Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 9, S. 32-39, 3, 2025
© Telif hakkı IJANSER'e aittir
Araştırma Makalesi

https://as-proceeding.com/index.php/ijanser
ISSN:2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 9, pp. 32-39, 3, 2025
Copyright © 2025 IJANSER
Research Article

Particle Swarm Optimization Algorithm for Distributed Flow Shop Scheduling Problem

Ümit TOK*[1], Züleyha YILMAZ ACAR[2]

[1]*Department of Computer Engineering, Institute of Sciences, Selçuk University, Konya, Türkiye*
[2]*Department of Computer Engineering, Faculty of Technology, Selçuk University, Konya, Türkiye*

[1]*(ORCID: https://orcid.org/0000-0002-6434-2363)*
[2]*(ORCID: https://orcid.org/0000-0002-4488-478X)*

**tokumitaydin@gmail.com*

(Received: 21 February 2025, Accepted: 28 February 2025)

(4th International Conference on Contemporary Academic Research ICCAR 2025, February 22-23, 2025)

ATIF/REFERENCE: Tok, Ü. & Acar, Z. Y. (2025). Particle Swarm Optimization Algorithm for Distributed Flow Shop Scheduling Problem. *International Journal of Advanced Natural Sciences and Engineering Researches*, 9(3), 32-39.

Abstract – The increasing prevalence of multi-factory models has led to the emergence of more complex problems. To solve these large and complex issues, our study utilizes the Distributed Flow Shop Scheduling Problem (DFSP) with Particle Swarm Optimization (PSO) for inter-factory transportation. DFSP aims to optimize job sequencing and inter-factory transportation times to minimize the total completion time (makespan). Solving this problem is particularly crucial in improving resource-sharing efficiency, minimizing costs, and ensuring quick market response in complex and diverse production environments. Experiments demonstrate that PSO optimizes job sequences and balances factory assignments, achieving a minimum makespan of 10 units. A Gantt chart was used to visualize resource utilization and planning efficiency for a better understanding of the solution. Compared to methods such as Genetic Algorithms and Simulated Annealing, PSO offers advantages in convergence speed and computational efficiency.

This study shows that PSO is one of the best methods applicable to real-world planning problems and logistical challenges. Future studies may combine PSO with other metaheuristic methods for faster resolution of dynamic planning problems.

Keywords –Optimization Algorithms, PSO, DFSP, Makespan.

I. INTRODUCTION

In the past, single-factory models were prevalent, but today, factories with multiple branches are rapidly increasing. With global economic growth, speed and cost have become crucial competitive factors among businesses [1]. Companies with multiple factory branches seek methods to work faster and at lower costs. Multi-factories, each with unique capacities, machines, and production lines, create complex workflows due to inter-factory transportation operations [2]. Enhancing factory efficiency, reducing costs, and delivering products promptly are frequently encountered real-world problems. Being a faster method that can provide computational efficiency even for complex problems, the PSO algorithm facilitates the solution of such problems [3]. To address time and cost concerns effectively, Distributed Flow Shop Scheduling Problems (DFSP) were developed [4]. DFSP can be considered the best optimization problem for various real-world applications and workflows involving multiple factories. Typically, artificial intelligence-based heuristic optimization methods are preferred to apply this problem. These methods help achieve optimal performance in tasks like sequencing jobs and minimizing time and costs. DFSP emerges as a more complex extension of Flow Shop Scheduling Problems (FSP) [5].

In recent years, DFSP has become a widely chosen method by both businesses and researchers due to the growing economy and increasing competition [6]. Optimizing and sequencing limited resources and workflows with DFSP is most effectively achieved using the heuristic optimization method, Particle Swarm Optimization (PSO). Solving this problem is critical for sharing resources efficiently, minimizing costs, and responding quickly to market demands, especially in complex production environments [7].

Optimization algorithms are typically inspired by nature, taking cues from animal behavior. Animals moving in swarms often achieve their goals more easily through seemingly random movements for food and safety [3]. PSO, a heuristic optimization algorithm, was developed in 1995 by Kennedy and Eberhart by studying the behavior of fish and insects. The algorithm imitates the social behavior of swarming animals, where each animal is referred to as a particle [8]. Although particles within the swarm appear to move irregularly, they exhibit a clustering behavior. Each particle in the swarm carries speed and position information and adjusts its position based on previous experiences to reach the best position. PSO fundamentally involves the position of individuals in the swarm approaching the best position within the group [10]. Many other optimization algorithms, such as Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO), also draw inspiration from nature [11].

This study aims to develop a solution to the DFSP problem that minimizes the total completion time (makespan). The PSO algorithm is chosen as the most suitable method for flow shop scheduling problems, offering faster results in complex multi-factory workflows and optimization problems requiring constant parameter adjustments.

II. MATERIALS AND METHOD

A. Problem Definition and General Description

The problem was defined using a scheduling method. Scheduling problems are critical for optimizing workflows and ensuring the efficient use of resources. Multi-factory scheduling problems, in particular, require coordinating both production line processes and inter-factory job transfers [12]. The solution to DFSP, a complex sub-category of these problems, depends on many variables.

In this problem, job sequencing in multiple factories and transportation times between factories must be considered. The objective is to minimize the total completion time (makespan). PSO initializes a particle swarm, with each particle searching for solutions based on its best value, ultimately finding the optimal solution [13]. This process requires minimal computation, making it computationally faster than many evolutionary algorithms.

B. Mathematical Definition

- Decision Variables are defined as follows:
  S: Sequence of jobs.
  F: Array indicating which factory processes the jobs.
  Tij: Transportation time between factory i and factory j.

- The objective function aims to minimize the total completion time, known as makespan:
  Minimize: $C_{max} = \max\{C_{i,m}\}$, here $\forall i \in$ Factories, $\forall m \in$ Machines.

Ci, m: Completion time of machine m in factory i.

- Constraints:
  Each job must be processed in one factory:
  $\sum_j F_{ij} = 1$, here $\forall i \in İ \in$ Jobs.

Job sequencing must be maintained machine-dependently.
Inter-factory transportation times must be included in the total time.

- The PSO algorithm follows these steps for solving the DFSP problem:

- Particle Representation:
  Each particle is defined by two components: job sequence (S) and factory assignment (F).

- Velocity Update:
  $v_i(t+1) = w \cdot v_i(t) + c1 \cdot r1 \cdot (P_{best,i} - x_i(t)) + c2 \cdot r2 \cdot (g_{best} - x_i(t))$.

- Position Update:
  $x_i(t+1) = x_i(t) + v_i(t+1)$.

w: Inertia coefficient.
c1,c2: Individual and social learning coefficients.
r1,r2: Random values between [0,1].
p_{best}, g_{best}: Local and global best positions.

Fitness function focuses on minimizing the total completion times (Cmax). The performance of particles is evaluated based on this value.

- Algorithm Parameters:
  Number of particles: 10.
  Maximum iterations: 50.
  Inertia coefficient (w): 0.3.
  Individual and social learning coefficients (c1,c2): 1.5.

- Dataset:
  6 jobs, 3 machines, and 2 factories.

- Processing Times matrix.

$$\begin{array}{c} \begin{bmatrix} 2 & 3 & 1 & 2 & 3 & 1 \\ 1 & 2 & 2 & 1 & 3 & 2 \\ 3 & 1 & 2 & 3 & 2 & 1 \end{bmatrix} \end{array} \begin{array}{l} \text{Machine 1} \\ \text{Machine 2} \\ \text{Machine 3} \end{array}$$

In the given matrix, each row represents a machine. Each row contains a total of 6 jobs, starting from Job 0 to Job 5. The numbers indicate the processing times for these jobs. The notation can be summarized as follows: Machine X completes Job Y in Z units of time.

For example:
Machine 1 takes 2 units of time to complete Job 0.
Machine 2 takes 1 unit of time to complete Job 3.
Machine 3 takes 1 unit of time to complete Job 5.

- Transportation Times matrix between factories.

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

III. RESULTS

A. Experimental Settings

In this study, the DFSP problem was solved using the PSO algorithm. The algorithm's performance was tested with the following parameters:

Number of particles: 10, 20, 50, 100
Maximum iterations: 50, 70, 90, 100
Inertia coefficient (w): 0.3
Individual learning coefficient (c1): 1.5
Social learning coefficient (c2): 1.5

The dataset consisted of 6 jobs, 3 machines, and 2 factories, utilizing previously defined processing and transportation times.

B. Obtained Results

The PSO algorithm has been tested with different numbers of iterations and particles to minimize the makespan of the DFSP problem. The solutions obtained from the algorithm are as follows in Table 1. In Table 1, an example result is shown with 10 particles and 50 iterations, and the values represented by it are explained.

Table 1. Results obtained with 10 particles and 50 iterations

Best job sequence	**Factory assignments**	**Lowest makespan**
[5 4 1 3 2 0]	[1 0 1 1 0 0]	11.0

The best job sequence here indicates the order in which the jobs are processed. For example, it shows that job number 5 is processed first, followed by job number 4, then job number 1, and subsequently the other jobs in order. The best factory assignments show which factory each job is processed in. According to the results, the jobs are processed in the following factories:

Job number 5 is processed in Factory 1.

Job number 4 is processed in Factory 0.

Job number 1 is processed in Factory 1.

Job number 3 is processed in Factory 1.

Job number 2 is processed in Factory 0.

Job number 0 is processed in Factory 0.

In trials with different particle numbers, the following results were obtained as Table 2.

Table 2. Results obtained with different values

Particle numbers	Iteration numbers	Lowest makespan
10	100	11
10	90	10
10	70	10
20	100	11
20	90	10
20	70	10
50	100	10
50	90	10
50	70	11
100	100	10
100	90	10
100	70	10

In Table 2, when the number of particles is 100, the lowest makespan always results in 10. However, during iterations where one particle is replaced by another, the makespan can increase and take values such as 11 or 12. In Table 3, the best cycle time and the best factory assignments are observed across iterations with numbers continuing up to 100. In this case, the lowest makespan value for the job sequence and factory assignments is always 10.

Table 3. Results obtained with different particle numbers

Particle numbers	Iteration numbers	Best job sequence	Factory assignments	Lowest makespan
100	100	[3 5 0 2 1 4]	[0 0 1 0 1 1]	10
100	90	[3 0 2 5 1 4]	[1 0 0 0 1 0]	10
100	70	[2 5 0 3 4 1]	[1 0 0 0 1 0]	10

When we create the convergence graph according to particle preference, it can be seen in Fig. 1 below that the makespan value is always 10 for the preference of 100 particles, while it can vary in other cases.

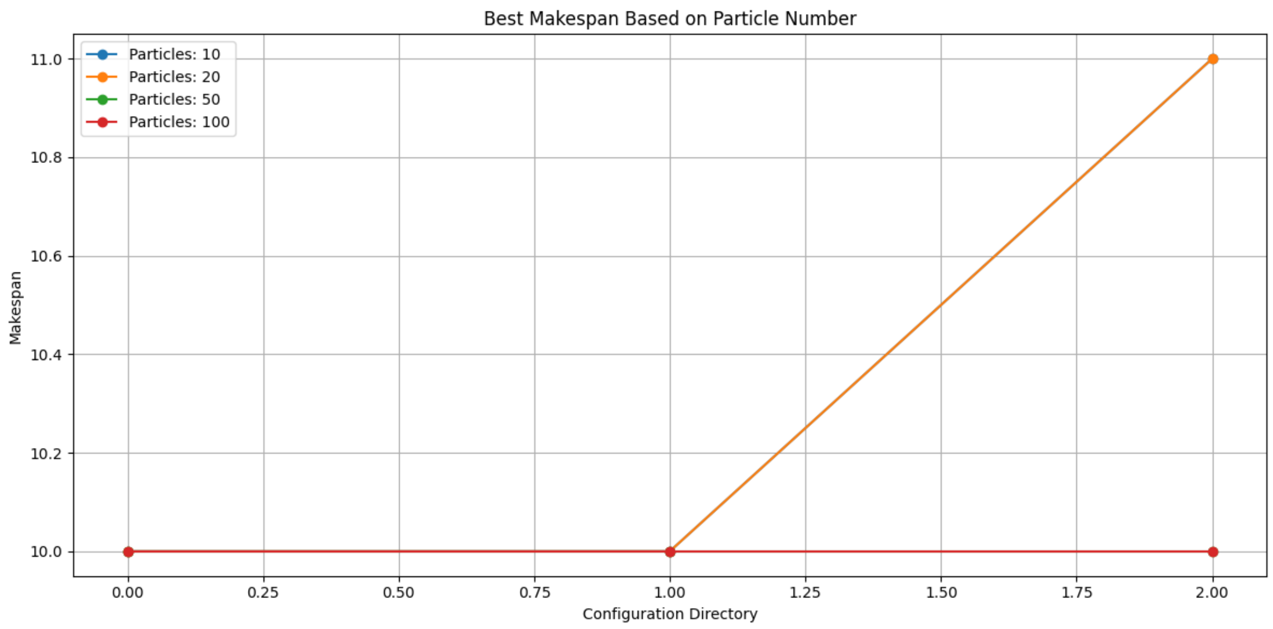

Fig. 1 Makespan according to particle preference

To better understand the results, the solution was visualized using the Gantt chart shown in Fig 2. The Gantt chart displays the timeframes in which jobs were processed on each machine and inter-factory transfers. In the Gant chart, we can better observe the optimal job sequence and the preferred factories.

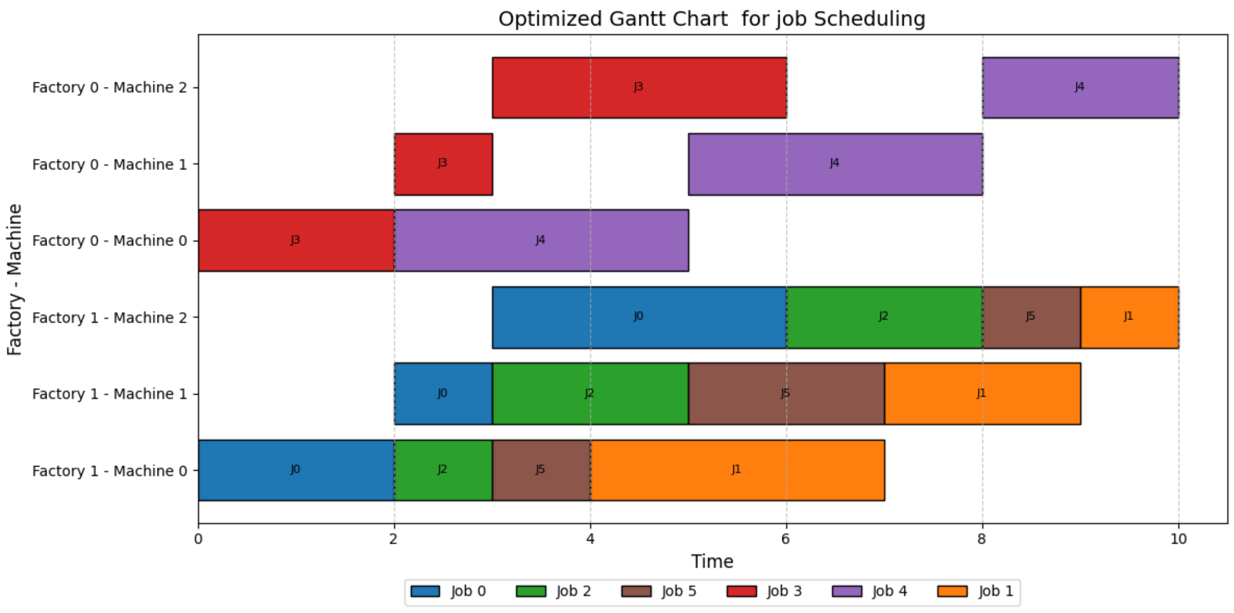

Fig. 2 Gantt chart

In the graph, the Horizontal Axis (Time) represents the timeline. It shows the time intervals during which jobs are processed in factories and machines. The Vertical Axis (Factory - Machine) displays the hierarchy of factories and machines. For example:

- Factory 0 - Machine 0: Machine 0 in Factory 0.
- Factory 1 - Machine 2: Machine 2 in Factory 1.

In the chart, different colors represent each job (Job). Labels indicate the job definitions (e.g., "J4" for Job 4). The chart clearly shows how jobs were balanced across factories and machines. Efficient resource utilization and minimized total completion times were observed. The best solution was typically obtained within the first 20 iterations, demonstrating the algorithm's ability to rapidly explore the solution space. The parameter combinations used positively impacted the algorithm's stability and success rate. PSO proved to be an effective method for minimizing total completion times.

These results demonstrate the effectiveness of the PSO algorithm in solving the DFSP problem. Strengths and weaknesses of the algorithm will be evaluated in the next Discussion section.

IV. DISCUSSION

This study aimed to solve the Distributed Flow Shop Scheduling Problem with Inter-Factory Transportation using the PSO algorithm. The results obtained demonstrate the algorithm's effectiveness in minimizing total completion times (makespan). The algorithm successfully optimized total completion times and effectively explored the solution space. Balanced job sequencing and factory assignments ensured efficient resource utilization. The algorithm quickly achieved optimization by finding the best solution in early iterations. The Gantt chart illustrates how jobs were balanced across factories and machines. An optimized solution was achieved by considering transportation times and machine conflicts. PSO effectively struck a balance between local and global best solution points, achieving good results rapidly. The parameters were adaptable for different problems. Concurrent processing of multiple jobs without conflict utilized resources efficiently and reduced total time. However, PSO sometimes struggles with local optima instead of achieving the global optimum. Inertia coefficient ("w"), individual ("c1"), and social ("c2") learning coefficients play critical roles in algorithm performance.

V. CONCLUSION

This study addressed the Distributed Flow Shop Scheduling Problem with Inter-Factory Transportation (DFSPT) using the Particle Swarm Optimization (PSO) algorithm. The problem involved sequencing jobs and optimizing transportation times across multiple factories to minimize total completion times (makespan). As a heuristic optimization method, PSO provided an effective solution for complex problems like DFSPT. The algorithm effectively optimized multi-factory workflows, minimizing total completion times. The Gantt chart demonstrated the balance and efficiency of the solution. The algorithm's rapid convergence capability achieved good results in early iterations. The balanced search mechanism between local and global best points effectively explored the solution space. The optimized job sequence and factory assignments provided a tailored solution for the given dataset. PSO was observed to have advantages over other methods in the literature.

This study highlighted the potential of PSO for solving complex optimization problems encountered in real-world scenarios like DFSPT. Future studies may enhance the algorithm's adaptability to changing conditions. Combining PSO with other methods such as Genetic Algorithm (GA) or Tabu Search (TS) may lead to more effective hybrid models. The algorithm's performance can be tested under limited constraints in dynamic and real-time systems such as logistics, production, and distribution.

In conclusion, this study provides significant contributions to the application of PSO in scheduling problems. Considering its position in the literature and practical applications, the algorithm is shown to be an effective solution method, especially in production and logistics fields.

REFERENCES

[1] J. Zhou, T. Meng, and Y. Jia, "Modelling and optimization of a distributed flow shop group scheduling problem with heterogeneous factories," Computers & Industrial Engineering, vol. 198, p. 110635, 2024.

[2] G. Zhang, B. Liu, L. Wang, D. Yu, and K. Xing, "Distributed co-evolutionary memetic algorithm for distributed hybrid differentiation flowshop scheduling problem," IEEE Transactions on Evolutionary Computation, vol. 26, no. 5, pp. 1043-1057, 2022.

[3] M. Y. Özsağlam and M. Çunkaş, "Particle swarm optimization algorithm for solving optimization problems," Polytechnic Journal, vol. 11, no. 4, pp. 299-305, 2008.

[4] P. Perez-Gonzalez and J. M. Framinan, "A review and classification on distributed permutation flowshop scheduling problems," European Journal of Operational Research, vol. 312, no. 1, pp. 1-21, 2024.

[5] T. Becker, J. Neufeld, and U. Buscher, "The distributed flow shop scheduling problem with inter-factory transportation," European Journal of Operational Research, 2024.

[6] C. Lu, Q. Liu, B. Zhang, and L. Yin, "A Pareto-based hybrid iterated greedy algorithm for energy-efficient scheduling of distributed hybrid flowshop," Expert Systems with Applications, vol. 204, p. 117555, 2022.

[7] W. Zhang, C. Li, M. Gen, W. Yang, Z. Zhang, and G. Zhang, "Multi-objective particle swarm optimization with direction search and differential evolution for distributed flow-shop scheduling problem," Mathematical Biosciences and Engineering, vol. 19, pp. 8833-8865, 2022.

[8] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," Soft Computing, vol. 22, no. 2, pp. 387-408, 2018.

[9] İ. Aydın and B. Aşıcı, "A particle swarm optimization-based ensemble classifier method for human activity recognition," Fırat University Journal of Engineering Sciences, vol. 32, no. 2, pp. 381-390, 2020.

[10] Y. Ortakci and C. Göloğlu, "Determination of the number of clusters using particle swarm optimization," SS, 2012.

[11] S. Akyol and B. Alataş, "Recent swarm intelligence optimization algorithms," Nevşehir University Journal of Science and Technology, vol. 1, no. 1, 2012.

[12] S. L. Gooskens et al., "Imatinib mesylate for children with dermatofibrosarcoma protuberans (DFSP)," Pediatric Blood & Cancer, vol. 55, no. 2, pp. 369-373, 2010.

[13] M. A. Çavuşlu, C. Karakuzu, and S. Şahin, "Hardware implementation of artificial neural network training with particle swarm optimization algorithm on FPGA," Polytechnic Journal, vol. 13, no. 2, pp. 83-92, 2010.