ASEC 2024 Conference

The 5th International Electronic Conference on Applied Sciences

04–06 December 2024 | Online

MDPI

PSO-Based Algorithm for Constrained Flow Shop Scheduling

Imane Laribi

Department of Industrial Engineering, University of Tlemcen

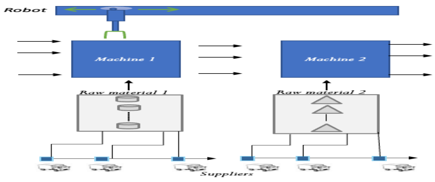
INTRODUCTION & AIM

Modern manufacturing systems face increasing complexity due to logistical constraints and resource limitations. Efficient scheduling is critical for optimizing production processes, especially in environments with multiple interdependent factors. This research investigates the Two-Machine Flow Shop Scheduling Problem (FSSP), a fundamental challenging optimization problem in manufacturing, extended by two significant constraints:

1. **Transport Robot Constraint**: Job movement between the two machines is managed by a single transport robot. The robot's transfer times directly influence the production schedule, making its integration into the scheduling framework essential.
2. **Raw Material Availability Constraint**: Raw materials are supplied intermittently by external suppliers at varying intervals, requiring careful synchronization between supply timing and production schedules to avoid delays caused by material shortages.

These constraints add a layer of complexity to the traditional FSSP, necessitating advanced optimization techniques.

Figure 1 below provides a schematic representation of the manufacturing configuration under study, illustrating the two machines, the transport robot, and the intermittent supply of raw materials.



The main objective of this research is to develop a metaheuristic-based scheduling approach to minimize the makespan. The algorithm integrates transport and raw material constraints within the optimization process, providing a robust solution to these logistical challenges.
This study aims to enhance production efficiency in flow shop systems, particularly in environments where logistical constraints critically impact overall manufacturing performance.

METHOD

The methodology employed to address these challenges is based on a **customized Particle Swarm Optimization (PSO)** algorithm. PSO is used to identify the optimal job sequence that minimizes the makespan, while implicitly accounting for the constraints related to the transport robot and raw material availability. The scheduling process does not involve separate optimization of the transport system or raw material management but integrates these factors into the optimization framework.

PSO is a metaheuristic that mimics the social behavior of particles in a swarm to search for optimal solutions. The algorithm is adapted to the discrete nature of the scheduling problem, where the goal is to determine the optimal job sequence that minimizes the makespan under the specified constraints. The customized PSO approach proceeds as follows:

1) **Initialization:** The swarm is initialized with a set of random job sequences, where each particle represents a potential solution to the scheduling problem .
2) **Velocity and Position Updates:** The velocity and position updates can be formally expressed as:

$$V_i^{t+1}=\text{Crossover}(Pbest_i^t, Gbest^t)$$
$$X_i^{t+1}=\text{Crossover}(X_i^t, V_i^{t+1})$$

- **Velocity Update (V_i^{t+1})**
  The velocity represents the movemen of a particle toward a new solution in the search space. It is influenced by two main components:
  - **Personal Best ($Pbest_i^t$):** The best solution found by particle up to the current iteration. This component encourages the particle to exploit its known optimal regions.
  - **Global Best ($Gbest^t$):** The best solution found by the entire swarm at iteration t. This acts as a collective guide, driving all particles toward the most promising region.

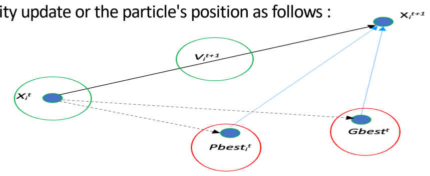
  The velocity update is performed using a crossover operator, adapted from Genetic Algorithms (GAs), which combines $Pbest_i^t$ and $Gbest^t$. This operator ensures that the particle's movement incorporates traits from both the personal and global optima, maintaining exploration while enhancing convergence.

- **Position Update (X_i^{t+1}):**
  After calculating the new velocity (V_i^{t+1}) the particle updates its position to generate a new solution in the discrete search space. This update uses the crossover operator to combine the particle's current position (X_i^t) and its updated velocity. This adaptation ensures that the particle transitions smoothly between solutions while maintaining diversity.

The advantage of this model is that it requires no parameter tuning, simplifying its application. However, it may become trapped in local optima if $Pbest_i^t$ = $Gbest^t$ or if $X_i^t = V_i^{t+1}$. To address this, a mutation operator, inspired by GAs is incorporated. This mutation introduces random perturbations to either the velocity update or the particle's position as follows :

V_i^{t+1} = *mutation* ($Gbest^t$)

X_i^{t+1}= *mutation* (X_i^t)



This improves the model's robustness by enhancing exploration diversity, mitigating the risk of stagnation, and increasing the likelihood of converging to a global optimum.

3) **Iterative Optimization:** The algorithm iteratively refines the job sequence by adjusting particle positions, continually optimizing the makespan **while respecting the transport and raw material constraints**.

RESULTS & DISCUSSION

The proposed PSO approach was evaluated using a series of benchmark problems, considering different scenarios involving raw material availability and transportation times between the two machines.

Problem instance	Makespan Reduction (PSO vs GA)	Relative Error (PSO vs ILP)
Small instances (up to 20 jobs)	6.73 %	3.23 %
Medium instances (20→60)	8.57 %	N/A
Large instanes (60 →120)	9.72 %	N/A

The performance of the Particle Swarm Optimization (PSO) algorithm was systematically evaluated and compared against the Genetic Algorithm (GA) and Integer Linear Programming (ILP) for different problem instances of the Two-Machine Flow Shop Scheduling Problem (FSSP).

For **small instances** (up to 20 jobs), the PSO algorithm achieved an **average reduction of 6.73%** in makespan relative to GA. Furthermore, the PSO solution exhibited a **3.23% relative error** when compared to the ILP results, indicating that PSO is capable of providing near-optimal solutions for small problem sizes.

For **medium instances** (20–60 jobs), PSO showed an **average reduction of 8.57%** in makespan relative to GA. However, due to the computational complexity associated with solving ILP for medium-sized problems, ILP was not applicable within a reasonable time frame. This limitation highlights the scalability challenges of ILP for problems of this size.

For **large instances** (60–120 jobs), PSO achieved an **average reduction of 9.72%** in makespan relative to GA. Similar to medium instances, ILP was not applicable due to its computational demands, reinforcing the advantages of PSO for larger problem sizes.

CONCLUSION

This research successfully develops a metaheuristic-based approach using Particle Swarm Optimization (PSO) to tackle the Two-Machine Flow Shop Scheduling Problem (FSSP) under the constraints of transport robot operations and raw material availability. The proposed methodology integrates these logistical challenges directly into the scheduling process, ensuring efficient resource utilization and minimizing makespan. The results show that the PSO approach performs effectively, achieving near-optimal makespans for smaller instances and improving performance for larger instances when compared to traditional methods.

FUTURE WORK / REFERENCES

Future work will focus on exploring multi-objective optimization techniques to further enhance scheduling efficiency by considering additional performance metrics.