Numerical Methods and Optimization

PROJECT 1

THE FLOW SHOP SCHEDULING PROBLEM

BY:

Belarbi Abdennour

Medjri Yasmina

Hamou Nadir

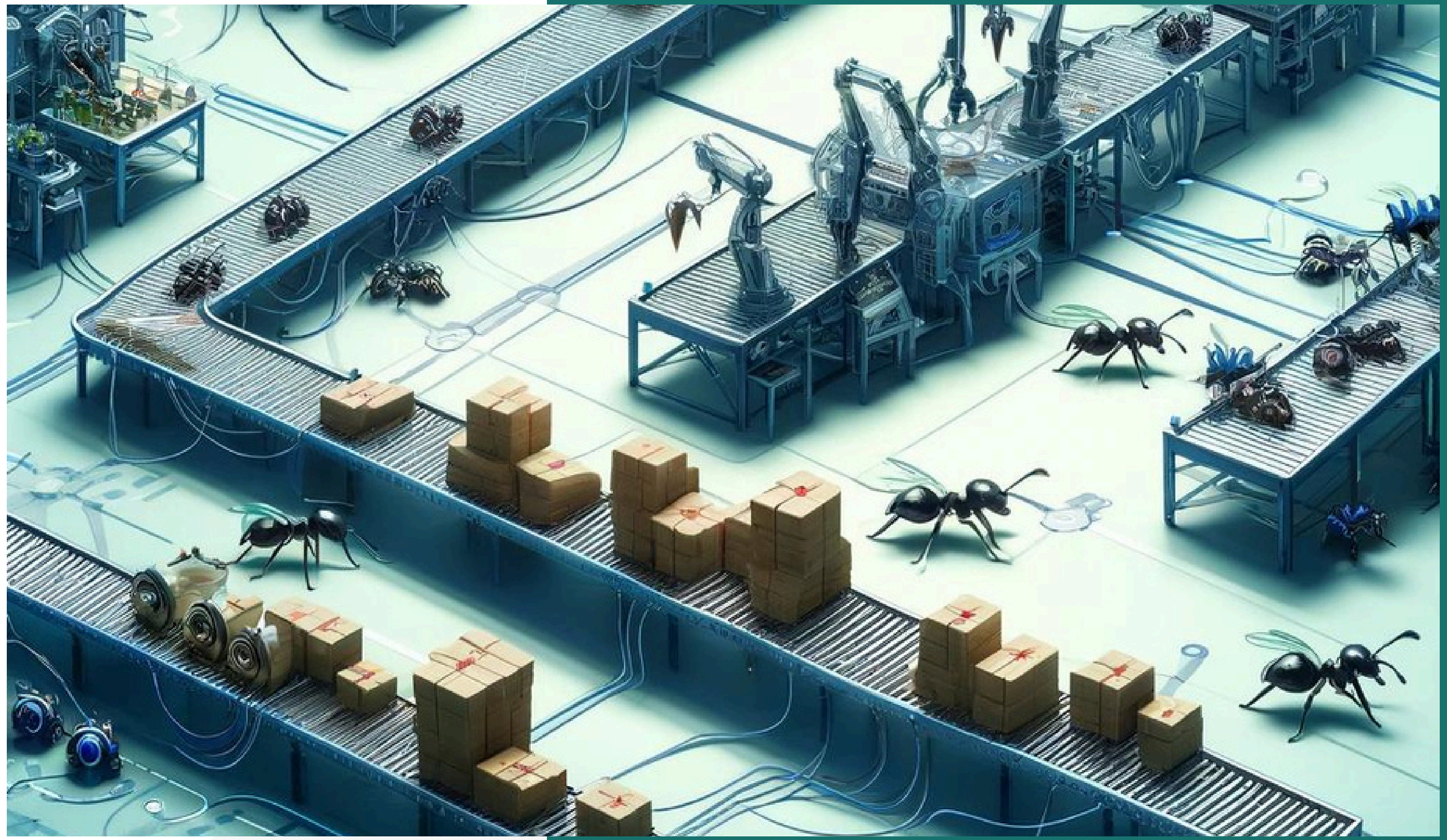
Boubekour Farida

Guendouz Nour El-Yakine

May 2024

TABLE OF CONTENT

SOLVING THE SHOP FLOW SCHEDULING PROBLEM USING ANT COLONY OPTIMIZATION ALGORITHM



MATHEMATICAL FORUMALTON

Formulation of The Problem :

Terminlogy :

-J = (1,...,N) set representing All jobs, j denotes job number.

-M = (1,...,M) set representing all machines, I denotes machine number.

-P is a matrix M*N representing Set of Processing time such that pij denotes processing time for j in machine I .

-ri denotes the minimum idle time for each machine
-di denotes the maximum idle time for each machine

Decision Variable :

S = matrix (M,J), such that sij denotes staritng time of job j on machine i.
C = matrix(M,J), such that cij denotes finishing time of job j on machine I
x_kj boolean varibale presents if j is the successor of job k.

Objective Function (Minimizing Makespan) :

Min z = max(Ci) sd (We minimize the maximum of all jobs time completions.)

Such That :
for a job k, he has one successor and one predecessor.
Σx_ik = 1 (job k has one predecessor only) for all I in {1,...,J}
Σx_ki = 1 (job k has one successor only) for all I in {1,...,J}

y >= Σ(cij- sij) = Σpij

Cij = Sij + pij

Constraint 1 :
Si,k ⩾ Ci,j (processing of operations can not be interrupted) (job k is successor of job j)
x_kj = 1

Constraint 2 : (idle times are respected)
di ⩾Si,k – Ci,j ⩾ ri , I in {1,2,..,N} and j in {2,..,M}.(job k is successor of job j)
x_kj = 1
Constraint 3 : (technological order of processes is respected)
Si,j ⩾ Ci-1,j ; I in {2,..,N} and j in {2,..,M}.(machines follow same order)

Complexity of the problem

NP-HARD!

The Flow Shop Scheduling Problem (FSSP) is known to be NP-hard. This can be explained by the following points:

1. **Permutation Flow Shop Scheduling Problem (PFSSP):**

- When the sequence of jobs is fixed across all machines (permutation assumption), the problem remains NP-hard. This is because even with a fixed order, the problem of determining the optimal sequence is combinatorial in nature and grows factorially with the number of jobs.
- For n jobs, the number of possible permutations is n!.

2. **Non-Permutation Flow Shop Scheduling Problem (NPFSSP):**

- In the non-permutation case, where each machine can process jobs in a different order, the problem is even more complex. The number of possible sequences grows exponentially.
- For each machine, there are n! possible sequences, leading to a total of (n!)^m possible schedules for m machines.

it is considered as an NP-HARD also due to the factorial growth of the problem size.

GANTT DIAGRAM

ANT COLONY OPTIMIZATION FOR FLOW SHOP SCHEDULING

MAIN FUNCTIONS USED IN THE CODE

01 Objective Function:

Purpose: Minimize makespan (completion time of the last job on the last machine).

02 Ant Colony Optimization (ACO):

Process:

- Iteration: Generate and evaluate job schedules.
- Pheromone Update: Adjust pheromone trails based on solution quality.
- Best Solution: Track and improve the best schedule.

03 Update Pheromones:

- Evaporation: Reduces pheromones to prevent stagnation.
- Deposit: Increases pheromones for better solutions to guide future ants.

04 Gantt Chart Visualization:

Purpose: Visualize job schedules and machine usage.

RESULTS ANALYSIS

Best solution found:

[[2 3 4 0 1]
[2 3 4 0 1]
[2 3 0 4 1]]

Makespan: 30.0

The best solution found by the Ant Colony Optimization (ACO) algorithm has yielded the following job sequences for each machine:

Machine 1 processes jobs in the order [2, 3, 4, 0, 1].

Machine 2 processes jobs in the same order [2, 3, 4, 0, 1].

Machine 3 processes jobs in the order [2, 3, 0, 4, 1].

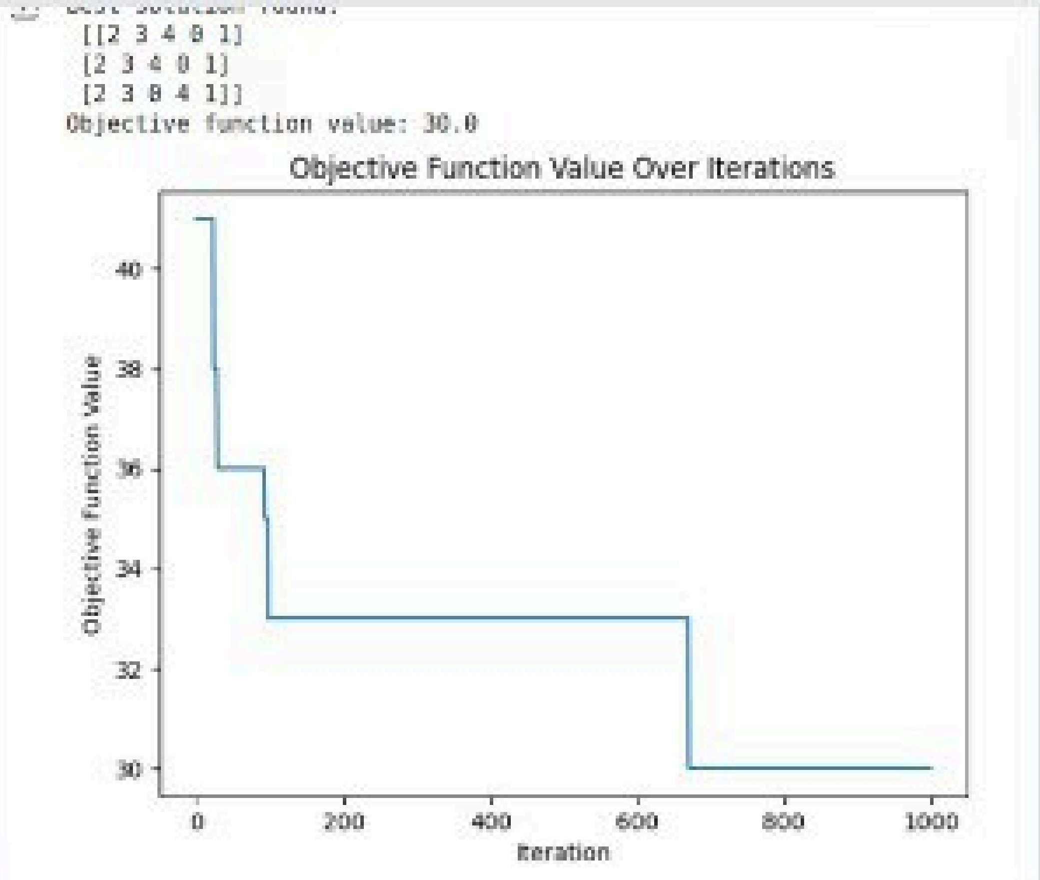
The makespan, or the total completion time for all jobs across all machines, is 30.0.

Variability of Results:

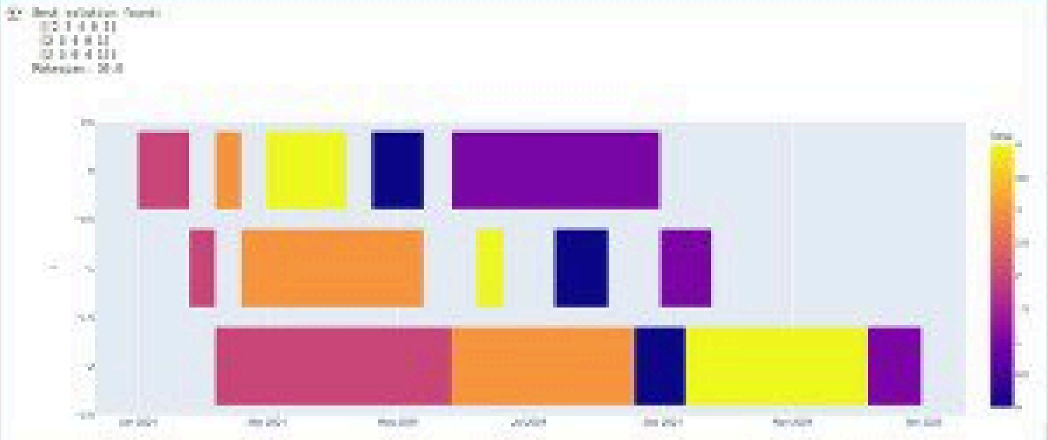
Each run of the ACO algorithm produces different results, including variations in the makespan. This variability is expected due to the stochastic nature of the ACO algorithm, which relies on random initial solutions and probabilistic decisions during the search process.

Recommendations for Consistency and Improvement:

- multiple runs
- parameter tuning
- visualization



[[2 3 4 0 1]
[2 3 4 0 1]
[2 3 0 4 1]]
Makespan: 30.0



THANK YOU

FOR YOUR NICE ATTENTION