Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

LOCAL SEARCH AND TABU SEARCH ALGORITHMS FOR MACHINE SCHEDULING OF A HYBRID FLOW SHOP UNDER UNCERTAINTY

Christin Schumacher
Peter Buchholz
Kevin Fiedler
Nico Gorecki

Informatik 4 – Modeling and Simulation
TU Dortmund University
Department of Computer Science
D-44221 Dortmund, GERMANY

ABSTRACT

In production systems, scheduling problems need to be solved under complex environmental conditions. In this paper, we present a comprehensive scheduling approach that is applicable in real industrial environments. To cope with the parameter uncertainty of real world problems, forecasting, classification, and simulation techniques are combined with heuristic optimization algorithms. Thus, the approach allows for identifying and including demand fluctuations and scrap rates, and offers a selection of suitable schedules depending on particular demand constellations in scheduling. Furthermore, we adapt seven optimization algorithms for two-stage hybrid flow shops with unrelated machines, machine qualifications, and skipping stages with the objective to minimize the makespan. The combination of methods is validated on a real production case of the automobile industry. The paper shows for the application case that metaheuristics provide significantly better results than SPT and safety factors, above a certain size, can reduce their effect preventing incomplete demand positions.

1 INTRODUCTION

In complex production environments, as they can be found in the automotive industry, machine schedules have to be determined under frequently changing customer demands and possible production failures or unplanned delays. The determination of almost optimal schedules in such an environment is a challenge because of the high complexity of the scheduling problem and the uncertainty of several system parameters. Even under simplifying conditions neglecting uncertainty and set-up times, the resulting deterministic optimization problems are NP-hard and can be solved exactly only for small and unrealistic configurations within a reasonable time. Heuristics and metaheuristics are more flexible and allow one to approximate the optimal schedule for the deterministic model for much larger configurations. A multitude of scheduling approaches deals with developing improved scheduling algorithms for deterministic test instances with only few realistic and mostly simplified conditions in comparison to practice. These studies are appropriate to show the basic functionality of the algorithms, but it is a long way to apply them under real production conditions (Ruiz et al. 2008). With the approach presented in this paper, we generate schedules based on realistic problem settings. For realistic models, data from the running production have to be used to define the actual instance of the scheduling problem. A statistical evaluation of past data allows one to determine safety margins to compensate scrap or unplanned demands. Before the schedule resulting from a deterministic optimization problem should be exploited in a production with its statistical fluctuation in the parameters, it should be tested and possibly improved in a stochastic simulation model.

Thus, we present a comprehensive methodology to solve scheduling problems within the scope of uncertainty of production parameters arising from a real application case of a supplier in the automotive industry. The basic problem describes a two-stage scheduling problem with unrelated machines, machine qualifications, and skipping stages. Seven heuristic algorithms, namely Shortest Processing Time heuristic (SPT), local search, and tabu search metaheuristics algorithms, are implemented to minimize the makespan and parameters are derived from available real-world production data. A composition of forecasting, classification, simulation, and heuristic algorithms is developed to identify high-risk articles with demand fluctuations, include and evaluate scrap rates, and select appropriate safety factors to be multiplied by the demands. A detailed simulation model from the production has been built using the software AnyLogic 8.5 (Borshchev and Grigoryev 2014). By means of simulation it is possible to include the interdependence of production parameters, to evaluate and possibly improve schedules before they are applied in the real production environment, and select the most suitable schedules depending on the particular demand constellation. In addition, the approach shows which production details should be included in mathematical algorithms and which should be included in a simulation model.

The structure of the paper is as follows: In the next section we present a detailed description of the scheduling problem. Related work is briefly summarized afterwards. Section 4 introduces seven heuristic algorithms to compute schedules for the hybrid flow shop problem. Afterwards, in Section 5, the available data from the production are described and methods are presented to estimate the parameters of the optimization problem from the data. Then, a detailed analysis of the schedules in the simulation model is introduced. In Section 7, some typical results of example runs are presented, followed by the conclusions which outline some steps for future research.

2 DESCRIPTION OF THE BASIC MODEL

The problem analyzed is a two-stage hybrid flow shop where a set of jobs $N = \{1,\ldots,n\}$ has to be processed in stages i, where $i \in M$ with $M = \{1,2\}$ and each job $j \in N$ has the same route to follow through the stages. At each stage i a job is to be processed by at most one machine l, $l \in M_i = \{1,\ldots,m_i\}$, where $m_i \geq 1$. As far as possible, we set the parameters according Ruiz et al. (2008) to support a homogeneous notation for machine scheduling studies. The following assumptions hold for our model.

1. In the first stage, all jobs are available for processing at time 0.
2. Jobs that are to be produced in the first and second stage can only begin to be processed in the second stage, after the whole job corresponding with $demand_{1j}$ has been finished in the first stage.
3. Each machine can only process one job at a time.
4. The processing of a job on a machine cannot be interrupted, i.e., preemption is not allowed.
5. The problem contains machine qualifications. Each job can only be processed on certain machines in each stage, where the set of eligible machines is E_{ij}, with $1 \leq |E_{ij}| \leq m_i$.
6. Some jobs visit only one of the stages. The set of stages to be visited is F_j, where $1 \leq |F_j| \leq 2$.
7. There are infinite buffers in front, between, and at the end of the two stages.
8. Processing times p_{ilj} are described by independent random variables separated by machine, stage and job. In the second stage, processing times are not machine-related.
9. For set-up times s_i, stochastic distributions over all machines and articles per stage are available.
10. For scrap rates r_{ij}, article- and stage-related stochastic distributions are available for both stages.
11. Available parts on stock $stock_{ij}$ are included.
12. Order quantities of the jobs $demand_{ij}$ underly uncertainty and may change even during the week when it is produced, which results in varying appropriate production volumes $productionvolume_{ij}$.
13. The objective is to minimize the makespan C_{max}, where C_{max} is the maximum completion time and C_{ij} defines the completion time of a job j on stage i, which can be determined with parameters $productionvolume_{ij}$ and p_{ij}. $C_{max} = \max\limits_{i\in M, j\in N} C_{ij}$.

With defining the precedence relations x_{iljk} and x as variables, schedules can be described in their full extent, where $x = (x_{iljk})_{i \in M, l \in M_i, j,k \in N}$. x_{iljk} will be used in the optimization algorithms and is defined as

$$x_{iljk} \quad := \quad \begin{cases} 1, & \text{if job } j \text{ precedes job k on machine } l \text{ at stage } i, \\ 0, & \text{otherwise.} \end{cases}$$

3 RELATED WORK

An enormous number of papers on scheduling for hybrid flow shops is available. Overviews can be found in Ribas et al. (2010), Ruiz and Vázquez-Rodríguez (2010) and Komaki et al. (2019). In the following, selectively a few studies of the overviews are discussed that dealt with hybrid flow shops to minimize the makespan and are related to our problem (see Section 2).

Jabbarizadeh et al. (2009) test three constructive heuristics for a problem with identical parallel machines, machine qualifications, sequence dependent setup times, and limited availability of machines. In contrast to our problem, skipping stages and unrelated machines are not considered. They assign the jobs in the processing stages $i > 1$ in order of completion times of the previous stage, namely according to Earliest Completion Times (ECT). Their evaluation shows that in combination with Shortest Processing Time (SPT), results are better than with Longest Processing Time (LPT). A heuristic based on the algorithm of Johnson (1954) gives the best results in their study. In the field of metaheuristics, a version of simulated annealing, which is a local search algorithm with acceptance of setbacks, outperforms a genetic algorithm. Ruiz et al. (2008) present a study dealing with a large number of realistic components and unrelated machines. But, they only compare constructive heuristics and find that NEH provides best solutions. Low et al. (2008) compare sixteen combinations of heuristics for unrelated machines with machine qualifications in the first stage and only one machine in the second stage. They do not consider skipping stages. They find that a modified Johnson rule with planning the second stage according to ECT of the first stage performs best. For a hybrid flow shop with identical parallel machines, skipping stages, and sequence-dependent setup times, Naderi et al. (2010) apply job sequencing and machine assignment in the same step. They do not take machine qualifications and unrelated machines into account. All stages i are scheduled according ECT of the stage i itself and taking into account the arrival times at the stages, if $i > 1$, which result from C_{i-1j}. Compared to other constructive algorithms for their problem and with their test data, this constructive algorithm provides good results. They also use Iterated Local Search, which dominates the other tested metaheuristics like genetic algorithms up to a number of 80 jobs. Dios et al. (2018) compare 24 constructive heuristics for a hybrid flow shop with identical parallel machines and skipping stages and the evaluation of their experiments show that two SPT-based and one LPT-based heuristic generate the best schedules according to C_{max}. They do not take unrelated machines into account.

Thus, none of the studies provides local search and tabu search algorithms for the problem mentioned in Section 2. In addition, the mentioned approaches compute schedules for deterministic models, but most production processes show stochastic behavior. One way to optimize stochastic models is the use of sample average approximations, which is used by Almeder and Hartl (2013) in combination with variable neighbourhood search to optimize a two-stage flow shop problem describing a real world production process in the metal-working industry. In contrast to our problem, the number of machines is smaller, machine qualification and setup times are not required, and the behavior of orders seems to be more homogeneous. Production data are not used to determine model parameters. Instead, sampling averaging of simulation results is used to determine the parameters of the optimization problem, which is analysed with similar methods that we apply, but tabu lists seem to be not used.

Our approach is also related to approaches that combine simulation and optimization. Overviews of methods can be found in Juan et al. (2015) and Figueira and Almada-Lobo (2014). Simulation optimization is also often applied in semiconductor manufacturing as described in Lin and Chen (2015). With regard to the different possibilities to combine optimization and simulation, our study falls into the category of first computing a schedule for a deterministic model, which is afterwards evaluated in the detailed simulation model. In an additional step it is then possible to improve the schedule based on the simulation results using techniques for simulation optimization.

In application scenarios, parameters for the optimization problem have to be estimated and future demands have to be forecasted. For parameter estimation and modeling from available data, standard methods of input modeling as summarized for example in Law (2015) may be applied. Classification of articles according to their future demands is more demanding. To the best of our knowledge, only Murray et al. (2015) have examined an application case of article classification. They focus on a higher stage in production planning than in our application, where we plan the machine scheduling for in-house production, which is one of the last stages in the supply chain. They use k-means clustering in order to group customers into segments and do not address the single article demands. Customers are the first layer in a supply chain, because they are causing the demand for articles, but the demand becomes more distorted and volatile when customers' order quantities are planned through the different stages of the supply chain (Lee et al. 1997). To forecast demands, various methods from statistics like regression or time-series exist. It turns out that no single best method with an optimal parameter set exists to predict demand in different settings. Instead, algorithm are developed to select good parameters (Kück and Scholz-Reiter 2013) or to even select the optimal prediction method together with the parameters as done by Scholz-Reiter et al. (2014). However, our results indicate that to obtain a robust schedule often a simple over-provisioning of the demand for high risk articles, classified by means of a clustering algorithm, is sufficient.

Although we do not develop a new optimization technique, the presented study is original, to the best of our knowledge, because it combines the statistical analysis of production data with adapted heuristic optimization methods to our problem and the subsequent detailed simulation of a real production problem. The experiments give insights in the behavior of different local search heuristics and clearly indicate that local search heuristics are able to improve the makespan significantly compared to schedules resulting from simple heuristics like SPT as previously used in literature to solve our specific problem.

4 COMPUTATION OF SCHEDULES

Taking available results from literature into account, we choose to apply SPT, local search, and tabu search algorithms, and ECT to compute and optimize schedules for the basic model described in Section 2 and extend them for machine qualifications, skipping stages, and unrelated machines. In detail, we schedule the first stage with one of the algorithms and combine them in each case with ECT for scheduling the second stage. This procedure leads to the algorithms that are described in Algorithms 1–4. With its variants by choosing the parameters $method \in \{shift, swap\}$ and $tabu \in \{true, false\}$, we construct in total seven different optimization algorithms for the basic model of Section 2.

To create an initial solution, with Algorithm 1 successively the jobs are scheduled on stage 1 according to their increasing processing times on the machines that become available. When all jobs are scheduled to stage 1, stage 2 is scheduled with Algorithm 2 according their completion times of stage 1. Based on this initial solution, six algorithms that can be created with Algorithm 3 and Algorithm 4 optimize the solution. Both Algorithm 3 and Algorithm 4 are computed in "shift" or "swap" variant. "shift" places one randomly selected job to another position in the existing schedule, whereas "swap" exchanges the positions of two randomly selected jobs of the existing schedule. After one of these moves and the consideration of eligibility restrictions for the machines and stages, the schedule is checked for improvement by comparing makespans. In addition, Algorithm 4 can be executed with or without a tabu list ($tabu \in \{true, false\}$). With a tabu list, the algorithms do not test again a solution x that has already been tested and is included in the tabu list at the moment. Algorithm 3 and Algorithm 4 terminate, if there is no improvement in the makespan within a defined amount of $iterations$.

Algorithm 1 (Shortest Processing Time, SPT)

1. Order jobs j according to their increasing average processing times in stage 1 ($productionvolume_{ij} \cdot p_{ij}$) and save the queue in $PrioList1$.

2. Whenever a machine in stage 1 becomes available, select next unscheduled job j of *PrioList*1 that is qualified for the given machine and schedule j on the available machine.
3. Execute Algorithm 2 (ECT).
4. **return** schedule x with $C_{max}(x)$.

Algorithm 2 (Earliest Completion Time, ECT)

1. Order jobs on stage 2 according to their completion times C_{1j}. If a job is not processed on stage 1, set $C_{1j} := 0$. Of course, if a job should not be processed on stage 2, the job is not in the sequence for stage 2. If two jobs have the same completion time C_{1j}, order these jobs alphanumerically. Save queue in *PrioList*2.
2. Whenever a machine in stage 2 becomes available, select next unscheduled job j of *PrioList*2 that is qualified for the given machine and schedule j on the available machine.

Algorithm 3 (Local Search – Random Descent)

1. Given a feasible initial solution x with makespan $C_{max}(x)$.
   Choose parameter $iterations > 0$ and set $termination := iterations$.
   Choose parameter $method \in \{shift, swap\}$.
2. **while** $termination \neq 0$
   (a) Duplicate $x_n := x$.
   (b) Randomly choose a job j on stage $i = 1$. For this job $\exists! x_{ilkj} = 1$. According to $x_{ilkj} = 1$ define l and k.
   (c) Randomly choose a machine $l_n \in E_{ij}$.
   (d) **if** $method = shift$
      i. Shift job j to machine l_n and randomly choose a position to insert job j on this machine l_n.
   (e) **if** $method = swap$
      i. Randomly choose a job j_s on selected machine l_n. For this job $\exists! x_{il_nk_nj} = 1$. According to $x_{il_nk_nj} = 1$ define k_n.
      ii. Exchange positions of selected jobs j and j_s with setting $x_{ilkj} = 0$, $x_{il_nk_nj_s} = 0$, $x_{ilkj_s} = 1$, and $x_{il_njk_n} = 1$.
   (f) Save solution in x_n, execute Algorithm 2 (ECT) and compute makespan $C_{max}(x_n)$.
   (g) **if** $C_{max}(x_n) < C_{max}(x)$
         $x := x_n$, $C_{max}(x) := C_{max}(x_n)$.
         $termination := iterations$.
      else
         $termination := termination - 1$.
3. **return** x with $C_{max}(x)$.

Algorithm 4 (Tabu and Local Search – Steepest Descent)

1. Given a feasible initial solution x with makespan $C_{max}(x)$.
   Choose parameter $iterations$ and set $termination := iterations$.
   Choose parameter $method \in \{shift, swap\}$.
   Choose parameter $tabu \in \{true, false\}$.
2. **if** $tabu = true$
      Initialize $T := \{x\}$.
      Choose parameter $t > 0$.
   else
      Initialize $T := \{\}$.
3. **repeat** $termination \neq 0$
   (a) Duplicate $x_n := x$.
   (b) Set $N := \{\}$.
      for all jobs j on stage i.
         For the selected job $\exists! x_{ilkj} = 1$. According to $x_{ilkj} = 1$ define l and k.
         for all $l_n \in E_{ij}$
            if $method = shift$

Shift job j to machine l_n and randomly choose a position to insert job j on this machine l_n.
Save solution in x_n, execute Algorithm 2 (ECT) and compute makespan $C_{max}(x_n)$.
if $x_n \notin T$
$N = N \cup \{x_n\}$.
if $method = swap$
for all j_s on the selected machine l_n
According to $x_{il_nk_nj_s} = 1$ define k_n.
Exchange positions of jobs j and j_s with $x_{ilkj} = x_{il_nk_nj_s} = 0$, $x_{ilkj_s} = x_{il_nk_nj} = 1$.
Execute Algorithm 2 (ECT).
Save solution in x_n and compute makespan $C_{max}(x_n)$.
if $x_n \notin T$
$N = N \cup \{x_n\}$.
if $tabu = true$
$T = T \cup N$.
if $|T| > t$
Delete the $|T| - t$ elements from T, which are added earliest.
if $N = \{\}$
Set $termination := 0$.
for all jobs $x_n \in N$.
if $C_{max}(x_n) < C_{max}(x)$
$x := x_n$, $C_{max}(x) := C_{max}(x_n)$.
$termination := iterations$.
else
$termination := termination - 1$.

4. **return** x with $C_{max}(x)$.

5 PARAMETER UNCERTAINTY

The algorithms for computing schedules presented above all assume that the complete information about the problem is available. This situation can be rarely found in practice. Usually information about a system is uncertain and at most historical data are available to quantify uncertainty.

If uncertainty is modeled by random variables, in principle a stochastic optimization problem can be formulated (Van Hentenryck and Bent 2006). The solution of stochastic optimization problems turns out to be much more complex than the solution of their deterministic counterparts where random variables are substituted by deterministic values. This in particular holds for realistic stochastic hybrid flow shop problems, as the ones analyzed in this paper. For those models, even the analysis of a single configuration cannot be evaluated analytically. Instead, stochastic simulation has to be applied. For an optimization of such models, stochastic discrete event simulation has to be coupled with heuristic optimization methods (Juan et al. 2019). However, since stochastic simulation is time consuming and heuristic optimization methods tend to use a large number of function evaluations, the computation of almost optimal schedules can be cumbersome when started from some random schedule. To reduce computation time, it is much better to compare different near-optimal schedules from deterministic optimization models using a detailed simulation model. Another point is that any model is an approximation of the real system. This implies that the optimal solution for the simulation model, even if it can be found, is not necessarily optimal for the real system and it is more important to find a schedule which gives good results and is robust against small changes in the parameters or the specification of the random variables.

The major source of information about model parameters is the available data from the running system. In production systems, often a large amount of data is available and can be used to model uncertainty appropriately. We have to distinguish between internal parameters of the production system like processing or setup times, scrap rates, and availability of machines and external parameters, mainly the varying demand. We begin with the internal parameters that, up to some extent, can be measured in the production system.

In practice, quality and availability of production data vary considerably. In Table 1, we analyzed several scenarios where data are available at different levels of detail, and show how this data can be used in optimization and simulation. One aspect that should be mentioned is that due to outliers with high values it seems to be preferable to use the median instead of the mean value for processing and setup times in the optimization model to get more realistic schedules. In contrast, for scrap rates it is useful to use the mean value due to having a buffer.

Table 1: Data concept for scheduling parameters in optimization and simulation.

	processing times p_{ilj} and setup times s_i		scrap rate r_{ij}	
case	optimization	simulation	optimization	simulation
article data on dedicated machine available	median per article and machine	random value from distribution per article and machine	mean value per article and machine	random value from distribution per article and machine
no article data on dedicated machines available for the article, but article data on the stage available	median per article over all machines	random value from distribution per article over all machines	mean value per article over all machines	random value from distribution per article over all machines
no article-related and machine-related data available, but article-related planned data on the stage available	planned data for the article on the stage	planned data for the article on the stage	-	-
no article-related and machine-related data available, but data for all articles on the stage available	median over all machines for all articles	random value from distribution over all machines for all articles	mean value over all machines for all articles	random value from distribution over all machines for all articles

External uncertainty, mainly created by varying demands, is out of the control of the producer, but can also be predicted based on historical data. In particular, at the first stages of longer supply chains, uncertainty in demands can be significant and seems to grow in volatility and amount over the stages (Lee et al. 1997). Figure 1 shows the ordered quantities over time of a specific article for a supplier in the automotive industry. The last value in each time series shows the demand that is finally delivered. The other points show the development of the previously ordered amount for the same date.

It can be noticed that in this example the demand grows considerably in the last week. Thus, any schedule that has been made with available data one week before delivery will fail. Also during the planned week itself, changes can occur for these high risk articles. Fluctuation in demands is, of course, article-specific and it is not necessary or possible to model every article in detail. Hence, we first perform a cluster analysis based on historical data to distinguish between low-, medium- and high-risk articles. A high-risk article changes its demand often and significantly in the weeks before and of production, whereas for low-risk articles ordered quantities and final demand are very similar. For clustering, articles of the first and second stage are considered, because their demands are highly related and similar in their level of demands. If one article needs to be produced in both stages, both demands are considered and clustered in the algorithm. Clustering for the field data resulted in 3 or 4 clusters. For high risk articles, we use over-provisioning to avoid situations where the demand cannot be satisfied. Therefore, the demand of high risk articles is multiplied by article and stage related safety factors ($SF2_{ij}$) to keep the slot in the schedule reserved.

The same procedure can be applied to all article demands, e.g. if the production is very sensitive about producing less than the needed customer demand. First their deterministic median of the related scrap rate (see Table 1) is included in the demand and after that the demand is multiplied by a safety factor ($SF1_{ij}$). Notice that an earlier completion, resulting in unused reserved slots is easy to handle with the algorithms presented in section 2 because it only causes at most slightly extended storage cost. Whereas an unexpected

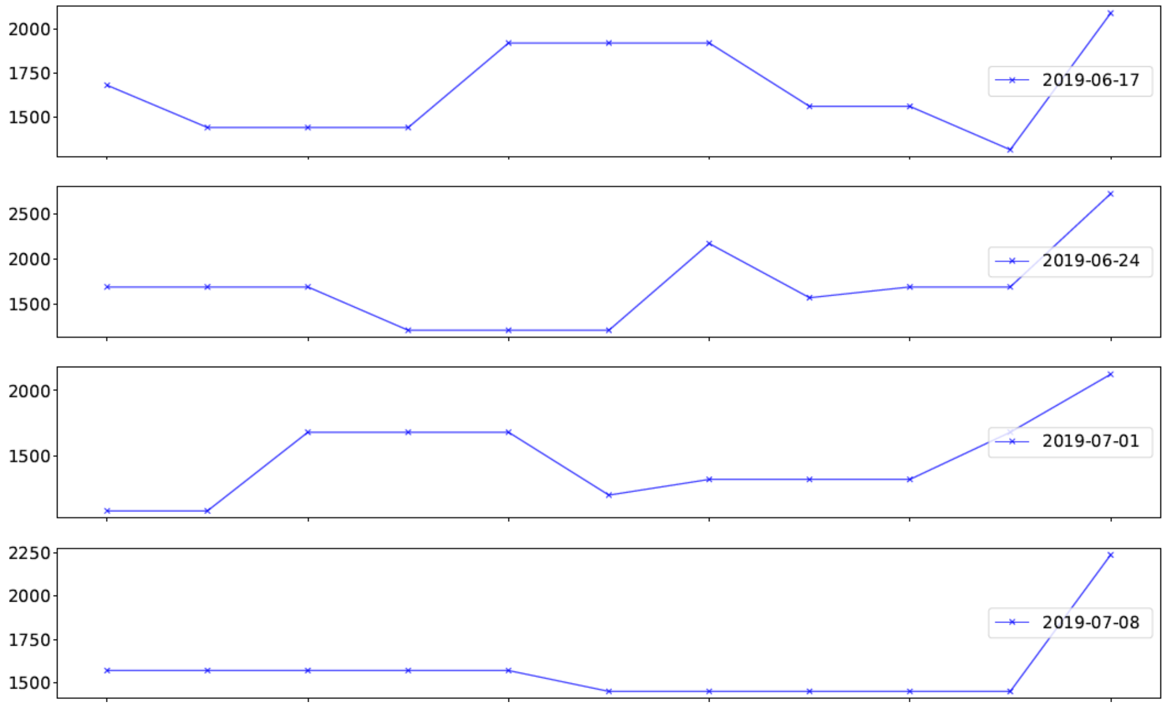

Figure 1: Time series of demands of a high risk article.

longer slot for articles would mean drastic delays for the schedule. All analyzed factors influencing the demand in our application case result in following formula to calculate the appropriate production quantity:

$$productionvolume_{ij} = \frac{demand_{ij} - stock_{ij}}{(1 - r_{ij})} \cdot SF1_{ij} \cdot SF2_{ij}$$

6 SCHEDULING UNDER UNCERTAINTY

The simulation model for our application case was build in AnyLogic 8.5. For every produced part the model generates values from the probability distributions mentioned in Table 1 and it considers also further specific characteristics of production. The following points of our application case are included in the simulation model, but not in the optimization model:

- There are two groups of machines. Only one machine of a group can be set up at the same time.
- Produced products, which are transported by conveyors to a quality check station in the first stage can block each other.
- Articles are moved from the first to the second stage by small carriers with fixed capacity.
- Detailed modeling of the connection of logistics and production at the beginning of the second stage with reservation of a trolley for a production worker, that puts the parts to the machine.
- Shift times: One of the two machines at the second stage does not produce during the night and both machines do not produce on weekends.
- Four stations with separate article related distributions, where scrap parts are identified.

To evaluate the algorithms by simulation, the schedules created by the algorithms of Section 4 are analyzed in the detailed simulation model using multiple replications in parallel. The number of replications is chosen with respect to the width of confidence intervals. The results of the C_{max} values are an indicator to judge how robust a given schedule is and which schedule is expected to perform best in production. To evaluate the scrap uncertainties analyzed in Section 5, we count how many simulation runs realize production volumes that are higher or equal to the required demands.

7 EXAMPLE RUNS

To evaluate the seven algorithms of Section 4, we use historical data of the company. Different safety factors are also considered to reduce and evaluate the risk of producing less demand than the customer requests, and a simulation of every schedule is performed. The application has the following characteristics:

1. Every week around 60 jobs j with $demand_{ij}$ are produced in the two-stage system.
2. The first stage contains 11 unrelated parallel machines and the second stage consists of 2 identical parallel machines.
3. Stochastic distributions per job and machine for the processing times p_{ilj} are available. The median of the processing time on stage 1 is about 80 seconds and on stage 2 it is 4 seconds per article.
4. For set-up times s_i in stage 1, stochastic distributions are available over all machines and article numbers. The median of set-up time on stage 1 is about 100 minutes and about 4 minutes on stage 2. On stage 2, set-up times are deterministic, because it is a conveyor belt production stage with a constant flow rate.
5. For scrap r_{ij}, article-related data are available for both stages. The overall average scrap rate in stage 1 is about 2,5 % and in stage 2 about 3,2 %.

We run the optimization algorithms with an Intel® Core™ i7-3770 3,4 GHz and 12 GB RAM and we run the simulations on a Intel® Xeon™ E5-2699 v4 2,2 GHz and 64 GB RAM.

Since it is very important in our application case that weekly demands are fulfilled completely, we first evaluate which safety factor for scrap (1,2 or 1,4 or 1,6) performs best. For this purpose, we have analyzed production volumes for each safety factor in the simulation for six selected calendar weeks, for which we have demand data. Figure 2 shows the percentages of simulation runs that produce one or more parts less for the weekly demand so that the weekly demand is not fulfilled. Per data point we run the simulation 280 times. Safety factors 1.4 and 1.6 provide similarly good results and it seems to be not realistic to reduce the probability of incomplete demands below 20 % without an extreme overproduction. For this reason, in the following we use $SF1_{ij} = 1,4$. We set $SF2_{ij} = 1,1$ to reserve slots for high-risk articles.

We run the metaheuristics with $iterations = 5000$ and a time restriction of 2 minutes without improvement. The metaheuristics include stochastic components. Therefore, the variation in objective function values with 16 replications for an exemplary week is shown in Figure 3. Steepest descent and tabu search algorithms provide the same values in every replication, while the random descent algorithms provide varying schedules. The borders in the box plot describe the first and third quartiles. Upper and lower limits describe the minimum and maximum makespan that appeared in the optimization runs. For random descent algorithms, the majority of values (included in the box) differs within about a quarter of a day, which means that the influence of stochastic is relatively small. Random descent (shift) provides worse makespan values and is less robust than random descent (swap), which means that for this data set and the chosen calendar week, random descent (swap) and the other algorithms with ”swap” are preferable. Even for the schedules of random descent, the variation in the results does not change the relations of performance between the metaheuristics concerning makespan essentially, so selecting one schedule per metaheuristic is a good choice.

Thus, for each week and each algorithm we now take one objective value, so that for each algorithm we have six data points. The analysis is illustrated in Figure 4. The largest computation time over all

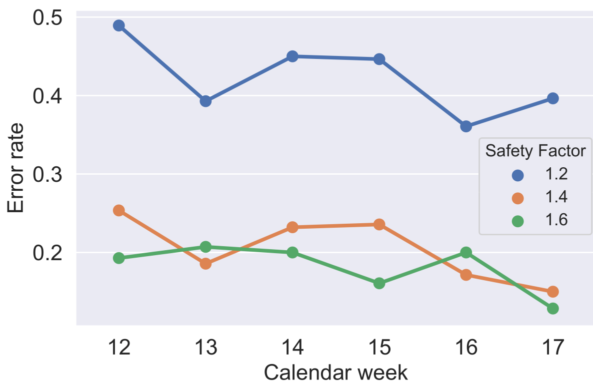

Figure 2: Percentage of simulation runs producing too few amounts due to scrap varying $SF1_{ij}$.

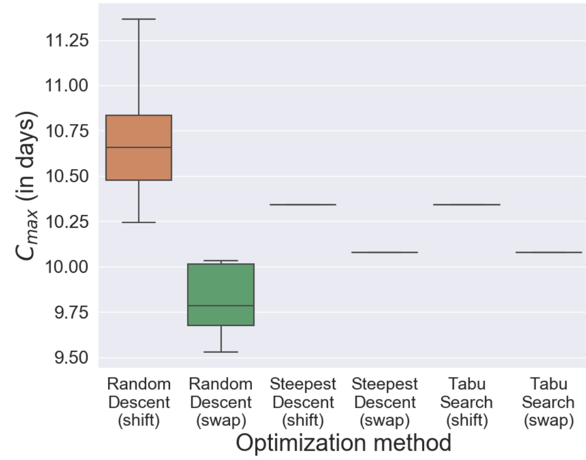

Figure 3: Optimization results with 16 replications per optimization method for calendar week 12 with $SF1_{ij} = 1{,}4$ and $SF2_{ij} = 1,1$.

algorithms and weeks was 2 minutes 49 seconds for tabu Search (swap) run and the minimal time was 6 ms for a calculation of SPT. We mentioned in Section 4 that we use the schedule resulting from SPT as initial solution for the local search and tabu search algorithms. Therefore, it is not surprising that according to Figure 4 the six metaheuristics provide better results than SPT. However, the difference between the values is considerable. By applying random decent (swap) instead of SPT, the makespan for calendar week 12 is reduced from 14,5 days to 9,5 days. In contrast, the differences of makespans comparing the six metaheuristics for each week are minor. Hence, with this evaluation, we cannot decide which of the metaheuristics performs best for the application.

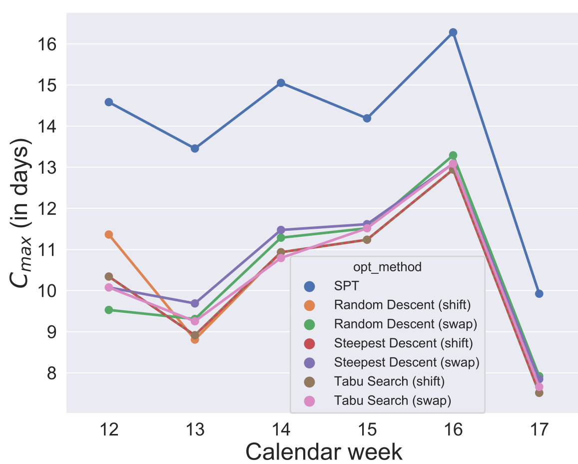

Figure 4: Optimization results per optimization method with $SF1_{ij} = 1{,}4$ and $SF2_{ij} = 1,1$.

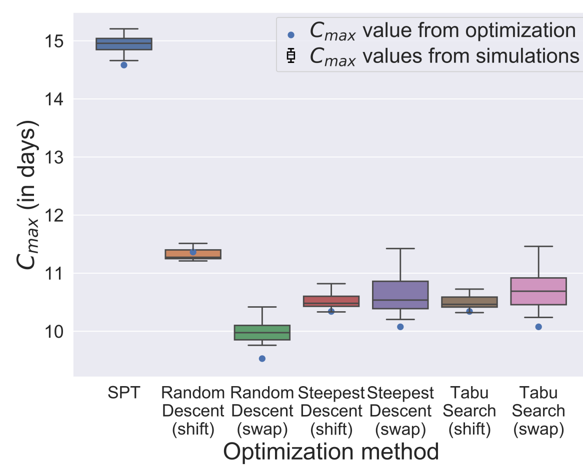

Figure 5: Simulation results for calendar week 12 with $SF1_{ij} = 1{,}4$ and $SF2_{ij} = 1,1$.

To decide under almost real conditions which method returns the schedules with the lowest makespan and which schedule is in comparison with others most robust against uncertainties in processing times or set up times, Figure 5 shows how makespans of the different schedules for week 12 performs in the simulation. The simulation is replicated 40 times, and only runs fulfilling the weekly demand are included.

Again, the borders in the box plot describe the first and third quartiles, and upper and lower limits show the minimum and maximum makespan of the simulation runs. Outliers are not included in Figure 5. The computation time of 40 parallel replications of the simulation is about 9 minutes. From the results it is obvious that for week 12 Random Decent (swap) provides the smallest mean value for makespan. It can also be seen that the schedule of Random Decent (swap) compared to the schedule of Steepest Decent (swap) it is more robust. The results of the simulation provide for all methods only slightly higher results than in the optimization, which is a strong hint that the additional details of the simulation model have only limited effects.

8 CONCLUSION

In this paper, we present a methodology to solve hybrid flow shop scheduling problems under uncertainty, which has been derived from a real industrial use case. It is shown that the combination of data analysis to estimate model parameters, heuristic optimization, and detailed stochastic simulation results in robust schedules that can be used in real systems. Metaheuristics provide significantly better results than SPT for the application case. The use of stochastic simulation to evaluate and improve schedules from the deterministic optimization problem indicates that robust schedules are generated. In addition, we have shown that, depending on the application, safety factors that can be multiplied by the demand, above a certain size, can reduce their effect on preventing incompletely fulfilled demand positions.

Our approach can be further developed by using more-efficient forecasting techniques to predict model parameters, by additional heuristics to generate schedules for the optimization model, and by combining the simulation model with additional techniques to perform optimization based on simulation.

REFERENCES

Almeder, C., and R. F. Hartl. 2013. “A Metaheuristic Optimization Approach for a Real-world Stochastic Flexible Flowshop Problem With Limited Buffer”. *International Journal of Production Economics* 145:88–95.

Borshchev, A., and I. Grigoryev. 2014. *The Big Book of Simulation Modeling*. AnyLogic.

Dios, M., V. Fernandez-Viagas, and J. M. Framinan. 2018. “Efficient Heuristics for the Hybrid Flow Shop Scheduling Problem With Missing Operations”. *Computers & Industrial Engineering* 115:88–99.

Figueira, G., and B. Almada-Lobo. 2014. “Hybrid Simulation-optimization Methods: A Taxonomy and Discussion”. *Simulation Modelling Practice and Theory* 46:118–134.

Jabbarizadeh, F., M. Zandieh, and D. Talebi. 2009. “Hybrid Flexible Flowshops With Sequence-dependent Setup Times and Machine Availability Constraints”. *Computers & Industrial Engineering* 57(3):949–957.

Johnson, S. M. 1954. “Optimal Two- and Three-stage Production Schedules With Setup Times Included”. *Naval Research Logistics Quarterly* 1(1):61–68.

Juan, A. A., J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. 2015. “A Review of Simheuristics: Extending Metaheuristics to Deal With Stochastic Combinatorial Optimization Problems”. *Operations Research Perspectives* 2:62–72.

Juan, A. A., J. Panadero, L. S. Reyes-Rubiano, J. Faulin, R. de la Torre, and J. I. Latorre. 2019. “Simulation-based Optimization in Transportation and Logistics: Comparing Sample Average Approximation with Simheuristics”. In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. Bae, S. Lazarova-Molnar, M. Rabe, P. Szabo, C. Haas, and Y.-J. Son, 1906–1917. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Komaki, G. M., S. Sheikh, and B. Malakooti. 2019. “Flow Shop Scheduling Problems With Assembly Operations: A Review and New Trends”. *International Journal of Production Research* 57(10):2926–2955.

Kück, M., and B. Scholz-Reiter. 2013. “A Genetic Algorithm to Optimize Lazy Learning Parameters for the Prediction of Customer Demands”. In *Proceedings of the 2013 International Conference on Machine Learning and Applications*, edited by M. Sayed-Mouchaweh and M. A. Wani, 160–165. Piscataway. NJ: Institute of Electrical and Electronics Engineers, Inc.

Law, A. M. 2015. *Simulation Modeling and Analysis*. 5th ed. New York, NY: McGraw Hill.

Lee, H. L., V. Padmanabhan, and S. Whang. 1997. "The Bullwhip Effect in Supply Chains". *Sloan Management Review* 38:93–102.

Lin, J. T., and C. Chen. 2015. "Simulation Optimization Approach for Hybrid Flow Shop Scheduling Problem in Semiconductor Back-end Manufacturing". *Simulation Modelling Practice and Theory* 51:100–114.

Low, C., C.-J. Hsu, and C.-T. Su. 2008. "A Two-stage Hybrid Flowshop Scheduling Problem With a Function Constraint and Unrelated Alternative Machines". *Computers & Operations Research* 35(3):845–853.

Murray, P. W., B. Agard, and M. A. Barajas. 2015. "Forecasting Supply Chain Demand by Clustering Customers". *IFAC-PapersOnLine* 48(3):1834–1839.

Naderi, B., R. Ruiz, and M. Zandieh. 2010. "Algorithms for a Realistic Variant of Flowshop Scheduling". *Computers & Operations Research* 37(2):236–246.

Ribas, I., R. Leisten, and J. M. Framiñan. 2010. "Review and Classification of Hybrid Flow Shop Scheduling Problems From a Production System and a Solutions Procedure Perspective". *Computers & Operations Research* 37(8):1439–1454.

Ruiz, R., F. S. Şerifoğlu, and T. Urlings. 2008. "Modeling Realistic Hybrid Flexible Flowshop Scheduling Problems". *Computers & Operations Research* 35(4):1151–1175.

Ruiz, R., and J. A. Vázquez-Rodríguez. 2010. "The Hybrid Flow Shop Scheduling Problem". *European Journal of Operational Research* 205(1):1–18.

Scholz-Reiter, B., M. Kück, and D. Lappe. 2014. "Prediction of Customer Demands for Production Planning – Automated Selection and Configuration of Suitable Prediction Methods". *CIRP Annals* 63:417–420.

Van Hentenryck, P., and R. Bent. 2006. *Online Stochastic Combinatorial Optimization*. Massachusetts Institute of Technology Press.

ACKNOWLEDGMENTS

We thank the reviewers for the detailed evaluation of the previous version of the paper which helped us to improve the paper in several aspects. This work has been realized with the financial support of the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the Research Training Group GRK 2193 "Adaption Intelligence of Factories in a Dynamic and Complex Environment".

AUTHOR BIOGRAPHIES

CHRISTIN SCHUMACHER is a researcher in the department of Computer Science at TU Dortmund university. She holds a M.Sc. in business mathematics and is member of the DFG Research Training Group "Adaption Intelligence of Factories in a Dynamic and Complex Environment". Her research interests include mathematical optimization, heuristics and discrete event simulation for machine scheduling and real-world optimization problems. Her e-mail address is christin.schumacher@tu-dortmund.de. Her website is https://ls4-www.cs.tu-dortmund.de/cms/de/home/schumacher/.

PETER BUCHHOLZ is a professor for modeling and simulation in the department of computer science of TU Dortmund university. His research interests include stochastic modeling, stochastic discrete event simulation, numerical methods for stochastic models, and optimization under uncertainty. He applies the techniques mainly in computer networks and logistics. His e-mail address is peter.buchholz@cs.tu-dortmund.de. His website is https://ls4-www.cs.tu-dortmund.de/cms/de/home/buchholz/.

KEVIN FIEDLER is a student assistant in the department of computer science of TU Dortmund university. He holds a B.Sc. in computer science from TU Dortmund University. His research interests include stochastic discrete event simulation with Anylogic and metaheuristics for machine scheduling. His e-mail address is kevin.fiedler@tu-dortmund.de.

NICO GORECKI is a student assistant in the department of computer science of TU Dortmund university. He holds a B.Sc. in computer science from TU Dortmund University. His research interests include data analysis and application of machine learning algorithms on real data. His e-mail address is nico.gorecki@tu-dortmund.de.