_{max} \geq 0 \quad (15)$$
$$X_{j.k} \in \{0.1\} \ \forall j.k \quad (16)$$

The constraint (3) determines the relationship between the completion times of each task on 2 sequential machines. The constraint (4) ensures the completion time of each task on machine one is larger or equal to its processing time on machine. The constraint (5) provides a relationship between the completion times of 2 consequential tasks on each machine. Constraint (6) determines the makespan. Constraint (7) ensures that each task is assigned to exactly one position. The constraint (8) describes the total time of the machine task. The constraint (9) indicates machine idle time. The constraint (10) describes the total energy consumption. The constraints (11-15) define each task as nonnegative. The constraint (16) determines binary variables.

3.1. Solution method

In this section, the solution method is explained. The solution procedure consists of **two stages**. In stage one using GAMS software the problem is solved in a small dimension and in the second stage the metaheuristic algorithm is used for solution in a big dimension. The validity of the model is done by solution in a small dimension. The used algorithm for the model solution is the NSGA-II algorithm. Its description is provided in the next section.

3.1.1. NSGA-II algorithm

NSGA-II algorithm is one of the most applicable and powerful algorithms for the solution of multi-objective problems and its efficiency is proven. In this algorithm solutions that are not dominated by any other solution are assigned a higher rank. The answers are prioritized based on the fact that there are several answers. Elitism for answers is assigned based on their ranking and not the dominance of other answers. The elitism sharing method is used for neighbor answers in order to regulate answer scattering desirably and for the answers to be distributed uniformly in search space. Due to the high sensitivity of the performance method and quality of the algorithm answer to elitism sharing and other parameters, the second version of NSGA is introduced. Besides all the efficiencies of NSGA II, we can call it the model of shaping many multi-objective optimization algorithms. The algorithm and its method in dealing with multi-objective optimization problems are used by various people for the creation of newer multi-objective optimization algorithms. No doubt the algorithm is one of the principal members of multi-objective optimization algorithms that they can be called the second generation of such methodologies. The significant characteristics of this algorithm include:

1- Definition of congestion distance as an alternative characteristic for methods like elitism sharing
2- Using from tournament selection operator
3- Storage and archiving of non-dominated answers that get in previous stages of the algorithm

3.1.2. Parameter adjustment

In this section, the parameters of the metaheuristic algorithm are tuned using the Taguchi L9 method. For using this method first for algorithm parameters 3 levels are defined. Then experiments are executed in this algorithm per all possible combinations. The proposed value for the parameters of the algorithm is according to the following table

Table 3. Parameter adjustment of NSGAII algorithm

Parameters	Each level value		
	Level 1	Level 2	Level 3
Population size (PS)	50	55	60
Crossover rate (CR)	0.9	1	1
Mutation rate (MR)	0.18	0.25	0.3
Maximum iterations (Max_iter)	130	150	170

Then by design L9 Taguchi various experiments are created and for each one NSGA-II algorithm is executed. Results for the MID indicator are provided in the table 4. In this table, all possible modes per various levels that are considered for factors of the algorithm are shown. For example, in experiment one all factors per lowest level exist in the model. In the second experiment factor PS with lowest level value and other factors with average level value exist. Other modes based on the permutation rule are completed in statistics. By execution of each experiment, a calculation of the indicator value of MID response level using this indicator is estimated.

Table 4. Response variable value in Taguchi technique for NSGA-II

Run	Algorithm parameter				MID indicator
	PS	CR	MR	Max iter	
1	1	1	1	1	0.555
2	1	2	2	2	0.668
3	1	3	3	3	0.544
4	2	1	2	3	0.423
5	2	2	3	1	0.589
6	2	3	1	2	0.676
7	3	1	3	2	0.604
8	3	2	1	3	0.885
9	3	3	2	1	0.696

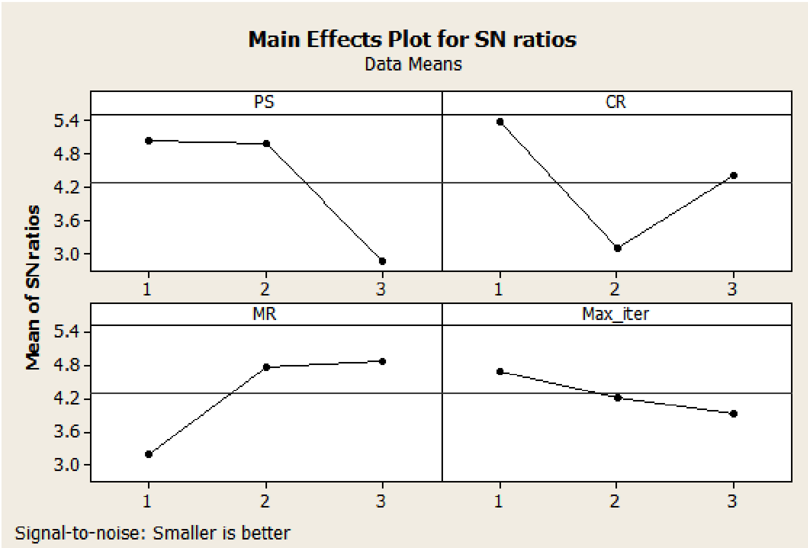

Figure 1. Output of MINITAB for Taguchi method in NSGA-II

According to the above figure, the best value for each parameter is determined by other examples with such values of the algorithm executed.

Table 5. Optimal values for parameters

Parameter	Optimal value
Population size (PS)	50
Crossover rate (CR)	0.85
Mutation rate (MR)	0.09
Maximum iterations (Max iter)	130

The steps of research are displayed in the following chart

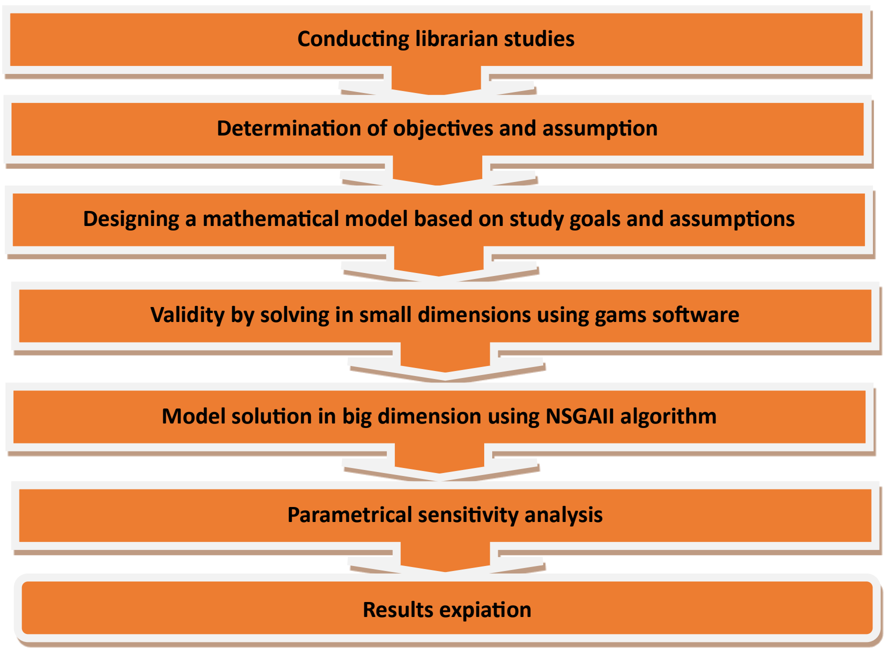

Figure 2. Steps of research

4. Findings

In this section using gams and Matlab software the modal in the previous section is implemented. Firstly validity of the model is done and then 2 methods are compared. Parametrical sensitivity analysis is done. Validity is done by the solution of the model in various dimensions and exploration of them on values of objective function is performed. Firstly model dimensions are provided in Table 6.

Table 6. Model dimension

Row	Number of machines	Number of tasks	Number of positions	Number of scheduled tasks	Number of machines in no wait flow shop
1	1	2	1	2	1
2	2	4	2	4	1
3	3	6	2	6	1
4	4	8	3	7	1
5	5	10	3	8	2
6	6	12	4	10	2
7	7	14	4	12	2
8	8	16	5	15	2
9	9	18	5	16	3
10	10	20	6	18	3
11	11	22	6	20	3
12	12	24	7	21	3
13	13	26	7	22	3
14	14	28	8	23	3
15	15	30	8	25	4
16	16	32	9	30	4
17	17	34	9	32	4
18	18	36	10	33	4
19	19	38	10	35	4
20	20	40	10	37	4

As we see in the above table 20 problems are introduced and each problem dimension is varied from the previous one. Model solution results are provided in the table 7.

Table 7. Results of the model solution in various dimensions

Problem	Completion time	Energy consumption	Computing time
1	1410	4555413	5
2	1584	4556952	15
3	1756	4558023	23
4	1931	4559482	31
5	2081	4560781	40
6	2255	4561826	48
7	2430	4563181	56
8	2616	4565005	64
9	2812	4566655	70
10	2968	4568569	75
11	......	......	low memory
12	......	......	low memory
13	......	......	low memory
14	......	......	low memory
15	......	......	low memory
16	......	......	low memory
17	......	......	low memory
18	......	......	low memory
19	......	......	low memory
20	......	......	low memory

As we observe in the above table by increasing the dimension the values of the objective function are increased too and this indicates that the objective function reacts to dimension increase and this implies model validity. The main note is that GAMS software only can solve the model until the tenth problem and from the tenth problem we encounter low memory error. The error shows that for the solution of problems, we should use from metaheuristic algorithm. In order to explain the validity clearly, objective function results are shown in Figures 3-5.

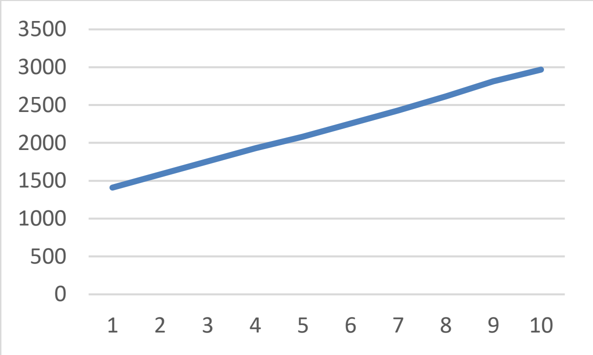

Figure 3. Increasing completion time by dimension increase

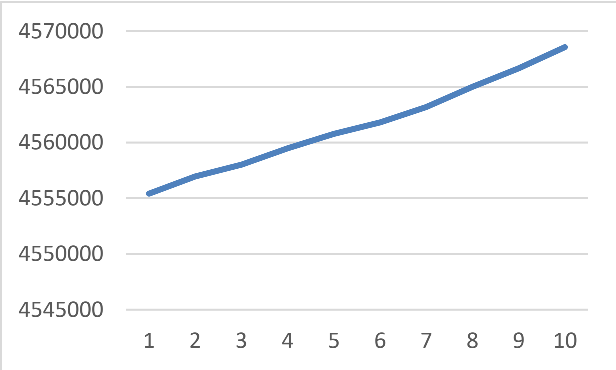

Figure 4. Increasing energy consumption by dimension increase

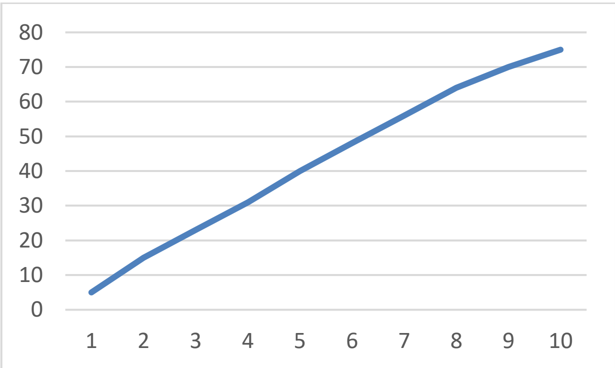

Figure 5. Increasing computing time by dimension increase

Based on the above Figures we can perceive that the model is valid and we can optimize the model in bigger dimension. Firstly comparison between methodologies with Gams and Matlab is done and a gap between the 2 methods is provided. In the following charts, the comparison is done.

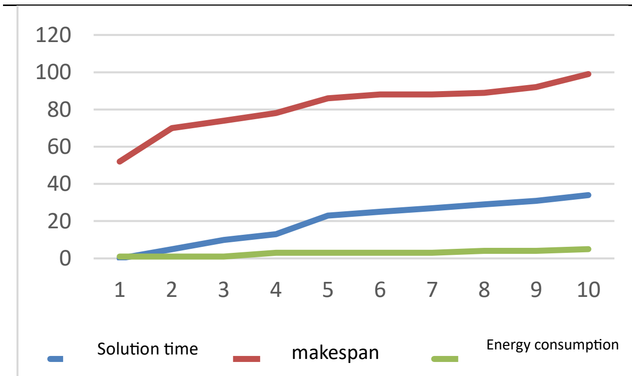

Figure 6. analysis of the gap between GAMS and MATLAB results

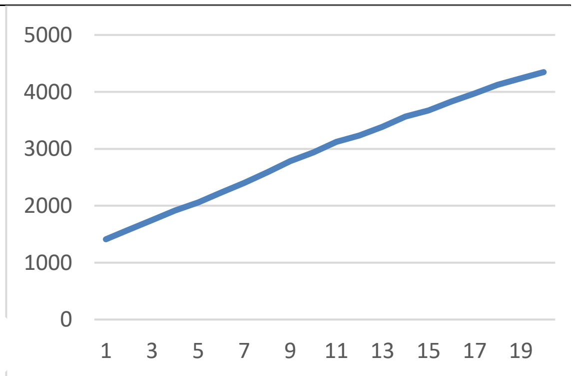

Figure 7. results of completion time by NSGA-II

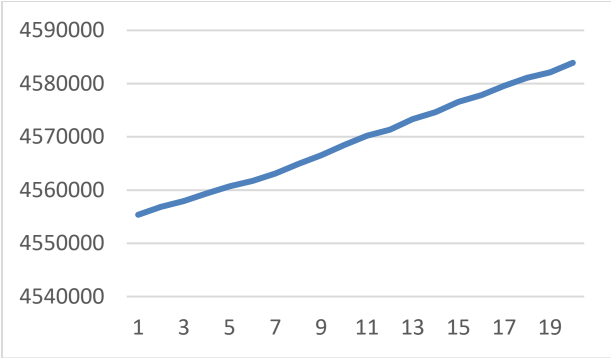

Figure 8. results of energy consumption by NSGA-II

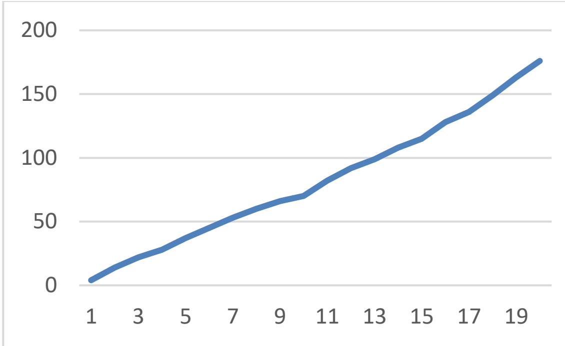

Figure 9. results from computing time by NSGA-II

As we observe in Figure 6 by increasing dimension the gap between GAMS and MATLAB results is increased and the results of NSGA-II are in a better position in comparison with GAMS software. The highest gap is about energy consumption and completion time is in second position. Based on Figure 7 we can see that completion time is increased linearly in each problem the previous one. While the NSGA-II algorithm is able to solve the model until the final problem. In Table 7 we can observe that energy consumption increases with dimension increases and the algorithm is successful in solving the problem until the final problem. In the next parametric sensitivity analysis is implemented. By sensitivity analysis, we can perceive which parameter has the higher impact on the model. For this goal impact of each parameter is increased to 10 percent in order to explore results.

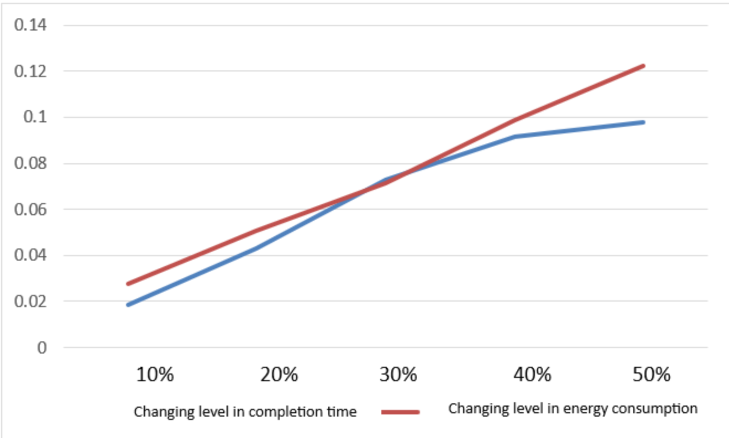

Figure 10. Sensitivity analysis of task processing time

As we can observe in Figure 10 task processing time can lead to increased completion time and energy consumption in the model. The impact of this parameter on energy consumption is linear while the impact on completion time is not linear. By less increasing we can see the decline of impact and total processing time impact on energy consumption is higher than completion time.

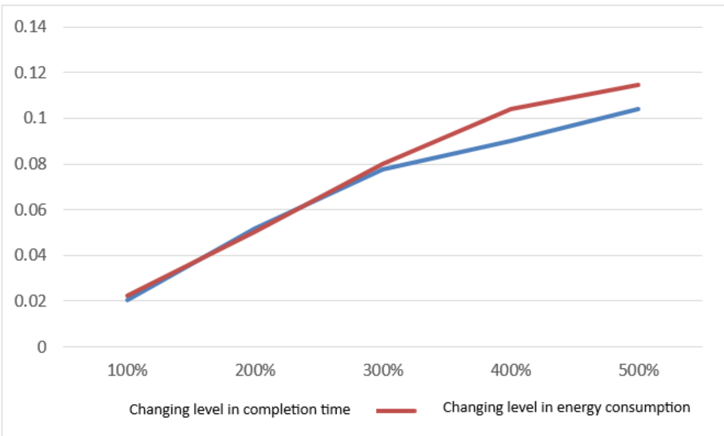

Figure 11. Sensitivity analysis of the number of tasks for each job

In Figure 11 impact of task number for each job is analyzed in no wait condition. Results indicate that increasing the number of tasks naturally increases completion time; however, its impact on energy consumption is more pronounced. Therefore, we can say that the number of tasks on energy consumption is stronger than the completion time, and the influencing pattern is non-linear.

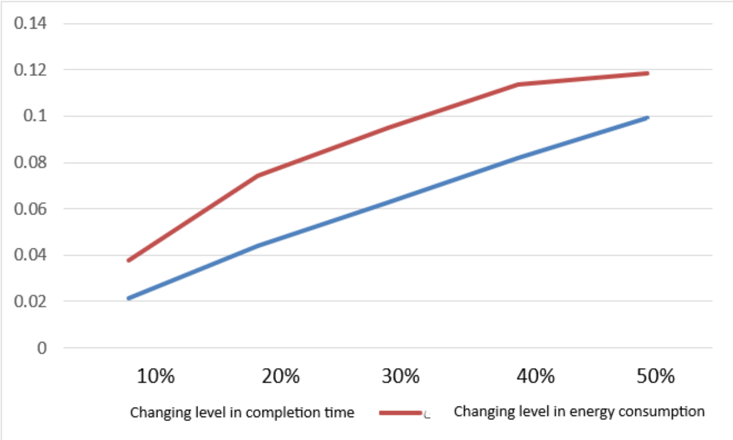

Figure 12. Sensitivity analysis of energy consumption

In Figure 12, we can see that the energy consumption of machines in active mode can cause total energy consumption non-linearly and completion time can be increased linearly. The difference between the above chart and the previous ones is that the distance between energy consumption and completion time is increased and, in this situation, influence on completion time is linear. In other words, after less increase in total energy consumption, it would be declined. But this is not true about completion time and increasing can be seen linearly.

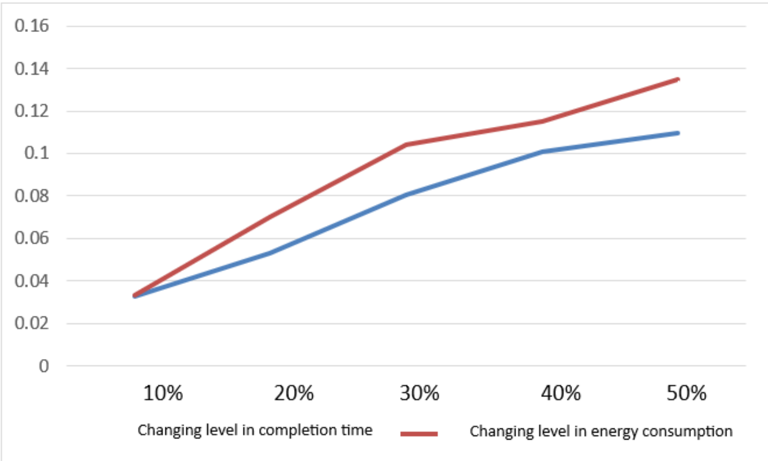

Figure 13. Sensitivity analysis of energy consumption in idle time

Besides energy consumption, there is another type of energy consumption in idle time of machines that their analysis indicates this parameter can increase the total energy consumption and completion time synchronically. Of course, the level of influence on energy consumption is higher than completion time.
In the next, we compare the influencing level of various parameters based on the following charts.

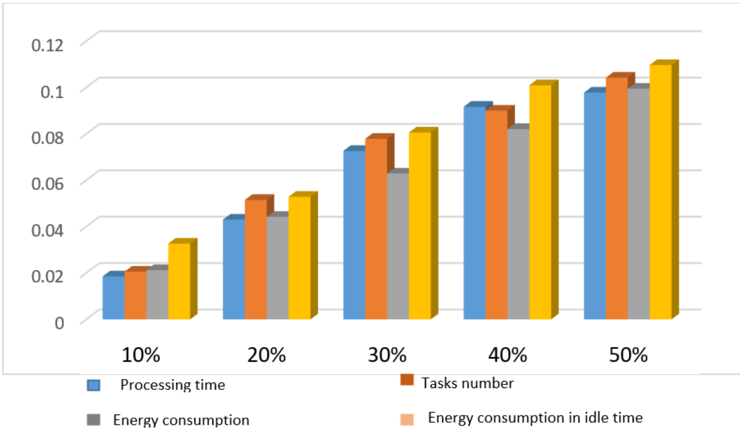

Figure 14. Comparison of influencing parameters on completion time

Based on Figure 14 we can perceive that energy consumption in idle time has the highest impact in various conditions on completion time. The second most influencing factor is task number. In other words the more the task number, it is expected the more increased completion time. The least influencing parameter among existing parameters is energy consumption in activity time.

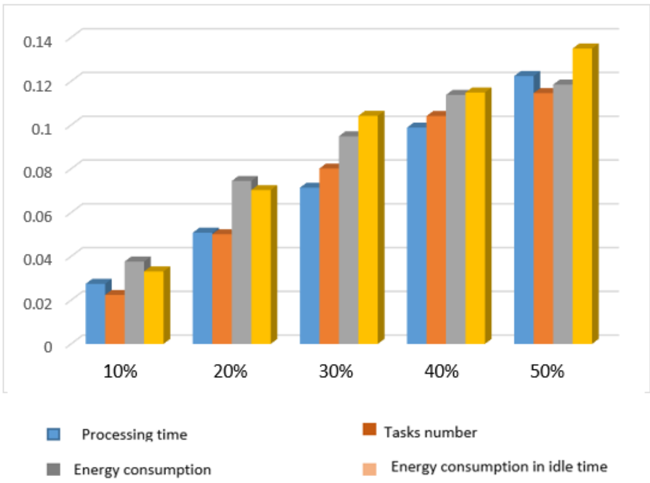

Figure 15. Comparison of influencing parameters on energy consumption

We can say the more such parameters are increased the more energy consumption is increased in idle time. While in 10 and 20 percent increasing the most influencing factors include energy consumption in task time. Processing time and number of tasks have less impact on energy consumption than 2 parameters of energy consumption in idle time and energy consumption in task time.

5. Conclusion

Results from findings indicate that task processing time can increase completion time and energy consumption synchronically. This is while if tasks are increased, completion time can be increased too but energy consumption would be increased higher. Energy consumption parameters can influence completion time more significantly than total energy consumption itself. Also, energy consumption in idle time of machines can increase total energy consumption and completion time synchronically that total energy consumption is influenced higher than completion time. Suggestions in this paper can be provided as follows:

1. The current model considers the energy consumption and completion time that future research can extend the model to another goal like risk or resilience
2. The current model is solved in certain conditions that in future research task processing time parameters can be considered uncertain.
3. The current model used a metaheuristic algorithm for a model solution that future research can use from other algorithms and compare them based on multi-objective measures.
4. In current research, machines are considered available and future research can consider the unavailability of machines.

References

Alfaro-Fernández, P., Ruiz, R., Pagnozzi, F., & Stützle, T. (2020). Automatic algorithm design for hybrid flowshop scheduling problems. European Journal of Operational Research, 282(3), 835-845.

Ayough, A., adeghi Nouri, F., Khorshidvand, B., Farhadi, F. (2025). Modeling workers rotation in divisional seru production systems. Computers & Industrial Engineering, 2025, 111141.

Ayough, A., Khorshidvand, B. (2019). Designing a manufacturing cell system by assigning workforce. Journal of Industrial Engineering and Management, 12(1), 13-26.

Fernandez-Viagas, V. (2022). A speed-up procedure for the hybrid flow shop scheduling problem. Expert Systems with Applications, 187, 115903.

Fernandez-Viagas, V., Talens, C., & Framinan, J. M. (2022). Assembly flowshop scheduling problem: Speed-up procedure and computational evaluation. European Journal of Operational Research, 299(3), 869-882.

Han, X., Han, Y., Zhang, B., Qin, H., Li, J., Liu, Y., & Gong, D. (2022). An effective iterative greedy algorithm for distributed blocking flowshop scheduling problem with balanced energy costs criterion. Applied Soft Computing, 129, 109502.

Karimi-Mamaghan, M., Mohammadi, M., Pasdeloup, B., & Meyer, P. (2023). Learning to select operators in meta-heuristics: An integration of Q-learning into the iterated greedy algorithm for the permutation flowshop scheduling problem. European Journal of Operational Research, 304(3), 1296-1330.

Khatami, M., Salehipour, A., & Cheng, T. C. E. (2023). Flow-shop scheduling with exact delays to minimize makespan. Computers & Industrial Engineering, 183, 109456.

Khurshid, B., Maqsood, S., Khurshid, Y., Naeem, K., & Khalid, Q. S. (2024). A hybridization of evolution strategies with iterated greedy algorithm for no-wait flow shop scheduling problems. Scientific Reports, 14(1), 2376.

Neufeld, J. S., Schulz, S., & Buscher, U. (2023). A systematic review of multi-objective hybrid flow shop scheduling. European Journal of Operational Research, 309(1), 1-23.

Perez-Gonzalez, P., & Framinan, J. M. (2024). A review and classification on distributed permutation flowshop scheduling problems. European Journal of Operational Research, 312(1), 1-21.

Tasgetiren, M., Kizilay, D., & Kandiller, L. (2024). Solving blocking flowshop scheduling problem with makespan criterion using q-learning-based iterated greedy algorithms. Journal of Project Management (Canada), 9(2).

Wang, Y., Wang, Y., Han, Y., Li, J., Gao, K., & Nojima, Y. (2023). Intelligent optimization under multiple factories: Hybrid flow shop scheduling problem with blocking constraints using an advanced iterated greedy algorithm. Complex System Modeling and Simulation, 3(4), 282-306.

Wang, Z. Y., Pan, Q. K., Gao, L., & Wang, Y. L. (2022). An effective two-stage iterated greedy algorithm to minimize total tardiness for the distributed flowshop group scheduling problem. Swarm and Evolutionary Computation, 74, 101143.

Zhang, W., Hao, J., & Liu, F. (2024). Effective social spider optimization algorithms for distributed assembly permutation flowshop scheduling problem in automobile manufacturing supply chain. Scientific Reports, 14(1), 6370.