Ant Colony Optimization Based Algorithm For Solving Scheduling Problems with Setup Times on Parallel Machines

Zsolt Mihály, Zsombor Sentes, Zoltán Lelkes

Abstract: In this paper, a production scheduling problem with sequence-dependent setup times on a set of unrelated parallel machines is addressed. The objective function is to minimize the total setup time . An algorithm based on ant colony optimization combined with a heuristic is proposed for solving large problems efficiently. It is shown that even a simpler version of the problem can not be tackled with MILP. ACO gives good results for the simpler problem version in a reasonable time. Even ACO can not give good results for the industrial problem. However, ACO combined with the heuristic can give us satisfactory results for the industrial problem in a reasonable time.

Keywords: production scheduling, ant colony optimization, heuristic, setup time, parallel machines, industry

Introduction

Production scheduling is a key success factor in all manufacturing industries. The choice of the schedule has a powerful impact on the cost of production [7]. Advanced optimization algorithms are needed in order to solve production scheduling problems (PSPs) efficiently.

Total setup time (TST) has a crucial effect on the profitability of equipement. In the latest review of scheduling problems with setup times, there are only two articles addressing TST when parallel machines are considered [1]. Cevikcan et al. developed an expert system [4]. Dinh et al. tackled the problem using MILP [6].

ACO is a widely used technique for solving PSPs with sequence dependent setup times. E. g. Arnaout et al. used ACO in order to minimize the makespan on unrelated parallel machines with sequence-dependent setup times [3], [2].

In this paper, a PSP with sequence-dependent setup times on a set of unrelated parallel machines is addressed. The objective function is to minimize total setup time. An algorithm based on the combination of ant colony optimization and a heuristic is proposed for solving large problems efficiently. To our best knowledge, this is the first paper which addresses the described problem using ACO.

The rest of the paper is organized as follows. The second part describes the examined PSP. The third part outlines the proposed algorithm. The fourth part shows the computational results. Finally in the fifth part, a concise conclusion about the research is presented.

Problem Statement

In this PSP, a set of n independent jobs should be scheduled on m unrelated parallel machines. Only one job can be executed on a machine at the same time. Each job j has a processing time on each machine m $p_{j,m}$. Sequence and machine dependent setup times G_{m,j_1,j_2} are also considered in this problem. G_{m,j_1,j_2} is the setup time when machine m switches the production from job j_1 to job j_2. The objective is to minimize total setup time. The examined PSP can be described as a mixed integer linear programming (MILP) model. A simplified version of the problem is already NP-hard [9]. It follows that the examined problem is NP-hard, too.

Proposed algorithm

The proposed algorithm combines ACO and a heuristic. The algorithm starts from an initial schedule. After the initialization of the algorithm, a solution is constructed for each ant. The best solutions are improved using the heuristic. The heuristic tries to find groups for ungrouped jobs. Groups consist of adjacent jobs without setup times. The heuristic checks for each ungrouped task if it can be placed into a group without impairing the objective. In the next step, the pheromones are updated. The convergence factor is calculated based on the new pheromone values. If the stop criterion is met, the algorithm stops.

The stop criterion is that the convergence factor is higher than a certain value. If the stop criterion is not met, the algorithm continues with constructing new solutions. A simplified flow chart of the algorithm can be seen on Figure 1.

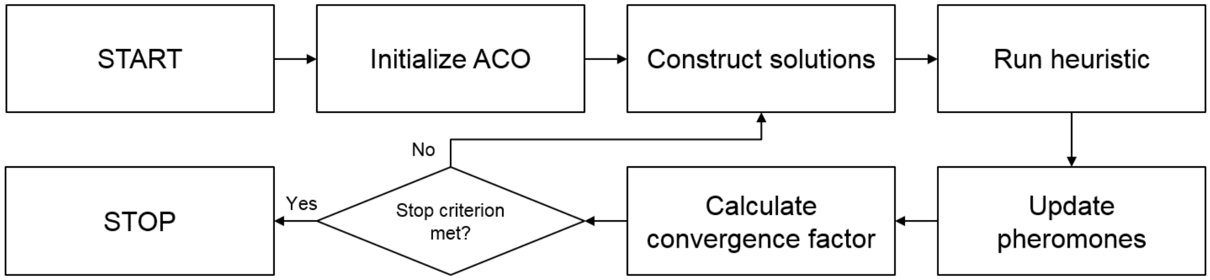

Figure 1: The outline of the proposed algorithm

The algorithm has been implemented in AIMMS modelling language [8]. It has already been used in other studies with success. E. g. Czuczai et al. used it for solving complex scheduling problems with rolling operation algorithm [5].

Computational results

Comparison with MILP

ACO has been compared with MILP on a simpler version of the investigated problem. There is only one machine to schedule. The objective is to minimize makespan. Both algorithms have been run on seven scheduling problems. The problems are different in the number of jobs: 10, 15, 20, 25, 50, 100, 150. When MILP could find optimal solution then ACO could find the same optimal solution as well. When the number of jobs is 25, MILP runs almost 40 times longer. When the number of jobs is at least 50, MILP can not find the optimal solution in a reasonable time. Even smaller instances of the simplified industrial problem can not be solved by MILP. On the other hand, ACO can give good results in a reasonable time. The data regarding the optimal solutions can be seen on Table 1. The runtime data regarding the investigation can be seen on Figure 2.

Table 1: The optimal makespans found by MILP and ACO

Number of jobs	10	15	20	25	50	100	150
MILP	787	1071	1355	1696	No data	No data	No data
ACO	787	1071	1355	1696	3290	6529	9881

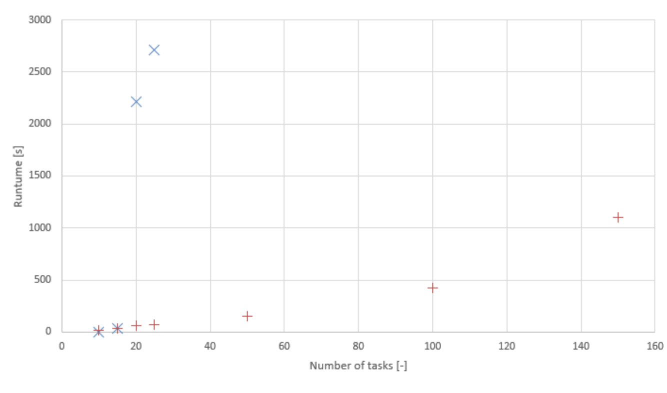

Figure 2: Comparison of MILP and ACO runtimes on single machine

Examining the proposed algorithm on the stated problem

Even ACO does not give satisfactory results when the industrial problem needs to be solved. It can happen that the new schedule is not better then the initial. For this reason, ACO has been combined with a heuristic. The ACO combined with the heuristic is able to give satisfactory results. In this part, a large problem is investigated. 193 jobs need to be scheduled on 10 machines.

The algorithm starts from an initial schedule. The initial schedule has been created by a real-life factory using a greedy randomized adaptive search procedure. The optimized schedule is shorter by 311 minutes, that is by 2.2% than the initial schedule. The runtime was 420 seconds. ACO solution construction and the heuristic consumed the most time. 212 seconds were spent on ACO solution construction, and 91 seconds on the heuristic. The initial and the optimal schedule can be seen on Figure 3 and 4.

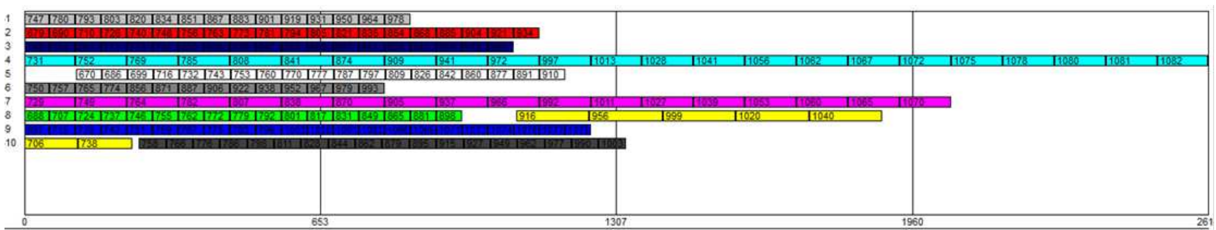

Figure 3: Initial schedule

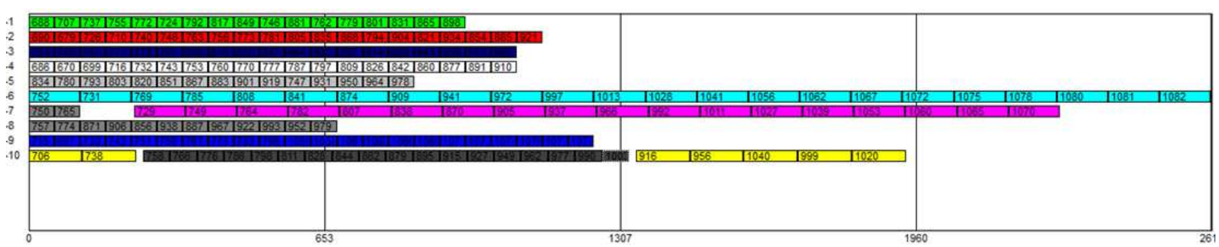

Figure 4: Optimized schedule

Conclusions

In this paper, a production scheduling problem with sequence-dependent setup times on a set of unrelated parallel machines is addressed. The objective function is to minimize total setup time. An algorithm based on the combination of ant colony optimization and a heuristic is proposed for solving large problems efficiently. It is shown that even a simpler version of the problem can not be tackled with MILP. ACO with the heuristic can give us satisfactory results in a reasonable time. A large problem has been solved with the proposed algorithm. 193 jobs needed to be scheduled on 10 machines. The optimized schedule is shorter by 311 minutes, that is by 2.2% than the initial schedule.

Acknowledgements

This research is supported by EFOP-3.6.1-16-2016-00006 "The development and enhancement of the research potential at John von Neumann University" project. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

References

[1] Allahverdi, A. The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 246:345–378, 2015.

[2] Arnaout, J.-P., Musa, R., and Rabadi, G. A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines-part ii: enhancements and experimentations. *Journal of Intelligent Manufacturing*, 25:43–53, 2014.

[3] Arnaout, J.-P., Rabadi, G., and Musa, R. A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, 21:693–701, 2010.

[4] Cevikcan, E., Durmusoglu, M. B., and Baskak, M. Integrating parts design characteristics and scheduling on parallel machines. *Expert Systems with Applications*, 38:13232–13253, 2011.

[5] Czuczai, B., Farkas, T., Rev, E., and Lelkes, Z. Rolling operation algorithm for solving complex scheduling problems. *Industrial and Engineering Chemistry Research*, 48:3898–3908, 2009.

[6] Dinh, T.-C. and Bae, H. Parallel servers scheduling with dynamic sequence-dependent setup time. *Smart Innovation, Systems and Technologies*, 16:79–87, 2012.

[7] Pinedo, M.L. *Scheduling*. Springer-Verlag, Berlin, 2016.

[8] Roelofs, M. and Bisschop, J. *AIMMS The user's guide*. AIMMS B.V., AP Haarlem, The Netherlands, 2016.

[9] Webster, S.T. The complexity of scheduling job families about a common due date. *Operations Research Letters*, 20:65–74, 1997.