A realistic hybrid flow shop scheduling problem with availability restrictions, priorities, and machine qualifications

Christin Schumacher[1] and Dominik Mäckel[2]

[1] TU Dortmund University, Department of Computer Science - Modeling and Simulation, Germany christin.schumacher@tu-dortmund.de
[2] TU Dortmund University, Department of Computer Science - Modeling and Simulation, Germany dominik.maeckel@tu-dortmund.de

Keywords: heuristics, machine availability, priority, machine qualification, hybrid flow shop, makespan.

1 Introduction

While real-world problems in practice contain numerous restrictions, problems in the literature mostly consider a few specifications at a time [1]. Our problem of an automotive supplier includes two production stages, where both stages contain unrelated machines ($FH2,\ ((RM^{(k)})_{k=1}^{2})$). Machine qualifications have to be considered (M_j), jobs might skip a stage ($skip$), priority groups of jobs need to be respected ($prec$), machines might not be available at beginning the production (rm), and machines have times of unavailability ($unavial$) [2, 3]. The objective is to minimize makespan C_{max}. According to Graham et al. [4], our problem can be categorized as follows:

$$FH2,\ ((RM^{(k)})_{k=1}^{2}) \mid M_j, prec, skip, rm, unavail \mid C_{max} \tag{1}$$

In the following, heuristics and metaheuristics are compared to determine the best solution method, concerning short computation times. Furthermore, a mixed-integer problem (MIP) is used for comparison. All developed methods are tested on historical data of the use case so that the most promising algorithms for the company can be found.

2 Related Work

Even for the problem of two identical parallel machines on one stage, and one machine on the other stage, Gupta [5, pp. 359–360] has proved *NP-hardness*. So hybrid flow shop problems are considered as *NP-hard*. Due to high computation times for *NP-hard* problems, MIP models, even for small instances of jobs, are often not applicable in practice. However, they are useful for benchmarks.

To the best of our knowledge, no study has developed a heuristic algorithm or a MIP for our problem (1). In their literature overviews Ribas, Leisten, and Framiñan [6], Komaki, Sheikh, and Malakooti [7], and Ruiz and Vázquez-Rodríguez [2] mention several hybrid flow shop problems. Listed problems of papers conform at most with one or two restrictions to M_j, *prec*, *skip*, *rm* or *unavail*. The problem which is most familiar to our application case is provided by Ruiz, Şerifoğlub, and Urlings [1] and is at the same time one of few papers dealing with six realistic restrictions in one hybrid flow shop environment. But, since our application case differs in several restrictions from the mentioned papers, algorithms in literature cannot be applied for (1).

3 Heuristic and metaheuristic methods

For the presented scheduling problem, eight priority rules, one genetic algorithm (GA), twelve local search strategies, and four simulated annealing variants are developed. Applied priority rules are extensions of Shortest Processing Time (SPT) [3], Longest Processing Time (LPT) [3], Johnson's algorithm and NEH [1]. For second stage scheduling, priority rules are combined with Earliest Completion Time (ECT) or job-based relation (JBR). ECT selects the job with the shortest completion time from all jobs to be scheduled and JBR preserves the permutation of jobs from the first stage. All developed priority rules use ECT for machine assignment at the stages, where the algorithm successively schedules the next element of an ordered list of jobs to the machine that can complete the job first and is available at the dedicated time and eligible for the job. Each priority group gets scheduled separately beginning with the highest priority.
Local search strategies have been tested in six variants:

- The neighborhood can be explored by generating a defined amount of moves from the current schedule and selecting the solution with best objective value, in case of improved makespan (Steepest Descent) or by selecting a random neighbor and accepting the solution, in case of improved makespan (Random Descent).
- The neighborhood can be build by "shift" and "swap" moves. "Shift" moves a random job to a random position in the schedule. "Swap" selects two random jobs and exchanges their positions. Both operations preserve machine qualifications and priority groups.
- Steepest descent can be executed using a tabu list that contains a finite set of prohibited moves to solutions that have already been visited. Tabu search also accepts solutions with equal C_{max}.
- Again, for all variants, ECT or JBR can be chosen for scheduling the second stage (see priority rules).

Simulated Annealing is expanded from the algorithm of Kirkpatrick, Gelatt, and Vecchi [8]. It enables escaping from local minima by accepting solutions with larger C_{max} by a given probability defined by the current temperature that is decreasing over the optimization process. It is also combined with neighborhood strategies "shift" or "swap". ECT or JBR can be applied to schedule the second stage.

In addition, genetic algorithm (GA) by Wu, Liu, and Wu [9] is modified. If a crossover or mutation operation results in a non-feasible chromosome, the chromosome is reset after the particular operation and the further algorithm again uses the original instance. To ensure priority orders, priorities are ensured in each operation.

4 Results

Developed scheduling heuristics are applied to the historical production data of the company. The first step to evaluate the algorithms is to determine the benchmark using a developed MIP by the authors. For our MIP model, we have extended the model by Ruiz, Şerifoğlub, and Urlings [1] to the constraints *unavail* and *prec*. Even after 3 days of computation time, the optimality of some solutions for weeks with high production volume cannot be proven via the MIP. On average of evaluated weeks, the gap is about 13 %. As well, a few weeks are tested with a maximum computation time of 14 days. Despite the high additional computation time, no significant improvements are generated.

Also, results of heuristics and metaheuristics did not provide better C_{max}-values. Thus, it can be concluded that the benchmarks of MIP are almost optimal. In the following, heuristics and metaheuristics are compared to benchmarks in order to measure their performance as follows: $\Delta C_{max} = \frac{heuristic\ solution\ -\ MIP\ solution}{MIP\ solution}$.

In contrast to MIP, all heuristic algorithms need less than one minute to provide solutions. Using ΔC_{max}, priority rules are compared over 20 selected calendar weeks (Figure 1). Besides, Table 1 calculates averages for all weeks. The evaluation shows, that SPT has the worst performance with both second stage strategies and NEH is the best priority rule with both second stage strategies. In the midfield, LPT dominates Johnson's algorithm.

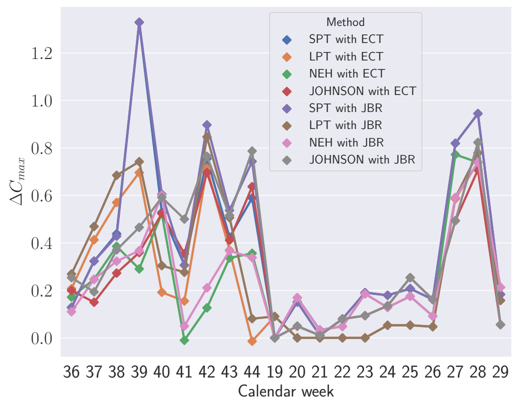

Fig. 1: ΔC_{max} of heuristics over the selected evaluation period.

Method	avg. ΔC_{max}
NEH with ECT	0.249141
NEH with JBR	0.249370
LPT with ECT	0.251440
JOHNSON with ECT	0.282263
LPT with JBR	0.297500
JOHNSON with JBR	0.329924
SPT with ECT	0.387423
SPT with JBR	0.411233

Table 1: Comparison of ΔC_{max} for selected heuristics using ECT as second stage strategy.

In the field of metaheuristics, GA with 90 000 objective function evaluations dominates the other metaheuristics (see figure 2 and table 2). Since the local search strategies evaluate 5 000 iterations, the parameters of the genetic algorithm have been reduced in "GA5000" to match with local search strategies regarding the number of function evaluations. Tabu search performs slightly better than local search strategies without tabu list in both cases. When "shift" is applied, it dominates "swap" in all cases. Also, simulated annealing is dominated by the other metaheuristics. Therefore, simulated annealing and "swap" are not considered in the tables and figures of this paper. All local search strategies perform significantly best by applying NEH as the initial solution. This shows a significant influence of the initial solution on the solution quality.

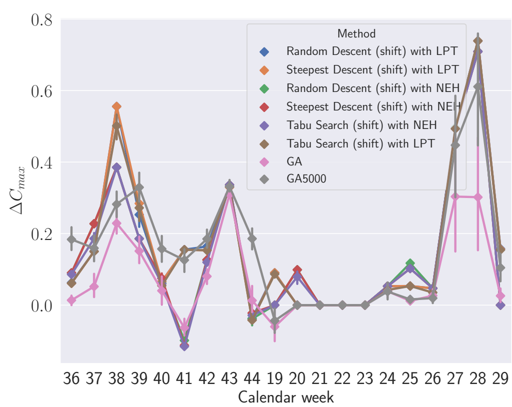

Fig. 2: ΔC_{max} of metaheuristics over the selected evaluation period.

Method	avg. ΔC_{max}
GA	0.073729
Tabu Search with NEH	0.134955
Steepest Descent with NEH	0.139813
Random Descent with NEH	0.139913
GA5000	0.156668
Tabu Search with LPT	0.162064
Random Descent with LPT	0.166863
Steepest Descent with LPT	0.167259

Table 2: Comparison of ΔC_{max} for selected metaheuristics using ECT as second stage strategy and shift move for local searches.

5 Conclusion

For our problem (1) a MIP, several heuristics, and metaheuristics have been developed and tested in manufacturing production. Regarding priority rules, NEH performs best. Comparing metaheuristics, a genetic algorithm with 90 000 function evaluations provides the best results followed by tabu search which only needs 5 000 iterations and lower computation time. For real-world applications, in future work, algorithms have to be modularized, so that it is possible to synthesize them automatically and reduce manual modeling effort.

References

[1] Rubén Ruiz, Funda Sivrikaya Şerifoğlub, and Thijs Urlings. “Modeling realistic hybrid flexible flowshop scheduling problems”. In: *Computers and Operations Research* 35.4 (2008), pp. 1151–1175.

[2] Rubén Ruiz and José Antonio Vázquez-Rodríguez. “The hybrid flow shop scheduling problem”. In: *European Journal of Operational Research* 205.1 (2010), pp. 1–18.

[3] Michael L. Pinedo. *Scheduling. Theory, Algorithms, and Systems.* 5th ed. Cham: Springer International Publishing, 2016.

[4] R.L. Graham et al. “Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey”. In: *Annals of Discrete Mathematics* 5 (1979), pp. 287–326.

[5] Jatinder N. D. Gupta. “Two-Stage, Hybrid Flowshop Scheduling Problem”. In: *The Journal of the Operational Research Society* 39.4 (1988), p. 359.

[6] Imma Ribas, Rainer Leisten, and Jose M. Framiñan. “Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective”. In: *Computers & Operations Research* 37.8 (2010), pp. 1439–1454.

[7] G. M. Komaki, Shaya Sheikh, and Behnam Malakooti. “Flow shop scheduling problems with assembly operations: a review and new trends”. In: *International Journal of Production Research* 57.10 (2019), pp. 2926–2955.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by simulated annealing”. In: *Science* 220.4598 (1983), pp. 671–680.

[9] Yi Wu, Min Liu, and Cheng Wu. “A genetic algorithm for solving flow shop scheduling problems with parallel machine and special procedure constraints”. In: *Proceedings of the 2003 Proceedings of the Second International Conference on Machine Learning and Cybernetics*. New York, N.Y and Piscataway, N.J: IEEE, 2003, pp. 1774–1779.