Ant Colony Optimization for DSM's Flexible Job Shop Problem

TU Delft

Can ant colony optimization be used to efficiently solve the production scheduling problem of DSM?

1. Problem

*Scheduling **DSM**'s batch manufacturing process of enzymes*

- **E** different enzyme types
- **N** jobs: each for one enzyme type **e**
- **R** operations per job **n**
- **M** machines: divided over disjoint sets (each machine **m** is capable to do one operation type **r**)

Different product types consecutively on same machine? → cleaning time!

- **M** different **E** x **E** change-over matrices

Objective: minimize makespan (total schedule length)

2. Background

- Variant of **Flexible Job Shop Problem (FJSP)**
- Many real world applications
- NP-hard: much literature, lots of different approaches

- Exact approach: Mixed Integer Linear Programming (MILP)
  → Can guarantee optimal solutions, but high runtime often infeasible for real-world instances ☹
- Heuristics & **metaheuristics**
  → Good solutions, practical runtime!

- **Ant Colony Optimization (ACO)**
  → Population-based, probabilistic algorithm
  → Inspired by foraging behaviour of ants
  → Precedent with sequence-dependent setup times

3. Application of Ant Colony Optimization

3.1 Weighted Disjunctive Graph Representation

- Nodes O_{nr}
  - → job n
  - → operation r
- Undirected edges: operation order per machine
- Directed edges: operation order per job
- Weights: processing, cleaning, … times

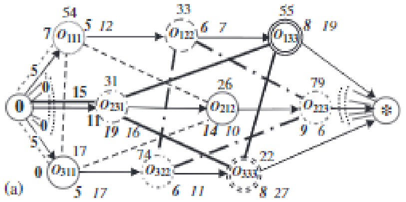

Figure 1: Example of a WDG for the FJSP from [1].

3.2 Algorithm

For each epoch:

For each ant:

For step 1 ... total number of operations:

1. Determine candidate nodes: first unscheduled operation of each job n O_{nr}

2. Determine feasible moves: $O_{n'r}$ --m-> O_{nr}
   - $O_{n'r}$: currently last operation of a machine m for operation type r

3. Select move using transition probability rule, depending on:
   - Pheromone amount
   - Heuristic function

4. Locally update pheromone amount, direct selected undirected edge and remove the now infeasible undirected edges

5. Calculate operation start and completion time and propagate to weights

If applicable, update best schedule found this epoch

Globally update pheromones of all moves in best schedule found this epoch

If applicable, update best schedule found

Output best schedule found

4. Experimental results

Instance	τ_0				
	0.01	0.05	0.1	0.5	1
2	51.3 += 2.1 1.5 += 0.4	50.0 += 2.0 2.9 += 1.2	50.0 += 1.7 2.6 += 0.9	50.3 += 0.6 2.2 += 0.3	49.7 += 0.6 2.0 += 0.5
7	122.0 += 2.0 33.0 += 9.6	122.3 += 4.0 25.0 += 3.3	121.3 += 2.5 23.7 += 3.0	122.7 += 4.2 29.0 += 3.4	122.7 += 1.2 51.1 += 27.0
12	196.3 += 6.4 181.7 += 140.6	194.0 += 2.0 145.2 += 28.0	193.7 += 6.4 129.0 += 36.6	196.3 += 4.0 156.7 += 45.1	194.0 += 5.6 145.7 += 13.4

Instance	q_0				
	0.1	0.3	0.5	0.7	0.9
2	58.7 += 1.5 2.1 += 1.0	51.3 += 1.5 2.0 += 0.7	47.7 += 1.2 1.6 += 0.3	44.3 += 0.6 2.1 += 0.2	44.7 += 1.2 1.3 += 0.2
7	143.0 += 2.6 28.7 += 0.4	121.3 += 4.0 35.2 += 5.8	111.3 += 2.1 25.5 += 5.0	104.7 += 2.5 22.5 += 3.0	100.7 += 1.5 26.8 += 5.3
12	234.3 += 1.2 119.8 += 24.0	198.0 += 4.6 113.5 += 39.4	175.7 += 2.3 135.5 += 69.1	165.7 += 2.3 95.5 += 11.7	159.3 += 0.6 113.9 += 31.1

Table 1, 2: Experimental evaluation of two hyperparemeters, τ_0 and q_0, shown for three different instances. The top entry in each cell is the average makespan += standard deviation, with below it the same for the runtime in seconds. Results were calculated over three independent runs, each having three epochs without improvement as stopping condition.

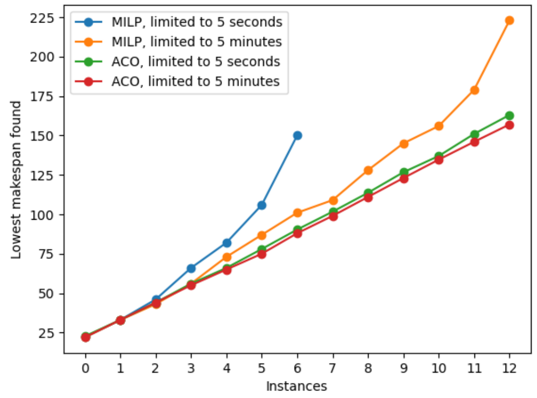

Figure 2: Comparison of the performance of the MILP solver and ACO algorithm (averages over three runs) for two different time limits, using τ_0 = 0.1 and q_0 = 0.9.

5. Conclusions

- T_0: not much impact on performance
- q_0: higher values → lower makespans and runtimes

- For all tested instances, ACO significantly outperforms MILP when optimizing for makespan
- Based on this, ACO is an efficient method to solve the production scheduling problem of DSM

However:

- Larger number of instance necessary to draw reliable general conclusions about method
- All results for makespan -> in practice multiple conflicting objectives

References

[1] Andrea Rossi and Gino Dini. "Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method". In: Robotics and Computer-Integrated Manufacturing 23.5 (Oct. 2007), pp. 503–516. ISSN: 07365845. DOI: 10.1016/j.rcim.2006.06.004.

Tim Numan
t.numan@student.tudelft.nl

Supervisors: Kim van den Houten and Dr. Mathijs de Weerdt
EEMCS - CSE3000 Research Project